arXiv:2603.09262v1 [cs.DS] 10 Mar 2026

关于在线加权不相交匹配问题*

Joan Boyar
University of Southern Denmark
joan@imada.sdu.dk

Shahin Kamali
York University
kamalis@york.ca

Kim S. Larsen
University of Southern Denmark
kslarsen@imada.sdu.dk

Ali Fata Lavasani
Huawei, Canada
fata.lavasani@gmail.com

Yaqiao Li
Shenzhen University of Advanced Technology
liyaqiao@suat-sz.edu.cn

Denis Pankratov
Concordia University
denis.pankratov@concordia.ca

2026年3月5日

摘要

我们提出并研究欧几里得平面中带有非交叉约束的在线匹配问题的加权版本：带非负权重的点在线到达，算法可以将新到达的点与此前到达且尚未匹配的某个点进行匹配。在经典模型中，对于新到达的点是否进行匹配以及如何匹配，一经决定便不可撤销。目标是在匹配中的边所对应的直线段互不相交这一约束下，使匹配点的总权重最大化。该问题的无权版本由 Atallah 于 1985年 在离线设定中提出；近来，已有多篇论文在有建议和无建议的在线设定中研究这一问题。

*Boyar 和 Larsen 部分获得了丹麦独立研究委员会项目资助，项目编号为 DFF-0135-00018B 和 DFF-4283-00079B。Kamali 部分获得了加拿大自然科学与工程研究委员会项目资助，项目编号为 DGECR-2018- 00059。Pankratov 部分获得了加拿大自然科学与工程研究委员会的资助。

我们指出，对于加权问题，确定性在线算法无法保证非平凡竞争比；但我们给出了有界权重情形下该问题的上下界。与确定性情形不同，我们证明，采用随机化方法时，任意权重下都能实现常数竞争比。我们还研究了该问题的其他变体，包括可撤销性和共线点；这两种变体都允许存在非平凡在线算法，我们也给出了可达到的竞争比的上下界。最后，我们证明了实现最优性所需建议复杂度的一个界，改进了已知的最佳界。

1 引言

我们提出并研究以下问题，称为在线加权非交叉匹配（OWNM）。假设欧几里得平面上的 $2n$ 个点 $p_1, \ldots, p_{2n}$ 依次在线到达。当 p_i 到达时，会揭示其正权重$w(p_i) \in \mathbb{R}_{>0}$，此时算法可以选择将 p_i与一个此前已揭示且尚未匹配的点进行匹配，也可以让 p_i 保持未匹配。在经典在线模型中，算法的决策不可撤销。问题还受到非交叉约束，即匹配中各边对应的直线段不得相交。假设这些点处于一般位置，[1] 目标是设计一种算法，使匹配点的权重总和最大化。

人们对匹配问题中的几何设定，尤其是欧几里得平面设定感兴趣，源于其在图像处理 [14]和电路板设计 [20]中的应用。在这类应用中，通常需要在表示顶点的各种几何形状（如矩形或圆）之间构造匹配，并使用直线段或更一般的曲线连接。由于边需要满足诸如避免边之间相交以及避免边与顶点相交等约束，几何因素也由此产生。这些约束会显著影响该问题的离线复杂度。

不相交匹配问题的无权版本（当所有 $i \in \{1, \ldots, 2n\}$ 都满足 $w(p_i) = 1$ 时）已在离线设定 [8, 19]和在线设定 [11, 30, 21, 24]中得到研究。我们将在第2节详细回顾该问题的发展历程

[1]“一般位置”是计算几何中一个常用术语，用来表示我们不打算处理特殊情况，通常是因为这类情况处理起来繁琐、乏味，还会使算法的核心部分变得冗杂。对于我们的问题，在某些变体中，我们希望忽略的一种特殊情况是（三个或更多）共线点。

。现在只需指出，预先知道所有点的位置的离线算法，可以在满足非交叉约束的同时匹配所有点。这就是竞争分析中所说的 OPT 概念；可以将其视为计算复杂度不受限制时最优的离线算法。因此， OPT 的值始终为 $W := \sum_{i=1}^{2n} w(p_i)$。在线算法的性能通过其竞争比来衡量；对于我们的问题，该竞争比对应于算法在最坏情况下能够保证实现的 W 的比例。

很容易看出，当点的权重不受任何限制时，不存在具有竞争力的确定性在线算法，即无法获得大于零的竞争比（特别是，这是定理 3 的直接推论）。我们研究了若干不同情形，在这些情形下，该问题存在能够达到非平凡竞争比的算法。

2 相关工作

给定 $2n$ 个处于 $\mathbb{R}^2$ 一般位置的点，基本的不相交匹配（NM）问题是寻找边数尽可能多的不相交匹配。注意，最小长度欧氏匹配是不相交的[2]。因此，完美不相交匹配总是存在。不相交匹配问题及其变体已在离线设定中得到广泛研究。Hershberger 和 Suri [19] 提出了一种算法，可在 $\Theta(n \log n)$ 时间内找到完美不相交匹配。Atallah [8] 首先提出了该问题双色版本的高效算法，随后 Dumitrescu 和 Steiger [15]也提出了此类算法。在该版本中，点被划分为两个子集，且匹配边只能在两个子集之间形成。文献中研究的其他问题变体包括要求不相交匹配是稳定的 [29, 18], 、要求两个不相交匹配兼容（如果两个不相交匹配中的边仅共享端点，则它们也互不相交） [2],，以及要求兼容的不相交匹配满足特定的多样性约束 [23]。

一些研究考察了所有NM上的优化问题。目标函数包括最大化匹配边的欧氏长度之和 [5], 最小化最长匹配边的长度 [1],以及其他类似的最小化与最大化组合 [22]。

[2]证明采用反证法。取一个总长度最小的完美匹配。如果它存在交叉，就考察由此诱导的四边形——对角线的长度大于一对对边的长度。现在，用较短的一对边替换对角线。这个新的完美匹配总长度更小，这与我们选择的完美匹配相矛盾。因此，总长度最小的匹配不可能存在任何交叉。

另一条研究路线是放宽非交叉约束，允许出现某些交叉。一个重要问题是理解交叉族的规模，即两两交叉的匹配边所组成的集合。Pach等人 [25] 取得的突破表明，最大的交叉族具有线性规模。Aichholzer等人 [4]研究了 k-交叉匹配的计数问题。

至少有两项研究 [10, 31] 探讨了点集上的加权NM，其中每个点的权重属于 $\{1, 2, \ldots, n\}$。Balogh等人 [10] 将一条边的权重定义为其两个端点权重之和模 n，并研究了边权互不相同的NM的典型大小（这称为非交叉调和匹配）。Sakai和Urrutia [31] 将一条边的权重定义为其两个端点权重中的较小值，并研究了最大加权NM的下界和上界。

在纯数学中，NM被用作理解群的表示论的工具 [7, 26]。满足特定性质的一组NM（即所谓的项链）被用于研究与某个多项式相关的调和代数曲线的拓扑 $\mathbb{C}$ [32]。文献 [3, 17]研究了极值图问题，其中大小为 k 的NM起到禁止子图的作用。

除了引言中提到的图像处理和电路设计方面的应用，NM 还应用于其他领域。其中一项重要应用是计算生物学。研究人员通过研究 NM 的一种受限版本（所有点都位于圆上）以及 k-不相交匹配（不存在 k 对彼此相交的边；当 $k = 2$时，它可简化为标准 NM），来理解 RNA结构 [9, 13, 33]。在与可见性问题（例如机器人学）和几何形状匹配相关的应用中，人们会将所研究的全部或部分点替换为几何对象 [27, 6]。例如，当问题是将对象与对象匹配时，可以从 A 中任意选择一个点 p，再从 B中任意选择一个点 q，在这两个对象 A 和 B 之间构造一条边（p, q），前提是该边（p, q）不会与其他对象相交。

在线 NM 直到最近才开始受到研究。Bose等人 [11] 开创了在线（无权）NM 的研究，并证明确定性算法的竞争比为 $2/3$；而 Kamali等人 [21] 给出了一种随机化算法，其期望匹配的点数约占所有点的 0.6695。在线双色NM也在 [11,30]中得到了研究。最后， [11, 24]研究了建议复杂度。特别是，Lavasani和Pankratov [24] 确定了在圆上最优求解在线双色NM所需的建议复杂度，并给出了下界 $n/3 - 1$

以及平面中在线 NM 建议复杂度的上界 $3n$ 。

3 主要贡献

我们针对问题 OnlineWeighted Non-Crossing Matching (OWNM) 及其算法和问题变体所得到的结果可概括如下:

•有界权重下的确定性算法：在 RestrictedOWNM 中，我们假设点的权重限制在区间 $[L, U]$ 内，其中 $L \leq U$，且算法在执行开始时已知该值。注意，通过缩放，我们可以不失一般性地假设 $L = 1$，即所有权重都属于 $[1, U]$。我们证明，任何确定性在线算法的竞争比都是 $O\left(2^{-\sqrt{\log U}}\right)$（定理 3）。我们还提出了一种名为 Wait-and-Match (WAM) 的确定性在线算法，其竞争比为 Ω $\left(2^{-2\sqrt{\log U}}\right)$（定理 7）。

• 随机化算法：我们证明了一个或许出人意料的结果：仅靠随机化就足以在任意权重下保证常数竞争比。我们提出了一种简单的随机在线算法，称为 Tree-Guided-Matching (TGM)，并证明其竞争比为 $1/3$（定理 9）。此外，我们还证明，即使对于该问题的无权版本，任何随机在线算法都无法达到优于 $16/17$ 的竞争比（定理 8）。

• 可撤销设置：我们证明，允许可撤销接受（模型定义见第 7 节开头），即使权重不受限制，也能通过确定性算法获得约 0.2862 的竞争比（定理12）。此外，我们还证明，任何允许撤销的确定性算法都无法达到优于 $2/3$ 的竞争比（定理 10）。

• 共线点：我们还研究了点不处于一般位置、而是全部位于一条直线上的情形（见第 8 节）。我们证明，即使在无权情形下，仅靠撤销或随机化也无法实现非平凡竞争比。然而，

我们提出了一种带撤销的随机化算法，在无权版本中实现了至少为 0.5 的竞争比。

- 建议复杂度：我们提出了一种称为 Split-and- Match（SAM）的新算法，该算法使用 $\lceil \log C_n \rceil < 2n$ 个建议位（模型的定义见第9节开头）来实现最优性（定理16），其中 C_n 是 n阶卡塔兰数。这改进了此前已知的问题 [24]建议复杂度上界 $3n$ 。由于 SAM 能够实现完美匹配，因此给定的点是否带有权重并不重要。

4 预备知识

本文所研究的匹配问题的输入是一个在线点序列 $I = \langle p_1, \ldots\ldots, p_{2n} \rangle$，其中的点处于一般位置，且 p_i 具有正实数权重 $w(p_i) \in \mathbb{R}_{>0}$。我们用 W 表示所有点的总权重，即 $W = \sum_{i=1}^{2n} w(p_i)$ 。对于受限 OWNM，权重被假设位于区间 $[1, U]$ 内，其中 U是某个已知值；该值属于问题定义的一部分，因此不是输入的一部分。当 p_i到达时，在线算法必须将其保持为未匹配点，或将其与某个未匹配点 p_j $(j < i)$ 进行匹配，前提是连接 p_i 和 p_j的线段（记为 p_ip_j）不与此前匹配对之间的线段相交。目标是最大化匹配点的总权重。对于在线算法 ALG（相应地，对于离线最优算法 OPT），我们使用 ALG（I）（相应地，OPT（I））表示该算法在输入 I上所匹配点的总权重。在上下文明确时，符号 pq 也用于表示经过点 p 和 q的整条直线，该直线将一个凸区域划分为两个子区域。

—

我们称确定性在线算法 ALG是 ρ-竞争的，如果存在常数 c，使得对每一个输入序列 I，我们有

$$\mathrm{ALG}(I) \geq \rho \cdot \mathrm{OPT}(I) - c.$$

对于随机化算法，上述不等式替换为如下形式

$$\mathbb{E}(\mathrm{ALG}(I)) \geq \rho \cdot \mathrm{OPT}(I) - c.$$

如果 $c = 0$，那么我们称竞争比 ρ 严格，并称 ALG 是严格 ρ-竞争的。如果 $c \neq 0$，那么该比率可称为渐近比率。注意

对于受限的 OWNM，在考虑渐近竞争性时，我们允许 c 依赖于 U。因此，达到渐近竞争比 ρ 的算法可以在 W 的 $(1-\rho)$ 比例之外，额外留下常数个点未匹配（无论这些点的权重如何）。

4.1 预热：双权重情形

作为预热，我们首先考虑点的权重只能为 1 或 $U > 1$ 的情形。我们后续的若干算法和分析都是双权重情形的推广，或受其启发。此外，正如我们将看到的，在这种情形下，我们的算法达到了紧确的竞争比 $1/3$ as $U \to \infty$。顺便一提，$1/3$再次作为我们在第 6 节中随机化算法的竞争比上界出现。

我们考虑以下称为"Two-Weight-Matching"（TWM）的算法。假设这些点出现在一个包围盒 B内。在整个执行过程中，TWM 维护着对 B的凸划分。初始时，整个 B 构成一个区域。只有当两个点位于同一个凸区域内时，算法才会匹配它们。当凸区域 R 中的两个点匹配后，连接它们的线段会延伸，直至碰到 R的边界，从而将 R 划分为两个更小的凸区域。

当点 p 到达区域 R，且其中已有一些未匹配点时，TWM 会在以下情况下将 p 与另一个点匹配：（1）如果 p 的权重为 U，则 p 会与权重最大的点 q 匹配；（2）如果 p 的权重为 1，且 R中存在一个权重为 U 的点，则 p 会与该点匹配；（3）否则，p 以及 R 中的所有未匹配点的权重均为 1，并且如果能够以一种不会使任何新凸区域变为空的方式进行匹配，p 就会与 R 中的一个点 q 匹配。

观察：在 TWM 的执行过程中，有一个不变量：如果一个区域包含多个点，则所有点的权重均为 1；并且没有任何区域包含超过三个未匹配点。

定理 1 对于 $U \geq 3$的双权重 OWNM，其竞争比至少为 $1/3$。

证明 在 TWM 完成时，令 m_1 为已匹配的权重为 1 的点的数量，m_U 为已匹配的权重为 U 的点的数量。因此，$\text{TWM} = m_1 + Um_U$。此外，令 r_1 为至少包含一个权重为 1 的未匹配点的区域数量，且 r_U

为包含一个权重为 U 的未匹配点的区域数量。根据上述观察，这些是所有可能包含未匹配点的区域，而这两种类型的区域互不相交。

由于将一个区域分成两个区域需要两个匹配点，$r_1 + r_U \leq \frac{m_1+m_U}{2} + 1$，因此 $r_1 \leq \frac{m_1+m_U}{2} + 1 - r_U$。

假设区域 R_1 中有一个权重为 U 的未匹配点，并令R_2 为通过匹配两个点 p 和 q而与 R_1 同时产生的另一个区域。如果 p 和 q 的权重都为 1，那么 R_1 和 R_2都至少包含一个权重为 1 的点，因此，R_1 不可能有权重为 U的未匹配点，这就产生了矛盾。因此，p 或 q 的权重为 U。也就是说，这两个点的匹配并未考虑是否可能留下空区域，因此，匹配之后，权重为 U 的点可能进入空区域 R_1 和 R_2。由此，$r_U \leq 2m_U$。

由于 OPT 匹配了所有点，

$$\begin{aligned}
\mathrm{OPT} &\leq m_1 + Um_U + 3r_1 + Ur_U \\
&\leq m_1 + Um_U + 3(\frac{m_1+m_U}{2} + 1 - r_U) + Ur_U, \text{ by the above} \\
&= m_1 + Um_U + 3(\frac{m_1+m_U}{2} + 1) + (U-3)r_U \\
&\leq m_1 + Um_U + 3(\frac{m_1+m_U}{2} + 1) + (U-3)2m_U, \text{ since } U \geq 3 \\
&\leq \frac{5}{2}m_1 + 3Um_U + 3\frac{m_U}{2} + 3 - 6m_U \\
&\leq 3m_1 + 3Um_U = 3\mathrm{TWM}, \text{ since } m_U \geq 1.
\end{aligned}$$

因此，$\mathrm{TWM} \geq \frac{1}{3}\mathrm{OPT} - 1$，结论得证。 □

现在我们证明，对于任意大的 U，不存在竞争比优于 $1/3$ 的算法。

定理 2 对于双权重 OWNM，任何确定性在线算法的竞争比至多为 $1/3 + 2/(3U + 3)$。

证明 在此证明中，对手给出的所有点都位于圆上，且所有弦均不相交。由于所有点都位于圆上，这也意味着每当两个点匹配时，就会形成一条弦。这实际上确保了在在线算法后续处理时，弦一侧区域中的点不会与另一侧区域中的点匹配。对于给定的算法 ALG，对手给出

$2k$ 个权重为 1 的点。 ALG 匹配了若干对点，形成了 s 条弦。如果 $s < \frac{k}{3}$，序列就此结束，此时比率为 $\frac{2s}{2k} < \frac{1}{3}$。否则，我们分别考虑 s 将圆划分成的 $s+1$ 个区域 ALG 。我们忽略其中一个区域，这样就能将其余的 s 个区域与弦一一对应。该区域中不会再给出任何点，而 ALG 对来自最初 $2k$ 个点的点所作的任何匹配都计入其他区域，因此忽略该区域不会导致我们低估 ALG 的收益。

令 S_4 表示至少含有四个未匹配点的区域，并令 S_0、S_1、 S_2 和 S_3 分别表示含有 0、1、2 或 3 个点的区域。下面我们讨论若干种情况。讨论完一种情况后，我们会在括号中列出 ALG 和 OPT 的收益。对于给定区域，我们将与该区域关联的弦所匹配的两个点也计入。因此，在任何区域中（我们忽略的区域除外）， ALG 和 OPT 的收益始终至少为二。 OPT 当然可以离线处理这些点，并匹配所有 $2n$ 点。不过，在证明中，我们会分别计算并比较 ALG 和 OPT 在每个区域中的加权点数（有时在创建新区域时会将两个区域合并计算）。

对于 S_0 区域，对手给出一个权重为 U 的点，而 ALG 无法匹配该点（ALG: 2， OPT: $U+2$）。

对于 S_1 个区域，对手给出一个权重为 U 的点。如果 ALG 不匹配当前这两个点，就不会再向该区域给出其他点（ALG: 2， OPT: $U+3$）。否则，对手会在表示最新匹配的直线两侧各给出一个权重为 U 的点，而 ALG 无法匹配这两个点（ALG: $U+3$， OPT: $3U+3$）。

对于 S_2 个区域，对手再给出一个权重为 1 的点。如果 ALG不将该点与另外两个点中的任意一个匹配，我们就转到 S_3 情形。否则， ALG创建了一个空区域，对手便在该区域内给出一个权重为 U 的点，而 ALG 无法匹配该点（ALG: 4， OPT: $U+5$）。

对于 S_3 个区域，对手再给出一个权重为 1 的点。如果 ALG不匹配该点，就不会再给出其他点，此时属于S_4 情形。否则，如果 ALG创建了一个没有点的区域，对手便在该区域内给出一个权重为 U 的点，而 ALG 无法匹配该点（ALG: 4，OPT: $U+6$）。否则，它创建了两个区域，每个区域各有一个未匹配点。我们按照 S_1 情形中的做法，分别独立处理这两个区域。

对于 S_4-区域，没有给出更多的点（ALG: 2， OPT: 至少6）。

处理完这些情形后，我们观察到，对于较大的 U， ALG 与 OPT 之间的最大比率为 $\frac{U+3}{3U+3} = \frac{1}{3} + \frac{2}{3U+3}$。

5 受限 OWNM 的确定性算法

本节将不再局限于双权重情形，而是探讨基于分类方案作出算法决策；在该方案中，所有权重落在同一类别内的点都以相同方式处理。

5.1 点分类。

在下界和上界论证中，我们都使用一种基于参数 $k \in \mathbb{N}$ 和 $U \in \mathbb{R}$ 的点分类，现说明如下。令 $k = \lceil \sqrt{\log U} \rceil$，并定义 $a_0, a_1, \ldots, a_k$ 的取值，使得

$$a_0 = 1, a_k = U, r = a_1/a_0 = a_2/a_1 = \ldots = a_k/a_{k-1},$$

这意味着 $r = U^{1/k}$ 和 $a_i = r^i$。给定一个值 $w \in [1, U]$，定义 $\lfloor w \rfloor$ 为满足 $a_i \leq w$ 的最大 a_i。下文中，权重为 w 的点称为类型 i，如果 $\lfloor w \rfloor = a_i$。因此，共有 $k+1$ 种不同类型，其中类型 k 仅包含值 U。两匹配点 x 和 y 之间线段的类型，定义为权重较大的端点的类型；也就是说，如果线段的一个端点属于类型 i，另一个端点的类型至多为 i，则 xy 属于类型 i。

5.2 否定性结果

定理 3 对于受限的 OWNM 问题，任何确定性在线算法的渐近竞争比率都是 $O\left(2^{-\sqrt{\log U}}\right)$。

证明 设 ALG为任意在线确定性算法。我们采用对手论证。对手将所有点都置于圆 C 上，因此算法所做的任何匹配都会在圆内形成一条弦，将之前的一个区域一分为二。在任意时刻，对手都会在 C 的一个活动区域中给出一个点；该区域由一条或两条弧构成，即由 C 边界上两个相邻点界定的圆周线段。初始时，整个圆构成活动区域。对手策略是维护从未匹配点到匹配点的映射，以确保匹配点的总权重与未匹配点的总权重之比为 $O\left(2^{-\sqrt{\log U}}\right)$。注意，这也意味着匹配点的总权重与所有点的总权重之比为 $O\left(2^{-\sqrt{\log U}}\right)$。

对手首先固定一个任意大的数 m （这是为了保证我们的界是渐近的）。对手在圆上任意位置给出权重为 1 的点，直到算法匹配了 m 对点，或圆上的点达到 $m2^k$ 个。若出现后一种情况，则竞争比至多为 $O\left(2^{-k}\right)$ $= O\left(2^{-\sqrt{\log U}}\right)$。 ——

因此，我们可以假设， ALG 最终匹配了 m 个点对，形成了互不相交的弦和 $m+1$ 个区域。现在，让每个匹配对各自负责一个不同的区域，不过第一个匹配对负责两个区域，最初是前两个区域。假设一条新弦 xy 将区域 R 分成两部分。令 $\{x_R, y_R\}$ 为负责 R的匹配对， R_1 为 R 中边界上有 $x_R y_R$ 的一侧， R_2 为另一侧。仍由 $\{x_R, y_R\}$ 负责 R_1，并让 $\{x, y\}$ 负责 R_2。这样，除第一个匹配对外，每个匹配对都恰好负责一个区域，而第一个匹配对负责两个区域。

对于每个区域 R，对手将 R 设为活动区域，执行以下过程，然后继续处理下一个区域，直到覆盖所有区域。令 $\{x_R, y_R\}$ 为负责 R的点对。根据 R 中未匹配点的数量，考虑以下两种情况：

情形1。 如果 R 中未匹配点的数量至少为 $2^k - 1$，对手就不会在 R 中给出任何点，而是继续处理下一个区域。在这种情况下，我们将 R 中的未匹配点映射到匹配对 $\{x_R, y_R\}$。注意，至少有 $2^k - 1$ 个权重为 1 的点被映射到总权重为 2 的线段。匹配点的权重与未匹配点的权重之比至多为 $2/(2^k - 1) \in O\left(2^{-\sqrt{\log U}}\right)$。 ——

情形2。 如果 R 中未匹配点的数量少于 $2^k - 1$，对手计划按以下方式给出一个点序列 $P = \langle p_1, p_2, \ldots\ldots, p_k \rangle$，其权重分别为 $a_1, a_2, \ldots\ldots, a_k$，并按权重升序排列（见图 1）。权重为 a_1 的点 p_1 出现在 R 中的任意位置（圆上）。权重为 a_i 的点 p_i 到达时， $i \in \{1, \ldots, k\}$， ALG 要么将它与一个权重为 1 的点匹配，要么将其留为未匹配点。后一种情况下，对手不会在 R 中给出更多点，而是继续处理下一个区域。

在前一种情况下，当 ALG将权重为 a_i 的点 p_i 与权重为 1 的点 q 进行匹配时，将 $p_i q$ 的一侧（其中包含至多一半的未匹配点）设为活动区域。对手继续在活动区域中给出 P 的剩余点。因此，位于

p_iq 另一侧的未匹配点将保持未匹配状态，因为 $\overline{p_i\overline{q}}$ 位于新点与这些未匹配点之间。

因此，在匹配 p_i 和 q之后， P 中能够与未来点进行匹配的权重为 1 的未匹配点数量至少减少一半。设 p_j 是算法在 P 中留下未匹配的第一个点。鉴于对手最多可以给出 k 个点，而 R中最初少于 2^k-1 个未匹配点，因此存在这样一个权重为 a_j的 p_j 。此时，对手结束针对 R 的过程，并继续处理下一个区域。

p_j 到达之前， R 中匹配对内点的总权重为:

$$\begin{aligned} M = & \underbrace{2}_{\text{for } (x_R, y_R)} + \underbrace{j-1}_{\text{endpoints of weight 1}} + \underbrace{a_1 + a_2 + \ldots + a_{j-1}}_{\text{endpoints with weight } a_i} \\ \leq & \frac{r^j - 1}{r-1} + j \end{aligned}$$

鉴于未匹配点 p_j 的权重为 a_j，匹配点与未匹配点的权重比至多为 $M/r^j \in O\left(1/r\right) = O\left(2^{-\sqrt{\log U}}\right)$。

鉴于每个匹配对至少对应一个区域，上述过程在所有情况下都建立了从匹配点到未匹配点的映射，其权重比为 $O\left(2^{-\sqrt{\log U}}\right)$，符合要求。□

5.3 正面结果：等待并匹配算法

我们提出一种名为“Wait-and-Match”的算法（WAM）。假设这些点出现在包围盒 B中。在其整个执行过程中， WAM 维护 B的凸划分 。最初，只有一个由整个 B构成的区域。如我们将描述的，该算法仅当两个点出现在同一凸区域中时才对它们进行匹配。每当凸区域 R 中的两个点被匹配时，它们之间的线段会被延长，直至触及 R的边界，从而将 R 划分为两个更小的凸区域。我们使用与第 5.1 节中所定义的相同点分类。

假设出现一个新的点 p，并令 R 表示 p的凸区域。在决定将 p 与哪个点匹配（如果有的话）时，算法按权重非递增顺序考虑 R 中的所有未匹配点。令 q 为下一个被考虑的点，并令 i为 p的类型的最大值

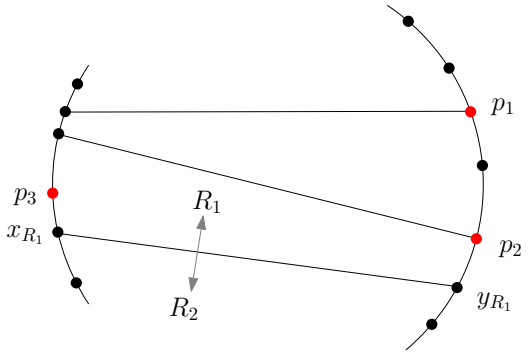


图1： $k = 3$下情形2中对手策略的示意图。两条弧构成活动区域。未标注的点权重均为1。假设在第一阶段， ALG 匹配了 x_{R_1} 和y_{R_1}，由此负责区域 R_1。注意， R_1 中未匹配点（权重为1）的数量为6，小于 $2^k - 1 = 7$。因此，在第二阶段，对手计划将权重分别为 a_1、 a_2、 a_3 的点 p_1、 p_2、 p_3 放入 R_1。假设 ALG 匹配了权重为 a_1的 p_1 ； 那么，对手会将 p_2 和 p_3放在线段下方，该线段连接这对匹配点（该处的未匹配点更少）。类似地，在权重为 a_2 的点 p_2 匹配后，对手会将 p_3 放在由此形成的线段上未匹配点所在的一侧。这样可以确保权重为 a_i 的某个点（此处为 a_3）保持未匹配状态，并映射到这些匹配对。

以及 q的类型。若pq的每一侧至少有$2^{k-i} - 1$ 个未匹配点，则算法将 p 与 q 匹配。若不存在合适的 q ， 则 p 保持未匹配。

示例： 假设 $k = 2$。那么 $a_0 = 1$， $a_1 = \sqrt{U}$， 以及 $a_2 = U$。设 p是一个权重为 1 的点。当一个点 p到达时，如果 pq两侧各至少有 $2^{2-2} - 1 = 0$ 个点，算法就将p 与任意一个权重为 U的点 q 匹配。也就是说，如果该区域内存在权重为 U 的未匹配点，算法就会无条件地将 p 与之匹配。类似地，如果该区域内不存在权重为 U 的未匹配点，只要 pq两侧各至少有 $2^{2-1} - 1 = 1$ 个点，算法就会尝试将 p与任意一个权重在 $[a_1 = \sqrt{U}, a_2 = U)$ 内的点 q匹配。最后，如果前述情况均未出现，只要 pq 两侧各至少有 $2^{2-0} - 1 = 3$个未匹配点，算法就会尝试将 p与任意一个权重在 $[a_0 = 1, a_1 = \sqrt{U})$ 内的点 q匹配。若该区域内至少有 7 个未匹配点，就会出现这种情况。$\diamondsuit$

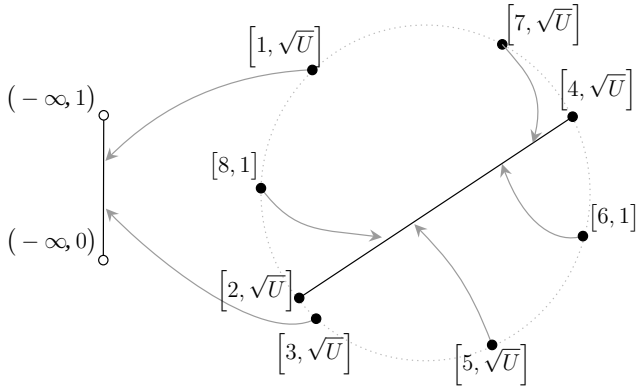


图2：用于分析 WAM 的映射示意图。在此示例中，我们有 $k=2$，且 $\{1,\sqrt{U},U\}$ 中有 8 个带权重点。这里，$[t,w]$ 表示输入序列中权重为 w 的 t^{th} 点。注意，点 1 和点 3 被映射到与权重为 U 的虚点 $(-\infty,\ 0)$ 和 $(-\infty,\ 1)$ 对应的线段上。

为了分析该算法，我们将每个未匹配点与一个匹配对对应起来。为便于分析，我们引入两个权重为 U的"虚拟"点（$-\infty$，0）和（$-\infty$，1），并将它们视为在输入序列揭晓之前就已匹配。假设一个新点 p到达区域 R。我们将 p 映射到构成区域 R边界的最新线段。图2展示了这种映射。

引理 4 每个类型为 i 的点都会被映射到类型至少为 i 的线段上。

证明 为导出矛盾，假设一个类型为 i 的点 p到达区域 R，并被映射到类型至多为 $i-1$ 的 p_R，q_R。

不失一般性，假设 p_R 在 q_R 之后到达。根据算法的定义，在 p_R 出现时，R 中至少有 $2^{k-i+1}-1$ 个未匹配点（否则，p_R 就不会与 q_R 匹配）。当 p 出现时，这些点仍未匹配（否则，R 就应该被划分，而 p 就应该被映射到其他线段）。因此，当 p 出现时，算法可以将它与平分这些未匹配点的点匹配，并且所得线段的两侧各至少有 $(2^{k-i+1}-2)/2=2^{k-i}-1$ 个点。这与算法让 p 保持未匹配的事实相矛盾。□

引理 5 设 s 为两个匹配点之间的任意线段。对于任意

i，至多有 $2^{k-i+2}-2$ 个类型为 i 的未匹配点映射到 s。

证明 为了导出矛盾，假设至少有 $2^{k-i+2}-1$ 个类型i 的点被映射到 s。那么，在通过延伸 s所形成的凸区域 R 中，至少必须有 $\lceil(2^{k-i+2}-1)/2\rceil = 2^{k-i+1}$ 个类型 i 的点。当这些点中的最后一个点（记为 p）到达时，它可以与平分其余点的点 q相匹配；在pq的每一侧将至少有$(2^{k-i+1}-2)/2 = 2^{k-i}-1$个点。由于 pq 属于类型 ~~i~~，该算法必定已将 p 与 q匹配，这与 p 和 q 均为未匹配点且被映射到 s的事实相矛盾。□

引理 6 对于 $U \geq 16$，映射到类型 j 线段上的未匹配点的总权重至多为 $a_{j+1}2^{k-j+3}$。

证明 注意，类型为 i 的点的权重至多为 $a_{i+1} = r^{i+1}$。因此，根据引理 4 和引理 5，映射到类型为 j 的线段上的未匹配点的总权重至多为

$$\sum_{i=0}^{j} r^{i+1}2^{k-i+2} = 2^{k+2}r\sum_{i=0}^{j}\left(\frac{r}{2}\right)^{i} = 2^{k+2}r\frac{(r/2)^{j+1}-1}{r/2-1} \leq a_{j+1}2^{k-j+3}.$$

[1]这里，我们假设 $r \geq 4$，这对 $U \geq 16$ 成立，因为 $16^{\lceil\sqrt{\log 16}\rceil} = 16^{2} = 4$。□

定理 7 对于足够大的 U，用于受限 OWNM 问题的确定性在线算法 WAM 的竞争比为

$$\Omega\left(2^{-2\sqrt{\log U}}\right).$$

证明 对于 WAM 产生的每个匹配对 pq，考虑由 p、q 以及映射到它们的未匹配点构成的点集。根据引理 6，如果 pq 的类型为 j，则该匹配对的权重与此集合中所有点的权重之比至少为 $\frac{a_j}{2a_j+a_{j+1}2^{k-j+3}} \geq \frac{1}{r2^{k+4}}$。

根据 $k = \lceil\sqrt{\log U}\rceil$ 和 $r = U^{1/k}$ 在第 5.1 节中的取值，可得 $r = 2^{(\log U)/k} \leq 2^{k}$。因此，由于算法 WAM 保证每个未匹配点都映射到某个匹配对，WAM 的竞争比至少为 $2^{-(2k+4)}$。□

6 随机化算法

在讨论完确定性情形后，我们将研究使用随机化时，期望竞争比能提高多少。

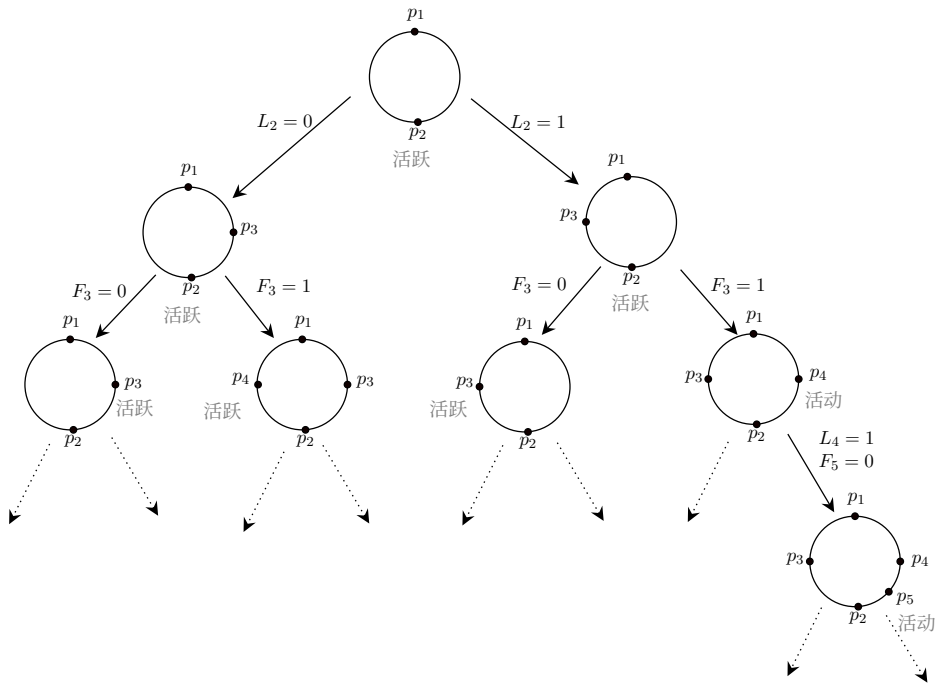


图 3：对手在随机变量 L_i 和 F_i 的引导下，展示随机输入最初几步构造的示意图。

6.1 否定性结果

根据 Yao 极小极大原理 [34], ，要证明随机化算法的竞争比至多为 γ，只需给出一个随机输入，使得任何确定性算法在该输入上的竞争比至多为 γ。我们将构造一个随机无权输入，类似于 Lavasani 和 Pankratov [24] 在建议模型中使用的输入，具体如下。

输入点都位于一个圆上；图 3 展示了对手构造过程最初几步的示意图。对于圆上的点 p，将由 p 界定的顺时针弧和逆时针弧分别称为 p 的左弧和右弧。起初，对手在圆上的任意一对对径点处给出 p_1 和 p_2，由此形成两段弧。对手将 p_2 设为当前活动点。对于其余点，对手使用两个由参数为 $1/2$ 的独立同分布伯努利随机指示变量构成的序列 L_1，……，L_{2n} 和 F_1，……，F_{2n} 来指导决策。如果 p_i 是当前活动点，L_i 决定 p_{i+1} 的位置。如果

L_i 为 1，对手就在 p_i 左弧的中点处给出 p_{i+1}；该弧由 p_i 和其左侧最近的点构成。如果 L_i 为 0，对手就在 p_i 右弧的中点处给出该点。给定 p_i 是一个活动点，若 F_{i+1} 为 1，点 p_{i+1} 就成为我们所称的跳过点。在这种情况下，对手将 p_{i+2} 设为下一个活动点，并在 p_i 的另一段弧的中点处给出该点（例如，如果 p_{i+1} 位于 p_i 的左弧上，那么 p_{i+2} 就在 p_i 的右弧上给出）。如果 F_{i+1} 为 0，对手就将 p_{i+1} 设为新的活动点。对手以新的活动点继续这一过程，直到生成 $2n$ 个点。

定理 8 没有任何随机在线算法能够以期望意义达到优于 $16/17$ 的竞争比。

证明 如本小节开头所述，根据 Yao 极小极大原理 [34], ，只需研究确定性算法在上述构造的随机输入上的行为，图 3 展示了该输入。现在固定一个确定性算法 ALG。由 ALG 匹配的点所形成的线段将圆划分为若干凸区域。如果一个未匹配点位于一个不会有新点到达的区域中，它就无法再被匹配，我们称之为孤立 点。假设 ALG 在到达时匹配了 p_i ，令 X_i 为指示随机变量：当 p_i 是活动点，且匹配它会导致至少一个点（可能是先前的点，也可能是点 p_{i+1}）变为孤立点时，该变量取值为 1。我们的目标是证明孤立点数量的下界，或者等价地，证明线段数量的上界。

我们用其他随机变量来表示 X_i。令 A_i 为指示随机变量，当 p_i 变为活动点时，该变量取值为 1。对于 $i \geq 3$，当且仅当 p_{i-1} 是活动点且 F_i 为 1 时， p_i 是跳过点。因此，

$$A_i = 1 - A_{i-1}F_i \geq 1 - F_i. \tag{1}$$

假设 p_i 变为一个活动点，并在到达凸区域 R时与 ALG 匹配，从而将 R 分割为 R_L 和 R_R，分别包含 p_i的左弧和右弧。若 R_L （或 R_R）不包含任何未匹配点，则称其为空。我们有以下四种情况：

- R_L 和 R_R 均为空：若 p_{i+1} 是跳过点，则它会变为孤立点。因此，$X_i \geq A_iF_{i+1}$;
- R_L 和 R_R 均非空：假设 p_{i+1} 位于 R_L 中并成为活动点，则 R_R 中的未匹配点会变成孤立点。 p_{i+1} 位于 R_R 中的情形类似。因此， $X_i \geq A_i(1 - F_{i+1})$;

- R_L 为空且 R_R 非空：如果 p_{i+1} 出现在 $p_i(L_i = 1)$的左弧上，则至少可以保证有一个孤立点。事实上，如果p_{i+1} 是一个跳过点，由于 R_L 为空，它会变成孤立点；如果它是新的活动点，则 R_R 中的未匹配点会变成孤立点。因此，$X_i \geq A_i L_i$;

- R_L 非空且 R_R 为空：对称地，$X_i \geq A_i(1 - L_i)$。

若 $i = 2n$，则不存在 p_{i+1}；如果 R_L 或R_R 非空，进行匹配 p_i 会使点成为孤立点。由于我们关注的是渐近竞争比，因此忽略这一边界情形。综上可得

$$X_i \geq \begin{cases} A_i F_{i+1}, & \text{if } R_L \text{ and } R_R \text{ are empty} \\ A_i L_i, & \text{if } R_L \text{ is empty and } R_R \text{ is non-empty} \\ A_i(1 - L_i), & \text{if } R_L \text{ is non-empty and } R_R \text{ is empty} \\ A_i(1 - F_{i+1}), & \text{if } R_L \text{ and } R_R \text{ are non-empty} \end{cases} \tag{2}$$

类似于（2）定义 Y_i，但将 A_i 替换为 $1 - F_i$，即

$$Y_i = \begin{cases} (1 - F_i)F_{i+1}, & \text{if } R_L \text{ and } R_R \text{ are empty} \\ (1 - F_i)L_i, & \text{if } R_L \text{ is empty and } R_R \text{ is non-empty} \\ (1 - F_i)(1 - L_i), & \text{if } R_L \text{ is non-empty and } R_R \text{ is empty} \\ (1 - F_i)(1 - F_{i+1}), & \text{if } R_L \text{ and } R_R \text{ are non-empty} \end{cases} \tag{3}$$

于是，（1）推出 $Y_i \leq X_i$。

现在，令 M 为一个随机变量，表示 ALG所生成的匹配的大小；对于每个 $1 \leq i \leq M$，令 T_i 为ALG进行第 i^{th} 次匹配时的步骤编号。注意，T_i 和 M 完全由序列 L_i 和 F_i（对于 $1 \leq i \leq 2n$）以及算法 ALG确定。因此，该算法保证至少有 $\sum_{i=1}^{M} X_{T_i}$ 个未匹配点。由于每条线段使用两个点，我们有 $\sum_{i=1}^{M}(X_{T_i} + 2) \leq 2n$。因此，根据 $Y_i \leq X_i$，有

$$2n \geq \sum_{i=1}^{M}(Y_{T_i} + 2) \geq \sum_{i=1}^{\lfloor \frac{M}{2} \rfloor}(Y_{T_{2i-1}} + Y_{T_{2i}} + 4) \geq \sum_{i=1}^{\lfloor \frac{M}{2} \rfloor}(Y_{T_{2i}} + 4). \tag{4}$$

定义另一个辅助随机变量序列 Z_1、Z_2、.....如下：

$$Z_i = \begin{cases} Y_{T_{2i}} + 4, & 1 \leq i \leq \lfloor \frac{M}{2} \rfloor, \\ Y' + 4, & i > \lfloor \frac{M}{2} \rfloor, \end{cases}$$

其中， Y' 是参数为 $1/4$的独立同分布伯努利随机变量。根据 Y_i的定义，由于 $Y_{T_{2i}}$ 和 $Y_{T_{2(i+1)}}$ 取决于互不相交的变量集合，$Y_{T_{2i}}$ 是参数为 $1/4$的独立同分布伯努利随机变量。因此，对所有 $i \geq 1$ ，序列 Z_i 均为独立同分布序列，其期望为 $17/4$。令随机变量 M' 为满足 $\sum_{i=1}^{M'} Z_i \leq 2n$的最大整数。注意，根据 Z_i的定义和(4)，我们有 $\sum_{i=1}^{\lfloor \frac{M}{2} \rfloor} Z_i \leq 2n$，因此 $M' \geq \lfloor \frac{M}{2} \rfloor$。由于 Z_i是期望为 $17/4$的独立同分布随机变量，根据更新定理，我们有 $\mathbb{E}[M'] = 2n/(17/4) = 8n/17$。因此， $8n/17 \geq \mathbb{E}[\lfloor \frac{M}{2} \rfloor] \geq \mathbb{E}[M]/2 - 1$，这意味着 $\mathbb{E}[M] \leq 16n/17+2$。由此，确定性算法 ALG 在期望意义下可以匹配 $2\,\mathbb{E}[M] \leq 2(16n/17 + 2)$个点。因此，期望竞争比为 $\frac{2\,\mathbb{E}[M]}{2n}$ ；当 $n \to \infty$时，其下确界为 $16/17$。□

6.2 PositiveResult：Tree-Guided-MatchingAlgorithm

我们提出一种名为“Tree-Guided-Matching”的随机化算法（TGM），无论点的权重如何，该算法都能提供以下统一保证：每个点出现在匹配中的概率至少为 $1/3$。该算法 TGM利用二叉树引导其匹配决策。二叉树的结构由输入点的相对位置决定，树中的每个节点都对应一个这样的点。直观地说，随着二叉树不断生长，它会在线细化平面向凸区域的划分，使得每个区域都有一个负责它的在线点。见图4。我们强调，这些区域以及为表示它们而创建的树并不代表任何匹配。它们的作用是确保一旦进行匹配，匹配就不会相交；同时，它们也指出了我们将尝试以一定概率与之匹配的点。

最初， p_1 设为树的根，并将 p_2 设为 p_1的子节点。稍微滥用一下记号，我们也用 pq 表示由点 p 和 q确定的直线。令 R_1 和 R_2 表示由 p_1p_2生成的两个半空间所对应的区域。我们再次强调，这只是定义树和区域的起始状态，并不表示 p_1 和 p_2 已匹配。是否会发生匹配，将在下文讨论。令p_2 同时负责 R_1 和 R_2。一般而言，当 p_i 到达区域 R ，且该区域由p_j 负责时（当然， $j < i$），将 p_i 设为p_j在二叉树中的子节点。直线 p_ip_j 将区域 R 分为两个子区域R' 和 R''，令 p_i 同时负责这两个子区域。此时，p_j 对 R 的责任随之解除，因为区域 R 已细分为 R' 和 R''。

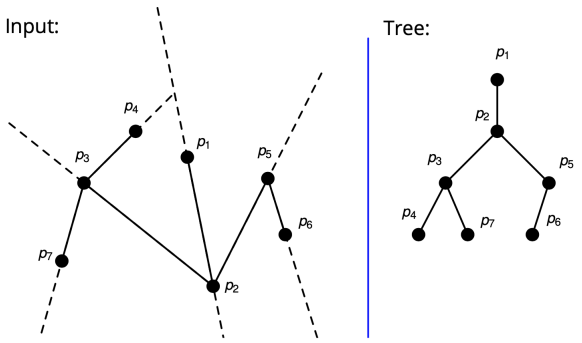


图4：左侧展示输入以及 TGM 如何划分平面，右侧展示其根据输入构建的树。基于这棵树，它随机地将节点与其父节点进行匹配，使得每个节点到达时，以 1/3的概率与其父节点匹配。

请注意，这意味着树中的每个节点至多有两个子节点。接下来，我们描述 TGM 如何选择匹配点。开始时，TGM 以概率 1/3将 p_2 与 p_1 匹配。此后，当 p_i到达时，令 p_j 为它在树中的父节点。如果p_j 未匹配且 p_i 是它的第一个子节点，则以概率 1/2将 p_i 与 p_j 匹配。如果p_j 未匹配且 p_i 是它的第二个子节点，则确定性地将 p_i 与 p_j 匹配。请注意， TGM 只尝试将在线点与其在树中的父节点匹配。

定理 9 每个点无论其权重如何，均至少以 1/3的概率被随机化算法 TGM 匹配。因此， TGM 达到至少为 1/3的严格竞争比。

证明 注意，由于 TGM 在二叉树中只将子节点与其父节点进行匹配，因此该匹配是不相交的。事实上，根据我们的树构造，并回顾点处于一般位置这一点，每个子节点都是位于一个凸区域内部 的点，其父节点负责该区域，且父节点位于该区域的边界上。因此，由匹配形成的线段不会与任何已有线段相交。

接下来，我们证明 TGM 的性能如所声称。根据 TGM 的定义， p_1 以概率 1/3 与 p_2 匹配。我们将证明，每个 p_i，$i \geq 2$ 在到达时恰好以概率 1/3 与其父节点匹配，这就意味着所要证明的结论。为此，我们采用归纳法。对于 p_2，基础情形成立。设 p 是当前到达的点， q 是它的

父节点。我们分两种情况讨论。

- 如果 p是 q的第一个子节点，那么根据归纳假设，此时 q 未匹配的概率为 $2/3$。因此，根据TGM，p 以概率 $(2/3)\cdot(1/2) = 1/3$与 q匹配。

- 如果 p 是 q的第二个子节点，那么此时 q 未匹配的概率为 $1 - 1/3 - 1/3 = 1/3$。根据 TGM，p 以概率 $(1/3)\cdot 1 = 1/3$与 q匹配。

□

7 可撤销接受

本节中，我们考虑可撤销设置。当新点 p到达时，算法可以选择在处理 p 之前，从匹配中移除一条已有的边。移除已有边的决定不可撤销。这样做的好处是，被移除边的端点以及边另一侧可能存在的点（不过我们的正向结果并未利用这一可能性）都可以成为与 p匹配的候选点，前提是满足非交叉约束。

7.1 否定性结果

Bose等人 [11] 证明，在无权版本中，不使用撤销的确定性贪心算法可以达到 $2/3$ 的竞争比。本节将证明，即使允许撤销，这一结果仍是最优的。

定理 10 即使在无权版本中，也不存在能够通过撤销达到优于 $2/3$的竞争比的确定性算法。

证明 固定一个确定性算法 ALG、一个任意大的 n，以及平面中的一个圆。对手逐个在圆上给出至少 $2n$ 个点，每个点的权重均为 **1**，并让 ALG 将它们匹配成对。因此，所有匹配都是圆上的弦。我们保持以下不变量：圆上始终有一个活动区域；对于每一对匹配点，始终恰有一个与之关联的唯一未匹配点。这样的点不在活动区域内，且在其所在区域中，没有其他与匹配对关联的点。此外，与该点关联的匹配对所对应的弦与其所在区域相邻。还可能有其他未 与匹配对关联的未匹配点。

初始时，整个圆都是活动区域。一个阶段包括对手在活动区域内的圆上逐个给出点，直到 ALG要么匹配一对点，要么撤销一项匹配（或者已给出的点总数达到 $2n$ ）。

我们首先考虑不存在撤销的情况。因此，当前点 p 只需与圆上的一个点 q 匹配。弦 pq 将活动区域分成两个子区域 R_1 和 R_2。不失一般性，假设 R_1 中未匹配点的数量至少与 R_2 中一样多。如果两个区域都没有点，我们就将一个点 p' 赋给 R_1。现在， R_1 中有一个未匹配点，其中一个与匹配对相关联；R_2 成为活动区域，阶段结束，并且不变量显然成立。

现在假设 ALG 撤销一个匹配。移除这个匹配会移除圆的一条弦，从而将两个区域合并成一个新的凸区域。注意，其中一个区域是活动区域，因此该区域中没有与匹配对相关联的未匹配点。弦只与两个区域相邻，因此另一个区域必定包含相关联的点。因此，撤销后，合并后的区域中没有与匹配对相关联的点。

如果 p 未被匹配，则这个联合区域成为活动区域，且该阶段结束。

如果 p 已匹配，则它可能与被撤销匹配中的某个点匹配。不过，被撤销匹配中的点至少有一个（ q）现在处于未匹配状态，因为 p 只与一个点匹配。由于 q 刚刚是某个匹配的一部分，它不可能是与某个匹配相关联的未匹配点，因此我们可以将 q 与刚刚建立的新匹配相关联。不包含 q 的那个由匹配形成的子区域成为活动区域，当前阶段随之结束。

归纳地，每个阶段结束后这些不变量都成立，而与每个匹配对相关联的未匹配点确保最多有 $2/3$ 个点被匹配。尽管点的数量可能是奇数，这仍给出了竞争比的渐近下界 $2/3$。□

7.2 PositiveResult：Big-Improvement-Match

我们提出一种带撤销的确定性算法，称为“大改进匹配”（BIM）。即使点的权重不受限制，该算法的严格竞争比仍约为0.2862。这表明，虽然撤销并不能提高无权

版本中的竞争比，但它为我们提供了一种具有常数竞争比的算法，而这是不带撤销的确定性算法无法达到的。

在下文中，我们略微滥用记号，用pq 表示由两个点 p 和 q确定的直线。 BIM 维护将欧几里得空间划分为若干区域的一个划分。划分中的每个区域都被指派当前匹配中的一条边，由该边负责该区域。每条边最多可以负责两个区域。 BIM 首先匹配前两个点 p_1 和 p_2 ，而不考虑它们的权重，并用p_1p_2将平面划分为两个半平面。 BIM随后指派 p_1p_2 负责这两个半平面区域。接下来，考虑一个到达已有区域 R的新点 p_i （对于 $i \geq 3$）。假设$p_jp_{j'}$ 是负责 R的边。如果 R中至少有一个未匹配点，则 BIM 将 p_i 与 R 中权重最高的未匹配点 p_k 进行匹配。然后$p_jp_{j'}$ 不再负责 R，并且区域 R 被 p_ip_k 划分为两个新区域。两个新区域的责任都被指派给 p_ip_k。如果 p_i是 R中唯一的点，则 BIM 按如下方式考虑撤销匹配（p_j，$p_{j'}$）。不失一般性，假设 $w(p_j) \leq w(p_{j'})$。我们引入一个参数 r，稍后将选择它以优化竞争比。如果 $w(p_i) < rw(p_{j'})$，则 BIM 使 p_i 保持未匹配。否则， BIM 移除匹配（p_j，$p_{j'}$），并将 p_i 与$p_{j'}$匹配。如果R 是 p_i 到达时$p_jp_{j'}$ 所负责的唯一区域，则 R被$p_ip_{j'}$划分为两个区域，并且$p_ip_{j'}$ 负责这两个新区域。（从 p_i 看，位于$p_jp_{j'}$ 另一侧的区域仍保持其边界，即使（p_j，$p_{j'}$）已不再属于该匹配。）否则，当 p_i 到达时，$p_jp_{j'}$ 除了负责 R 之外还负责 R'。在这种情况下，在移除匹配（p_j，$p_{j'}$）之后，区域 R 和 R' 被合并为区域 $R'' = R \cup R'$，并且 R'' 被$p_ip_{j'}$ 划分为两个区域， BIM使$p_ip_{j'}$ 负责这两个新区域。

命题 11 关于 BIM，有以下观察：

1. 所有负责的边都由两个当前已匹配的点定义。2. 每条边最多负责两个区域。3. 所有区域都是凸的。4. 当匹配边 (p_j,$p_{j'}$) 因点p_i 到达区域 R而被替换时，边 (p_i,$p_{j'}$) 完全位于 R 中。

证明 (1) 成立，因为一条边只有在其端点被匹配后才会成为负责边。当另一条边开始负责某个区域时，

原来的边便不再负责该区域。(2) 成立，因为一条边所负责的区域仅有两个：它的端点被匹配时产生的两个区域。当边所负责的某个区域中的两个点被匹配后，该边便不再负责该区域；但如果此前它一直负责两个区域，则仍会负责其中一个区域。(3) 可归纳地得出，因为用一条线段将两个凸区域分隔开会产生两个凸区域。此外，当BIM 移除一条边时，该边是其负责的两个区域中任一区域内最后创建的匹配。(4) 由(3)得出。□

定理12 对于带撤销和任意权重的 OWNM 问题，BIM 在 $r \in (1, \sqrt{2}]$ 下的严格竞争比至少为 $\min\left\{\frac{r^2-1}{r^3}, \frac{1}{1+2r}\right\}$。

证明 对于一条边所负责的每个区域，我们考虑该区域中未匹配点的总权重。这些点分为两类：一类在执行过程中的某个时刻曾被匹配，但由于撤销而变为未匹配；另一类在整个算法执行期间从未被匹配。

考虑所有已创建边的任意子序列 $\langle e_1, \ldots, e_k \rangle$，其中 e_1 是在某个区域中第二个未匹配点到达时创建的，其余可能的边则通过撤销创建，即 e_i 导致 e_{i-1} 因 $2 \le i \le k$而被撤销，且 e_k 属于 BIM 的最终匹配。设 $e_j = (p_{i_j}, p_{i_{j+1}})$ 且 $w(p_{i_j}) \le w(p_{i_{j+1}})$，从而 $e_{j+1} = (p_{i_{j+1}}, p_{i_{j+2}})$。因此，对于 $3 \le j \le k+1$，p_{i_j}在 $p_{i_{j-1}}$ 之后到达。 BIM 曾经匹配过的每一对都包含在某条这样的边序列中。点 $p_{i_1}, \ldots, p_{i_{k-1}}$ 可能是 e_k 所负责的某个区域中的未匹配点。

我们界定曾经位于由某个 e_j 负责的区域中的点的总权重。我们将其分开，首先考虑在执行过程中曾经匹配、但后来不再匹配的点（即 p_{i_j}），再考虑之后从未匹配过的点。

设 $\alpha = w(p_{i_k})$，则对于每个 $2 \le j \le k$，$w(p_{i_j})$ 至多为 $\alpha r^{-(k-j)}$ and$w(p_{i_1}) \le w(p_{i_2}) \le \alpha r^{-(k-2)}$。设 $\beta = w(p_{i_{k+1}})$。这些点的总权重

在此序列中为

$$\begin{aligned}\sum_{j=1}^{k+1} w(p_{i_j}) &= w(p_{i_1}) + \sum_{j=2}^{k} w(p_{i_j}) + w(p_{i_{k+1}}) \\ &\le \alpha r^{-(k-2)} + \alpha\left(\frac{r}{r-1}\right)(1 - r^{-(k-1)}) + \beta \\ &= \alpha\left(r^{-(k-1)}\frac{r(r-2)}{r-1} + \frac{r}{r-1}\right) + \beta.\end{aligned}$$

现在，我们考虑其他从未匹配过、但曾经处于某个由 e_j 负责的区域中的点。在创建 e_1 之后且在 e_2之前，第一个点 q_1 可能会到达某个由e_1 负责的区域。由于 q_1 没有与 p_{i_2}匹配， $w(q_1) < rw(p_{i_2})$。（注意，到达该区域的第二个点随后将与 q_1匹配，从而划分该区域；这些子区域将不再被视为 e_k 最终负责的区域的一部分。）现在，假设另一个点 q_2在为某个 $2 \le j < k$ 创建e_j 和e_{j+1} 之间到达，并且在某个由 e_k 负责的区域中保持未匹配状态。那么， q_2 和 $p_{i_{j+2}}$ 都不可能与 $p_{i_{j-1}}$ 处于同一区域，否则其中一个就会与 $p_{i_{j-1}}$ 匹配（或者 $p_{i_{j-1}}$ 已经匹配，且该区域已被划分）。根据命题 11.2, e_i 最多负责两个区域，因此$p_{i_{j+2}}$ 会在 q_2仍未匹配时到达与其相同的区域。这就产生了矛盾，因为 Bim 会将它们匹配起来。因此，除 q_1之外，在由 e_k 负责的区域中唯一一个从未匹配过的点 q_2，是在 e_k 之后到达的，并且 $w(q_2) < rw(p_{i_{k+1}})$。

那么，对于 $k \ge 2$，由 e_k 负责的区域中的未匹配点的总权重至多为

$$\left(r^{-(k-3)} + r^{-(k-1)}\frac{r(r-2)}{r-1} + \frac{r}{r-1}\right)\alpha + (1+r)\beta.$$

如果 $r \le \sqrt{2}$，那么 $r^{-(k-3)} + r^{-(k-1)}\frac{r(r-2)}{r-1}$ 至多为零，因此当 $k \ge 2$时，我们可以将总权重界定为： $(\frac{r}{r-1})\alpha + (1+r)\beta$. 因此，对于 $k \ge 2$ ，序列 $\langle p_{i_1}, p_{i_2}, \ldots, p_{i_{k+1}}\rangle$ 中匹配点的权重与所有关联到该序列的点的总权重之比至少为$\frac{\alpha+\beta}{(\frac{r}{r-1})\alpha+(1+r)\beta}$。由于 $\frac{r}{r-1} > 1 + r$, –对于 $1 < r \le \sqrt{2}$，当 β取最小值时，该比率达到最小，而这发生在 $\beta = r\alpha$。因此， $k \ge 2$ 的竞争比至少为 $(1+r)/(\frac{r}{r-1} + r(1+r)) = \frac{r^2-1}{r^3}$。 ——

现在，考虑 $k = 1$ 的情况，并令 α 和 β 的含义与上文相同。那么，序列 $\langle e_{i_1}, e_{i_2}, \ldots\ldots, e_{i_k}\rangle$ 仅由一条边组成。因此，

匹配点的权重为 $\alpha+\beta$，算法执行结束时可能有两个未匹配点 q_1 和 q_2 计入这条边。我们有 $w(q_i) < r\beta$，因此，对于 $k = \frac{1}{\alpha+\beta}$ 的情况，匹配点的权重与序列所关联的所有点的总权重之比至少为 $\alpha+\frac{1}{1+2r}(\ )\beta$。注意，当 β 趋于无穷大时，该比率达到最小值，并变为 $1/(1+2r)$。

取上述两种情形中较差的比率，即可证明定理的结论。□

推论13 选择参数 BIM、 r^*，使其为方程 $\frac{1}{1+2r} = \frac{r^2-1}{r^3}$ 的正解，约为 1.2470，可得竞争比为 $\frac{1}{1+2r^*}$，至少为0.2862。

证明 令定理12中取最小值的两项相等，并求解 r，即可得到 r^* 的值，从而得出竞争比的下界。

为证明该算法的这一结果是紧的，考虑以下输入：权重为 α 的 p_1 到达单位球面的北极，接着权重为 $\beta \geq \alpha$ 的 p_2 到达单位球面的南极，然后权重为 $r^*\beta - \varepsilon$ 的 p_3到达单位球面的西极，最后权重为 $r^*\beta - \varepsilon$ 的 p_4到达单位球面的东极。算法最终会将 p_1 与 p_2匹配，而 p_3 和 p_4 未匹配。因此，在这样的实例中，该算法的竞争比为$(\alpha+\beta)/(\alpha+\beta+2r^*\beta-2\varepsilon)$。令 β 趋于无穷大且 ε趋于零，可知该算法在严格意义上无法保证优于 $1/(1+2r^*)$的竞争比。 □

8 共线点

本节研究所有点都位于一条直线上的不相交匹配问题。首先，我们将证明，与一般情形相反，即使点无权，单独使用随机化或撤销也无法获得任何非平凡竞争比。换言之，这表明我们前面介绍的随机化算法和撤销算法的结果依赖于点处于一般位置。

定理14 当点以任意位置到达时，即使这些点无权且都位于一条直线上，

(1) 没有随机化算法能够达到优于 $2/n$ 的竞争比，且

(2) 没有带撤销功能的确定性算法能够达到优于 $1/n$ 的竞争比。

证明(1) 我们构造一个随机输入，界定任意确定性算法在该输入上的竞争比，并利用 Yao 极小极大原理证明该定理。首先，将 p_1 和 p_2 放置在直线上的任意位置，然后将 p_3 放在它们之间。对于之后的每个点 p_i，随机选择相对于上一个点 p_{i-1} 左侧或右侧的线段，并将 p_i 放在线段中点。这个过程持续进行，直到直线上放置了总共 $2n$ 个点。

如果确定性算法匹配了 p_1 和 p_2，那么它能够匹配更多点的概率为零。如果它没有匹配 p_1 和 p_2，则会出现以下情况：每当算法匹配一个新到达的点时，未来所有点都从同一侧到达、从而阻止后续匹配的概率为1 2 。因此，该算法所创建匹配的期望大小至多为 $1 + \frac{1}{2}(1 + \frac{1}{2}(1 + \frac{1}{2}(1 + \cdots))) = 2$。然而，始终可以实现完美匹配，因此竞争比为 $\frac{2}{n}$ 。

（2）对手策略旨在确保算法在任何时刻至多维持一个匹配对。起初，对手将点放置在任意位置，直到算法在两个点之间形成匹配。匹配形成后，对手开始在该匹配对之间放置新点，迫使算法撤销该匹配，才能形成任何新的匹配。

算法撤销匹配后，对手恢复在任意位置放置点。然而，一旦算法创建另一个匹配，对手就再次在匹配对之间放置点，阻止算法取得任何进一步进展。这个循环持续进行：不存在匹配时，对手自由放置点；一旦形成匹配，对手就在匹配对之间放置点。这样可以确保算法始终只能维持一个匹配，从而得到$1/n$的竞争比。□

接下来，我们介绍一种带撤销的随机化算法，称为“随机撤销匹配”（RRM）。对于无权情形，该算法实现了至少为 0.5 的竞争比（见算法 1）。 RRM 维护将直线划分为若干区间的划分，并且仅在这些区间内匹配点。这些区间可以是开区间、半开区间或闭区间。初始时，整条直线是一个开区间。在算法执行过程中，始终满足以下

直线划分条件：

- 开区间和半开区间不包含匹配对。
- 每个半开区间恰好包含一个未匹配点，位于其闭边界上。
- 闭区间恰好包含两个点，二者均位于边界上并相互匹配。

这些条件在开始时成立，此时直线为一个开区间，并且在算法的每一步之后仍然有效。算法的决策确保这些条件在其整个执行过程中得以保持。

到达开区间的第一个点保持未匹配，划分不变。当第二个点到达开区间时，它与先前可用的点匹配。此操作将该开区间拆分为一个包含匹配点的闭区间，以及位于匹配线段两侧的两个新开区间。

当一个点到达闭区间时， RRM 撤销该匹配线段，将新点随机匹配到当前可用点之一，并将该区间划分为一个包含新匹配点的闭区间，以及一个以先前匹配点位于其闭边界上的半开区间。

最后，当一个点到达半开区间时，它会与其闭边界上的未匹配点进行匹配，将该区间分割为匹配点之间的闭区间和一个开放的空区间（见图 5）。

定理15 当点为无权点并在线上到达时，具有撤销权限的随机化算法 RRM 的竞争比率至少为 $1/2$。

证明 令 $f(n)$、$g(n)$ 和 $h(n)$ 分别表示当 n 个点到达闭区间、半开区间和开区间时匹配对的期望数量。对于 $n = 0$，不存在匹配对。根据 RRM的定义，当一个点到达开区间时，它保持未匹配状态；当一个点到达闭区间时，一个匹配对被撤销并创建一个新的匹配对，因此匹配对的数量不变。然而，当一个点到达半开区间时， RRM 会创建一个新的匹配。因此，我们得到初始条件： $f(0) = f(1) = g(0) = h(0) = h(1) = 0$

算法 1 $RandomRevokingMatching$

过程 $RandomRevokingMatching$

当收到一个新点 p_i **时**

令 I 为 p_i 到达的区间。

如果 I 是一个开区间（x, y）**则**

如果 I 包含另一个点p_j **则**

将 p_i 与p_j匹配。

如果 p_i 位于p_j 的左侧**则**

将 I划分为(x, p_i)、$[p_i, p_j]$ 和(p_j, y)。

else

将 I划分为(x,p_j)、$[p_j, p_i]$ 和(p_i, y)。

否则如果 I是一个闭区间 $[p_j, p_k]$ **则**

撤销$p_j p_k$。

将 r 均匀随机地设为 0 或 1。

如果 $r = 0$ **则**

将 p_i 与p_j匹配

将 I划分为 $[p_j, p_i]$ 和 $(p_i, p_k]$

else

将 p_i 与 p_k匹配

将 I划分为 $[p_j, p_i)$,和 $[p_i, p_k]$

否则如果 I是一个半开区间 $[p_j, x)$ **则**

将 p_i 与p_j匹配。

将 I分成 $[p_j, p_i]$ 和 (p_i, x)

否则如果 I 是一个半开区间 $(x, p_j]$ **则**

将 p_i 与p_j匹配。

将 I分成 (x, p_i) 和 $[p_i, p_j]$

以及 $g(1) = 1$。

如果有两个或更多点落在一个开区间内，则前两个点将进行匹配，从而把该区间划分为两个开区间和一个闭区间。当预计有 n 个点落在原来的开区间内时，新的区间将分别包含 k_1、k_2 和 $n - k_1 - k_2 - 2$ 个点，其中k_1, $k_2 \geq 0$ 和 $k_1 + k_2 \leq n - 2$。因此，对于 $n \geq 2$，我们有：

$$h(n) \geq 1 + \min_{\substack{k_1,k_2 \\ k_1,k_2 \geq 0 \\ k_1+k_2 \leq n-2}} \{h(k_1) + f(k_2) + h(n - k_1 - k_2 - 2)\}.$$

在闭区间内到达的第一个点会撤销已有的匹配

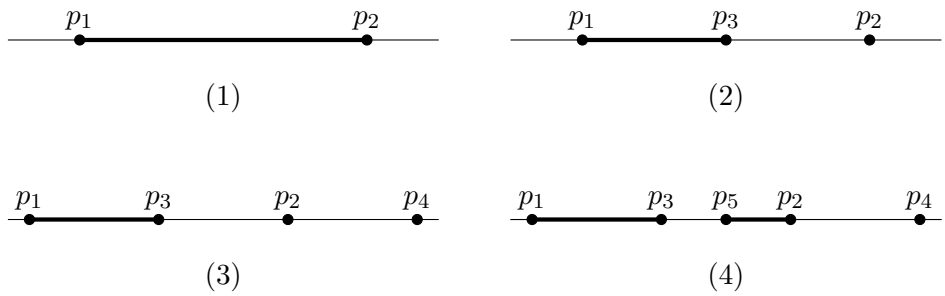


图5：（1）前两个点到达后，将直线划分为 $(-\infty,\ p_1)$、$[p_1, p_2]$ 和 $(p_2,\ \infty)$。（2）点 p_3 到达 $[p_1, p_2]$ 中。 RRM 撤销 p_1p_2，随机将 p_3 与 p_1 匹配，并将 $[p_1, p_2]$ 划分为 $[p_1, p_3]$ 和 $(p_3,\ p_2]$。（3）点 p_4 到达 $(p_2,\ \infty)$ 中，由于其到达的区间为空，因此保持未匹配状态。（4）点 p_5 到达 $(p_3,\ p_2]$ 中，并与 p_2 匹配，将区间分割为 $(p_3,\ p_5)$ 和 $[p_5, p_2]$。

线段并随机与其中一个点匹配，将区间划分为一个半开区间和一个闭区间。预计还会有 $n \geq 2$ 个点到达时，新的区间将分别接收 k 和 $n-k-1$ 个点，其中 $0 \leq k \leq n-1$。因此，对于 $n \geq 2$，有：

$$f(n) \geq \min_{0 \leq k \leq n-1} \left\{ \frac{1}{2} \left[f(k) + g(n-k-1) \right] + \frac{1}{2} \left[f(n-k-1) + g(k) \right] \right\}.$$

最后，当第一个点到达一个半开区间时，它会与唯一可用的点匹配，将区间划分为一个闭区间和一个开区间。预计还会有 $n \geq 2$ 个点到达时，新的区间将分别接收 k 和 $n-k-1$ 个点，其中 $0 \leq k \leq n-1$。因此，对于 $n \geq 2$，有：

$$g(n) \geq 1 + \min_{0 \leq k \leq n-1} \left\{ h(k) + f(n-k-1) \right\}.$$

接下来，我们用归纳法证明 $n \geq 0$ 的以下不等式：

$$f(n) \geq \frac{n-1}{4}, \qquad g(n) \geq \frac{n+1}{4}, \qquad h(n) \geq \frac{n-1}{4}.$$

不难验证，对于 $n = 0$ 和 $n = 1$，这些不等式成立。对于归纳步骤，任取满足 $0 \leq k \leq n-1$、$k_1,\ k_2 \geq 0$ 的 $k,\ k_1,\ k_2$，并且

$k_1 + k_2 \le n-2$，我们有：

$$\begin{aligned} h(n) &\ge 1 + h(k_1) + f(k_2) + h(n-k_1-k_2-2) \\ &\ge 1 + \frac{k_1-1}{4} + \frac{k_2-1}{4} + \frac{n-k_1-k_2-3}{4} = \frac{n-1}{4}. \end{aligned}$$

对于 $f(n)$，我们得到：

$$\begin{aligned} f(n) &\ge \frac{1}{2}\left(f(k) + g(n-k-1)\right) + \frac{1}{2}\left(f(n-k-1) + g(k)\right) \\ &\ge \frac{1}{2}\left(\frac{k-1}{4} + \frac{n-k}{4} + \frac{n-k-2}{4} + \frac{k+1}{4}\right) = \frac{n-1}{4}. \end{aligned}$$

对于 $g(n)$，我们有：

$$g(n) \ge 1 + h(k) + f(n-k-1) \ge 1 + \frac{k-1}{4} + \frac{n-k-2}{4} = \frac{n+1}{4}.$$

因此，当 $2n$ 个点到达直线上时， RRM 的匹配对期望数量至少为 $h(2n) \ge \frac{2n-1}{4}$，其中最优值始终为 $\overline{n}$，这意味着 RRM 的竞争比为 $1/2$。□

9 带建议的算法

本节中，我们将考虑 OWNM 问题的建议复杂度，即要达到最优，足够的完美信息位数是多少。我们采用磁带建议模型，这是标准的建议模型。在该模型中，可信预言机按照预先约定的协议与在线算法协作。预言机可以提前访问整个输入序列。预言机通过在建议磁带上写入比特与在线算法通信。当一个输入项到达时，算法从磁带上读取若干建议比特，并根据迄今为止从磁带读取的建议以及迄今为止到达的输入项，为新输入项作出决策。我们关注的问题是，在最坏情况输入下，为使在线算法能够最优地解决问题，预言机与算法之间需要通信的比特数上界是多少。对于带建议的在线算法的介绍，感兴趣的读者可参阅综述 [12] 及其中的参考文献。

我们提出一种带建议的在线算法，称为“分割与匹配”（SAM），并证明其具有最优性。对于输入序列

长度为 $2n$ 的输入序列， SAM 使用一族 C_n 条建议串，其中 C_n 是第 n^{th} 个卡塔兰数。卡特兰数有许多种不同的定义。或许与此处最相关的是 $C_0 = 1$， $C_n = {}^{2}{}_{(2n-1)\ n+1}\ C_{n-1}$，因为它最清楚地展示了 $\overline{C_n}$ 的渐近增长方式。注意，$O(\log C_n) = O(n)$，因此 $O(\log \log C_n) = O(\log n)$。预言机使用埃利亚斯δ编码方案 [16]，对每条建议串进行编码，该方案需要 $\log C_n + O(\log \log C_n)$ 位。

SAM 预言机与算法共同维护一个将平面划分为凸区域的平面划分以及一个责任关系，其中一个点最多可被指定负责一个区域，且每个区域最多只能有一个负责点。每个已定义的区域最终接收到的点的总数均为偶数。任何尚未被划分为子区域的区域所包含的未匹配点均不超过一个。当新点 p到达区域 R时，如果 R 没有负责点，则将 p 指定为 R的负责点，并且此时 p 保持未匹配。否则，假设在 p到达时， q 是 R 的负责点。在这种情况下，移除 q 的责任，并通过将 R 细分为 R_1 和 R_2来细化平面划分，即由pq形成的 R 的子区域。如果对每个 $i \in \{1, 2\}$而言， R_i 中点的总数（包括未来的点，但不包括 p 和 q）均为偶数，则将 p 与 q 进行匹配（我们将此事件称为“安全匹配”），并且没有任何点负责 R_1 或 R_2。否则， p 与 q 不进行匹配，并将 q 指定为 R_1的负责点，将 p 指定为 R_2的负责点。注意，当某个区域有负责点时，按照约定假定该点位于该区域内，尽管它也可以位于边界上。

为了在建议模型中实现上述过程， SAM 预言机（见算法 2）会创建一个长度为 $2n$ 的二进制串 D，其中 i^{th} 位表示 p_i 是否到达某个已为其分配了负责点的区域，而 p_i 和p_j 构成一个安全匹配。字符串 D 被编码到纸带上，并传递给 SAM。 SAM 算法（见算法 3）从纸带上读取 D 的编码（在在线点到达之前），从编码中恢复 D，然后利用 D 中的信息运行上述过程，创建安全匹配。

并非所有长度为 $2n$的二进制串都能构成有效的 D。因此，预言机和算法可以预先约定所有可能字符串 D的排列顺序，我们称之为建议族。随后，预言机在纸带上写下该排列中与给定输入的D 相对应的字符串的索引。下面的定理证明了

算法 2 *Split-And-Match* 预言机。

过程 *Split-And-Match-Oracle*
  $D \leftarrow [0]$
  使 p_1 负责平面
  对于 $i = 2$ 至 $2n$，**执行**
    令 R 为 p_i 到达的区域
    如果 R 有一个负责点p_j **则**
      撤销p_j的责任
      按p_ip_j将 R 划分为 R_1 和 R_2
      如果 R_L （以及 R_R）将包含偶数个点
总共**则**
        $D.append$(1)
      else
        使p_j 和 p_i 分别负责 R_1 和 R_2
        $D.append$(0)
    else
      令 p_i 负责 R
      $D.append$(0)
  将 D 传递给算法

该算法的正确性，并确立了所声称的建议复杂度上界。

定理 16 SAM 通过大小为 C_n 的建议族实现完美匹配。

证明 由于请求序列包含 $2n$ 个点，最终到达的点数为偶数。归纳地，被划分的区域在两个子区域中始终各有偶数个点，并且这些子区域均为凸区域。因此，到达某个区域的任何点都可以与已在该区域中、并负责该区域的点匹配。随着新点到达， SAM 的执行过程中只会出现两种区域:

- I型：该区域没有负责点，在创建时为空，且未来进入该区域的点数为偶数，并且

- II型：该区域有一个负责点，它是在该区域创建时区域内唯一的点，并且未来将有奇数个

算法 3 *Split-And-Match* 算法。

过程 *Split-And-Match*(D)
  当收到一个新点 p_i **时**
    令 R 为 p_i 到达的区域
    如果 R 有一个负责点p_j **则**
      撤销 p_j 的负责关系
      将 R 划分为 R_1 和 R_2，依据是$\overline{p_ip_j}$
      如果 $D[i] == 1$，**则**
        将 p_i 与p_j匹配
      else
        分别让p_j 和 p_i 负责 R_1 和 R_2
        让 p_i 保持未匹配状态
    else
      让 p_i 负责 R
      让 p_i 保持未匹配

点到达该区域。

我们通过归纳法（对区域内后续到达的点数进行归纳）证明，无论区域的类型如何，算法最终都会匹配区域内的所有点。对于 I 型区域，基础情形显然成立：后续到达的点数为 0，因此无需证明。对于 II 型区域，基础情形也很简单：一个点到达该区域后，按照算法，它将与负责点匹配（因为 R_1 和 R_2 为空）。

对于归纳步骤，考虑一个 I 型区域 R，并假设将有 $2k$个点到达该区域。第一个到达该区域的点将成为该区域的负责点，从而使其类型变为 II 型。之后还有 $2k-1$个点将到达该区域，根据对 II 型区域应用归纳假设，即可得出结论。现在，考虑一个 II 型区域R，并假设有 $2k-1$ 个点到达该区域。令 q 为 R的负责点，令 p为第一个到达 R的点。注意，pq 将 R 划分为 R_1 和 R_2，并且总共有偶数个点将到达 R_1 和 R_2。有两种可能的情形。情形 1：如果 R_1 和 R_2均为 I 型，那么 p 与 q匹配。情形2：如果 R_1 和 R_2 均为 II 型，则不会形成匹配。在两种情形下，通过分别对 R_1 和 R_2 应用归纳法，都可以确立 R 的归纳步骤。

注意，在算法执行开始时（任何点到达之前），整个平面都是一个 I 型区域。因此，该算法的正确

性可通过将上述结论应用于该区域得出。

为了确立建议复杂度的界，请注意，根据该算法的定义，每当 $D[i]$ 为1时，SAM便匹配最近的点，否则不进行匹配。因此， D 含有相同数量的0和1，且 D 的任何前缀中1的个数都不超过0的个数。这使得 D 成为一个Dyck字，并且已知大小为 $2n$ [28]的Dyck字有 C_n 个。□

10 结论

我们引入了在线加权非交叉匹配问题的加权版本。我们证明了没有任何确定性算法能够保证该问题的常数竞争比。随后，我们探讨了克服这一限制的若干方法，并针对所考虑的每种情形给出了新的算法和界。特别地，我们给出了以下结果：当点的权重被限制在范围 $[1, U]$内时的确定性算法、对权重不加限制的随机化算法、允许撤销的确定性算法，以及带有建议的确定性算法。

仍有许多开放问题。特别是，我们的界并不紧，缩小任一设定中的差距都将是有意义的。研究离线文献中所考虑的该问题其他版本的在线设定也很有趣。例如，可以允许在线算法在给定预算内产生一定数量的交叉，或者可以考虑受理解RNA结构启发的 k-非交叉约束。本文考虑的是顶点加权版本，但也可以考虑该问题的边加权版本，其中边的权重既可以是抽象的，也可以与几何相关。

参考文献

[1] A. Karim Abu-Affash、Paz Carmi、Matthew J. Katz 和 Yohai Trab elsi。平面上的瓶颈非交叉匹配。计算几何，47(3)：447–457，2014年。[2] Oswin Aichholzer、Sergey Bereg、Adrian Dumitrescu、Alfredo Garc´ıa、Clemens Huemer、Ferran Hurtado、Mikio Kano、Alberto M´arquez、David Rappaport、Shakhar Smorodinsky、Diane Souvaine、Jorge Ur-

rutia 和 David R. Wood。相容几何匹配。计-算几何，42(6-7)：617–626，2009年。[3] Oswin Aichholzer、Sergio Cabello、Ruy Fabila-Monroy、David Flores-Penaloza、Thomas Hackl、Clemens Huemer、Ferran Hurtado 和 David R. Wood。几何图中的边移除与非交叉构型。离散数学与理论计算机科学，12(1)，2010年。[4]Oswin Aichholzer、Ruy Fabila-Monroy、Philipp Kindermann、Irene Parada、Rosna Paul、Daniel Perz、Patrick Schnider 和 Birgit Vogtenhuber。带交叉的完美匹配。*Algorithmica*，第1–20页，2023年。[5] Noga Alon、Sridhar Rajagopalan 和 Subhash Suri。平面上的长非交叉构型。载于第*9*届年度计算几何研讨会（*SoCG*），第257–263页，1993年。[6] Greg Aloupis、Esther M. Arkin、David Bremner、Erik D. Demaine、S´andor P. Fekete、Bahram Kouhestani 和 Joseph S. B. Mitchell。利用非交叉线段在平面中匹配区域。第十六届西班牙计算几何会议，2015年。[7] Drew Armstrong、Christian Stump 和 Hugh Thomas。非嵌套划分与非交叉划分之间的一致双射。*Transactionsof the American Mathematical Society*，365(8)：4121–4151，2013年。[8] Mikhail J. Atallah。平面上的一个匹配问题。*Journal ofComputer and System Sciences*，31(1)：63–70，1985年。[9] Vineet Bafna、S. Muthukrishnan 和 R. Ravi。计算RNA字符串之间的相似度。载于第*6*届年度组合模式匹配研讨会（*CPM*），计算机科学讲义第937卷，第1–16页。施普林格，1995年。[10] J´ozsef Balogh、Boris Pittel 和 Gelasio Salazar。随机标记圆上点的近完美非交叉调和匹配。离散数学与理论计算机科学论文集，AD：17–26，2005年。[11] Prosenjit Bose、Paz Carmi、Stephane Durocher、Shahin Kamali 和 Arezoo Sajadpour。在线点的非交叉匹配。载于第*32*届加拿大计算几何会议（*CCCG*），第233–239页，2020年。

[12] Joan Boyar、Lene M. Favrholdt、Christian Kudahl、Kim S. Larsen 和 Jesper W. Mikkelsen。带建议的在线算法：综述。*ACMComputing Surveys*，50(32):1–34，2017年。第19号文章。[13] William Y. C. Chen、Hillary S. W. Han 和 Christian M. Reidys。随机 k-非交叉 RNA 结构。美国国家科学院院刊，106(52):22061–22066，2009年。[14] Scott Cohen。在图像中寻找颜色和形状模式。博士论文，斯坦福大学，1999年。[15] Adrian Dumitrescu 和 William Steiger。平面中的一个匹配问题。离散数学，211(1-3):183–195，2000年。[16] Peter Elias。通用码字集与整数表示。*IEEE*信息论汇刊，21(2):194–203，1975年。[17] Andr´as Gy´arf´as。二部几何图非交叉子图的 Ramsey 和 Tur´an 型问题。载于布达佩斯大学科学年鉴，第54卷，47–56页，2011年。[18] Koki Hamada、Shuichi Miyazaki 和 Kazuya Okamoto。强稳定与最大弱稳定非交叉匹配。*Algorithmica*，83(9):2678–2696，2021年。[19] John Hershberger 和 Subhash Suri。半动态凸包算法的应用。*BIT*数值数学，32(2):249–267，1992年。[20]John Hershberger 和 Subhash Suri。印刷电路板中的高效逃逸布线。载于第*5*届国际算法与数据结构研讨会（*WADS*），计算机科学讲义第1272卷，462–471页。施普林格，1997年。[21] Shahin Kamali、Pooya Nikbakht 和 Arezoo Sajadpour。一种用于在线点非交叉匹配的随机化算法。载于第*34*届加拿大计算几何会议（*CCCG*），198–204页，2022年。[22] Ioannis Mantas、Marko Savi´c 和 Hendrik Schrezenmaier。完美非交叉匹配的新变体。离散应用数学，343:1–14，2024年。[23] Neeldhara Misra、Harshil Mittal 和 Saraswati Nanoti。多样化的非交叉匹配。载于第*34*届加拿大计算几何会议（*CCCG*），249–256页，2022年。

[24] Ali Mohammad Lavasani 和 Denis Pankratov。在线非交叉匹配的建议复杂度。计算几何，110:101943，2023年。[25] J´anos Pach、Natan Rubin 和 G´abor Tardos。平面点集决定许多两两相交线段。载于第*51*届 *ACM SIGACT* 年度计算理论研讨会（*STOC*），1158–1166页，2019年。[26] Rebecca Patrias、Oliver Pechenik 和 Jessica Striker。由非交叉划分构造不变多项式的网基。数学进展，408:108603，2022年。[27] David Rappaport。f等宽物体可见性匹配的紧确界。载于日本离散与计算几何会议，246–250页。施普林格，2002年。[28] Steven Roman。卡塔兰数导论。数学精要教材。Birkh¨äuser，2015年。[29] Suthee Ruangwises 和 Toshiya Itoh。稳定非交叉匹配。载于国际组合算法研讨会（*IWOCA*），计算机科学讲义第11638卷，405–416页。施普林格，2019年。[30] Arezoo Sajadpour。在线点的非交叉匹配。硕士论文，曼尼托巴大学，2021年。[31] Toshinori Sakai 和 Jorge Urrutia。点集上的重非交叉递增路径与匹配。载于第十四届西班牙计算几何会议，209–212页，2011年。[32] David Savitt。非交叉划分格中的多项式、蜿蜒线与路径。美国数学会汇刊，361(6):3083–3107，2009年。[33] Alexander A. Vladimirov。非交叉匹配。信息传输问题，49(1):54–57，2013年。[34] 姚期智。概率计算：迈向统一的复杂度度量。载于第*18*届计算机科学基础年度研讨会，222–227页，1977年。