arXiv:2304.01317v1 [cs.IT] 3 Apr 2023

U参数化受限编码通用框架

Daniella Bar-Lev、Adir Kobovich、Orian Leitersdorf 和 Eitan Yaakob i

摘要

受限编码是编码理论中的一个基础领域，旨在解决如何通过受约束信道实现高效通信。对于具有固定约束的信道（例如，传输消息中不得出现某个固定子串集合中的任何子串），已有通用的最优解；然而，人们越来越需要支持依赖于消息长度、用于描述某种属性的参数化约束（例如，不得出现连续 log(n) 个零）。已有若干研究通过迭代算法处理这类参数化约束，但为了确保收敛性，这些方法需要针对每种约束构造复杂方案，以保证单调递进。本文提出一种通用框架，通过一种新颖简单的迭代算法，处理任意参数化受约束信道问题。我们将该迭代算法的一次执行归约为有向无环图遍历，并由此证明了一个出人意料的结果：即使无需任何单调递进，也能保证收敛性，且具有高效的平均时间复杂度。

我们通过将这一通用框架应用于多种局部和全局信道约束，展示其有效性。我们首先探讨涉及不同长度非法子串的局部约束；通用构造的核心是迭代替换禁用窗口。我们将这一局部算法应用于最小周期性、最小汉明重量、局部近似平衡汉明重量，以及此前尚未解决的最小回文约束，并通过对通用算法进行简单调整，展示了最先进的结果。随后，我们继续探讨全局约束，展示所提出的构造在无重复编码、反向互补编码以及尚待解决的全局近似平衡编码问题上的有效性。针对反向互补约束，我们还解决了一个此前尚未处理的版本，该版本涉及重叠窗口。总体而言，所提出的框架能够以相当简便的方式生成最先进的构造，同时首次支持多种约束的同时集成。

1 引言

受限编码是一门基础领域，应用广泛，涵盖磁存储、光存储、闪存、DNA存储、无线通信和卫星通信 [Imm04,MRS01,Imm22]。这些应用面临各自独特的挑战和约束，数据编码必须满足这些要求。例如，磁存储要求相邻两个“1”之间的“0”的数量不超过给定上限，因为感测电路中的电压干扰可能导致错误 [Imm90]。在闪存中，使用受限码来缓解相邻字线与位线之间的单元间干扰 [QYS14, BB11]。近年来兴起的基于DNA的数据存储领域 [EZ17] （将数字信息存储为合成DNA链）同样会对编码数据施加约束，以提高分子稳定性，并减少合成和测序过程中的错误 [BLC+16, EZ17,HMG19,OAC+18]。总体而言，受限编码领域致力于开发高效的构造，将一般信息编码为满足信道约束的消息。

受限编码源于解决固定约束 ——即不依赖消息长度的约束。例如，局部 固定约束涉及一组固定字符串，这些字符串不能

作为子串出现在编码数据中（例如，序列“0011”不能出现在编码消息的任何位置）。过去，人们已对这类约束进行了充分研究，并已建立通用框架，可为任何此类固定局部约束 [MRS01]构造方案。该过程首先构造一个接受所有受约束词的确定性有限自动机，然后利用状态分裂算法 [ACH83] 设计编码器，使码率趋近信道容量。此外，还可通过佩龙–弗罗贝尼乌斯定理 [Min88] 计算信道容量；该定理将信道容量与自动机邻接矩阵的谱半径联系起来。

相反，对于能够描述某种一般性质且依赖于消息长度的参数化 约束，需求日益增长。例如，对于长度为 n 的消息，一种局部 参数化约束可以要求所有长度为 ℓ（n）的子串的汉明重量至少为 p（n）。目前，这类约束尚无统一的通用框架；现有方法往往针对特定应用量身定制，因此面对新的约束或试图同时实施多项约束时，通常需要付出大量努力。具体而言，对于对整体编码消息中的 ℓ（n）长度子串施加限制的局部 参数化约束，先前的研究 [VWI97,VW98,VWI10,NCI20,SWZGY17,IC17,IC20,LY18] 通常采用一种称为“序列替换”的方法 [VWI97]。其基本做法是迭代地 移除禁用子串，并在字符串末尾用一种替代编码取而代之；该编码由禁用子串的缩短编码以及该子串在编码消息中的索引组成。此外，序列替换的一种变体已用于涉及子串对的全局约束 [EGYM21, NCK+23]。然而，由于子串之间相互依赖，证明迭代编码器收敛性会遇到困难：替换一个禁用子串可能导致另一个子串变为禁用子串。因此，为保证单调递进，针对每种约束都需要利用其性质进行复杂的专门修改；例如，一种方法是在迭代过程中使消息逐渐“缩短”，随后再用一个精心选择的后缀扩展消息，该后缀既满足约束，又能被解码器识别 [VWI10, NCI20, EGYM21, NCK+23]。因此，参数化约束总体上仍缺乏统一框架；即便仅考虑局部参数化约束，我们也需要针对每种信道约束设计复杂的专用编码。

在本文中，我们提出了首个用于参数化受限编码问题的通用框架，并展示了它在多种局部和全局信道约束上的有效性。该框架的核心是一种非常简单的算法，其灵感来自近期关于受约束周期性的研究 [KLBLY22] 。该算法并不具备单调递进性，但仍能高效收敛，这是通过将问题归约为有向无环图遍历的证明所保证的。由于算法简单，它具有很强的通用性，适用于多种信道。我们先以最一般的形式介绍核心算法，随后介绍两种实用的一般化单符号冗余构造，以及首个用于设计可同时满足多个约束的构造的一般性交集方法。我们将此框架应用于多种实际的局部和全局约束，展示了相较于以往工作的参数改进 [VWI97, VW98, VWI10, NCI20, SWZGY17, IC17, IC20, LY18]；由于框架自动保证收敛，所用算法也大为简化。此外，我们还解决了此前尚未解决的局部约束——最小回文串约束，并将其作为交集方法的一部分，用于提出一种混合局部—全局构造，以满足DNA存储中一个此前尚未解决的约束 [WSVDG+18, NCK+23]。最后，我们结合通用框架、交集方法以及受算术编码启发的技术 [RL79]，解决全局近似均衡汉明重量约束这一开放问题。

总体而言，本文的贡献如下：

1. **通用框架：** 在构造 1 中，我们提出了首个通用框架，可直接应用于任何参数化受限编码问题，即使没有单调递进，也能保证高效收敛。具体而言，给定一个单射步进 函数，该函数仅将不满足约束的消息映射到任意非起始消息，此框架便能为该约束构造一种编码方案，并使其以较低的平均时间复杂度收敛。我们通过证明编码器的迭代过程类似于图遍历，进而证明从起始节点出发无法到达图中的任何环，从而证明该方案对任意信道约束均能收敛。

2. **单符号冗余框架：** 参数化受限码的一种常见形式涉及只需要一个冗余符号的构造。在构造 2 中，我们给出了构造 1 的简化版本，该版本只需单个冗余符号，且更易于采用。总的来说，我们发现该构造接收禁用消息的枚举，并返回合法消息的枚举；后续示例将表明，通常找到禁用消息的枚举要简单得多。

3. **交集方法：** 同时遵循多种不同的参数化信道约束，对 DNA存储 [EZ17]等应用尤为重要。以往的研究难以开发出能同时满足多种约束的构造，因为每种约束都需要定制独立的构造，因此各个构造并不相同。然而，本文在构造 3 中提出了一种通用方法，可自动将多个约束（每个约束均通过通用框架实现）整合到单个构造中——这是可行的，因为这些独立构造都基于同一个通用框架。

4. **局部约束：** 我们将通用框架应用于作用于传输消息子串上的局部约束，从而为局部约束提供一种通用构造。随后，我们将这一构造应用于多种实际的局部信道（最小汉明重量、局部近似平衡汉明重量和最小周期性），并证明，相较于以往最先进的构造，我们的方法在参数方面有所改进，且算法简单得多。我们还解决了此前尚未解决的最小回文约束问题，之后又利用该结果改进了另一种全局约束。

5. **全局约束：** 我们展示了通用框架在更具挑战性的全局约束上的有效性，这类约束定义于整个消息之上。我们首先讨论无重复约束，并改进了针对重叠和非重叠重复的最先进结果。随后，我们继续讨论二级结构（DNA 反向互补约束）；以往的研究仅成功处理了非重叠子串，而我们在这一情形下取得了显著改进。此外，通过采用交集方法，我们首次也能处理重叠窗口，并达到相近的参数。最后，我们解决了近似平衡重量约束这一开放问题 [NCI20], ，提出了一种仅需一个冗余符号的高效算法。

本文结构如下。第2节介绍初步定义，随后提出并证明适用于参数化信道约束的通用框架及其一般扩展。第3节将该框架应用于局部参数化约束，第4节则探讨全局参数化约束。最后，第5节总结全文。

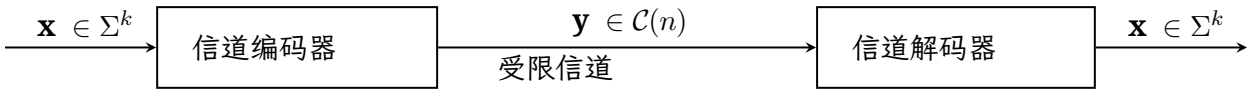


图1：针对参数化信道约束 $\mathcal{C}$ 的构造（编码器和解码器），应用于长度为 $n = k + r$ 的信道。只有当 $\mathbf{y} \in \mathcal{C}(n)$ 时，受限信道才允许传输。

2 通用框架

本节提出一种用于参数化受限编码的新型通用框架，该框架由非单调迭代构造组成，但仍能保证高效的平均时间复杂度。我们首先介绍若干关于参数化受约束信道的初步定义，接着提出本文的基本迭代构造。然后，我们通过将该算法归约为有向无环图遍历来证明其收敛性，并展示其具有高效的平均时间复杂度。最后，我们介绍该构造的两个有趣的一般特殊情形，后续章节将利用它们开展具体应用。

2.1 参数化信道约束

本节明确界定参数化形式的信道约束。参数化约束与固定约束的不同之处在于，它们按照一种一般形式来描述，该形式可推广到长度不同的消息。例如，一种参数化信道约束可以是：长度为 n 的消息不得包含连续 $\log(n)$ 个零。我们将其推广如下：

定义 1（参数化信道约束）。 应用于长度为 n 的信道的参数化信道约束 $\mathcal{C}$，是指仅能传输满足 $\mathbf{y} \in \mathcal{C}(n)$ 的消息 $\mathbf{y} \in \Sigma^n$ 的信道。

在本研究中，我们着重于为参数化信道约束构造高效的构造（编码器和解码器）。如图1所示，这些构造将长度为 k 的所有消息编码为长度为 $n = k + r$、且满足长度为 n 的信道之参数化信道约束的消息。我们明确界定信道编码器和信道解码器如下：

定义 2（信道编码器）。 参数化信道编码器 $f_k : \Sigma^k \to \mathcal{C}(n)$ 将长度为 k 的一般消息编码为长度为 $n = k + r$ 且满足参数化约束 $\mathcal{C}$ 的消息。我们将该编码器的冗余定义为 r。

定义 3（信道解码器）。 参数化信道解码器 $g_k : \mathcal{C}(n) \to \Sigma^k$ 满足 $g_k(f_k(\mathbf{x})) = \mathbf{x}$ 对任意 $\mathbf{x} \in \Sigma^k$ 均成立。

2.2 所提出的构造

本节提出针对参数化信道约束的通用构造。我们首先讨论其方法，然后继续明确定义编码器和解码器。

我们首先建立对通用迭代构造的直观认识。考虑 Σ^n长度为 n的所有候选消息组成的集合，如图 2 右侧所示（其中每个节点代表一条候选消息）。该空间的一种划分方式是依据约束$\mathcal{C}(n)$，图 2 中的节点按照是否满足该约束着色。我们还考虑该空间的一个次级划分 $S \subseteq \Sigma^n$，其大小为 $|\Sigma|^k$，用于表示算法的初始状态（其中 $\psi : \Sigma^k \to S$ 是对 S的一种高效枚举）。最后，我们假设存在一个高效的单射 步进函数 $\varphi: \mathcal{C}(n) \to S$。编码器首先通过

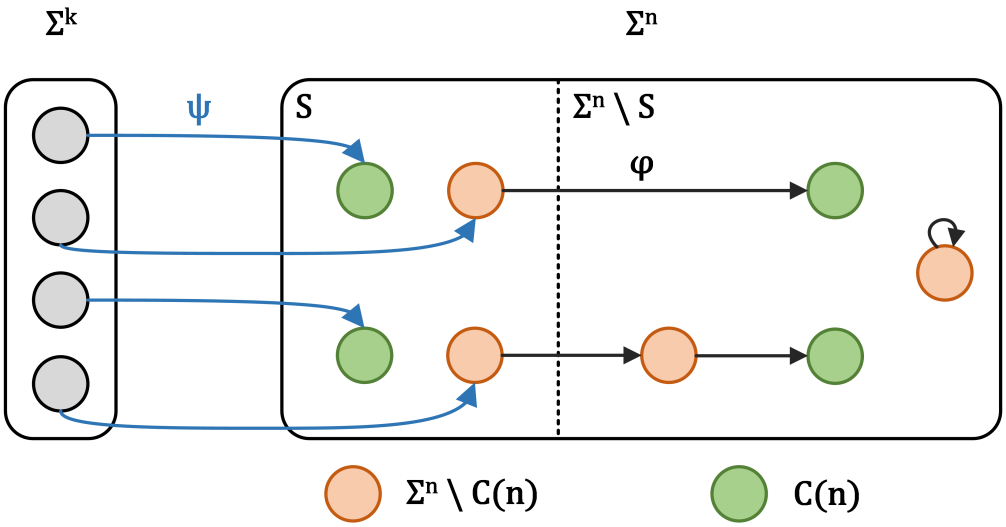


图 2：示例配置通用迭代构造用于 $n = 3$ 和 $k = 2$.

使用 ψ将$\mathbf{x} \in \Sigma^k$ 嵌入 S，然后迭代地对消息应用 φ，直到该消息属于 $\mathcal{C}(n)$。编码器的操作可以看作类似于一台有限状态机，其中 S 表示初始状态，而 $\mathcal{C}(n)$表示接受状态。我们注意到，对 φ 的要求非常宽松，因为只需它是单射函数，整个算法便有效；例如，我们看到，图 2 中的 φ 甚至具有一个自环，但算法永远不会到达这个状态，因为不存在从初始状态通往该自环状态的路径。在以往的参数化受限编码应用中 [VWI97,VW98,VWI10,NCI20,SWZGY17,IC17,IC20,LY18]，迭代算法的收敛性总是通过单调递进来证明的（即消息趋近于一条有效消息）；然而，在这里我们不要求任何此类单调递进，尽管如此，整个构造仍能高效收敛。

我们在算法 1 和算法 2 中形式化地描述了编码器和解码器算法。首先介绍算法 1 中的编码器，它使用 ψ，φ以及指示函数 $\mathbf{1}_{\mathcal{C}(n)}\colon \Sigma^n \to \{0,1\}$ (当且仅当 $\mathbf{y} = 1$ 满足 $\mathbf{y} \in \Sigma^n$ 当且仅当 $\mathbf{y} \in \mathcal{C}(n)$ 时成立)。编码器按如下方式运行：首先，$\mathbf{x}$ 使用 ψ嵌入到 $S \subseteq \Sigma^n$ 中，然后迭代应用 φ，直到消息处于 $\mathcal{C}(n)$ 中。算法 2 中的解码器通过使用φ^{-1}，ψ^{-1} 和指示函数 $\mathbf{1}_S\colon \Sigma^n \to \{0,1\}$，逆向执行编码器的步骤以导出原始消息。解码器本质上是使用 φ^{-1}“解开”编码器的步骤，并利用只有初始状态属于 S 这一事实来确定何时终止解码。总之，我们将该构造概括如下：

构造 1（通用）。对于给定的约束 $\mathcal{C}(n) \subseteq \Sigma^n$，以及以下条件：

1. 子集 $S \subseteq \Sigma^n$ 使得 $|\mathcal{C}(n)| \geq |S| \geq |\Sigma|^k$，2. 单射嵌入 $\psi\colon \Sigma^k \to S$ 和指示函数 $\mathbf{1}_S\colon \Sigma^n \to \{0,1\}$，3. 单射函数 $\varphi : \mathcal{C}(n) \to S$ 和指示函数 $\mathbf{1}_{\mathcal{C}(n)}\colon \Sigma^n \to \{0,1\}$，

然后，算法 1 和 2 构造出一种高效的构造方法，使用 $r = n - k$个冗余符号，平均时间复杂度为$O(|\Sigma|^r \cdot T(n))$，其针对 $T(n)$ 的最大复杂度是 φ，φ^{-1}，$\mathbf{1}_S$，$\mathbf{1}_{\mathcal{C}(n)}$。

算法 1 通用迭代编码器 f_k

输入： $\mathbf{x} \in \Sigma^k$。**输出：** $\mathbf{y} \in \mathcal{C}(n)$。1: $\mathbf{y} \leftarrow \psi(\mathbf{x})$。2: **当** $\mathbf{1}_{\mathcal{C}(n)}(\mathbf{y}) = 0$ **成立时** 3: $\mathbf{y} \leftarrow \varphi(\mathbf{y})$。4: **结束循环** 5:**返回** $\mathbf{y}$。

算法 2 通用迭代解码器 g_k

输入： $\mathbf{y} \in \mathcal{C}(n)$，使得 $f_k(\mathbf{x}) = \mathbf{y}$ 对于某个 $\mathbf{x} \in \Sigma^k$。**输出：** $\mathbf{x} \in \Sigma^k$。1: **当** $\mathbf{1}_S(\mathbf{y}) = 0$ **时**2: $\mathbf{y} \leftarrow \varphi^{-1}(\mathbf{y})$。3: **结束循环**4:**返回** $\psi^{-1}(\mathbf{y})$。

2.3 收敛性与时间复杂度

现在，我们通过将算法执行过程类比为有向无环图遍历，证明所提出的编码器和解码器能够收敛，且平均时间复杂度高效。以往研究 [VWI97, VW98, VWI10, NCI20, SWZGY17, IC17,IC20, LY18] 在受约束信道的迭代构造中，证明收敛性所面临的难点在于：如果不存在单调递进，编码器可能会陷入无限循环。因此，以往所有研究都针对每种信道设计了复杂的定制转移函数，专门用于保证单调递进。相反，我们将在本节证明，即使不存在单调递进，构造 1 也不会进入环，因此对于任意 $\mathcal{C}(n)$ 都将始终收敛。

有趣的是，尽管 φ 中可能存在环，编码器却永远不会到达这些环。例如，考虑图 2 中的自环：一个状态 $\mathbf{y} \in \Sigma^n$，使得 $\varphi(\mathbf{y}) = \mathbf{y}$。从理论上讲，如果编码器到达这个 $\mathbf{y}$，算法就永远不会收敛，因为它会陷入无限循环。然而，我们将在下文证明，由于构造具有两个微妙的性质，编码器总能从根本上避免此类可能导致无限循环的$\mathbf{y}$： φ 的像包含于 S，并且 φ 是单射。定理 1 利用这一点证明，编码器永远不会遇到任何长度的环，而不只是满足 $\varphi(\mathbf{y}) = \mathbf{y}$ 的$\mathbf{y} \in \Sigma^n$。

定理 1. 算法 1 中的编码器总能收敛。

证明。 我们将编码器建模为在有向图 $G = (V, E)$ 上进行遍历，其中节点表示消息，边表示 φ：

$$V = \Sigma^n \qquad E = \{(\mathbf{u}, \mathbf{v}) \mid \mathbf{u} \notin \mathcal{C}(n),\ \varphi(\mathbf{u}) = \mathbf{v}\}. \tag{1}$$

图 2 右侧展示了该图的一个示例。我们首先观察图中节点入度的以下性质：对于任意 $\mathbf{y} \leq 1$，d_{in}（$\mathbf{y} \in \Sigma^n$），因为 φ 是单射；对于任意 $\mathbf{y} \in S$，d_{in}（$\mathbf{y}$），因为 φ 的像包含于_ S 中。现在我们反设该算法对于输入 $\mathbf{x} \in \Sigma^k$ 不收敛。设 $\mathbf{y}^0, \mathbf{y}^1, \ldots$ 为编码器经过的中间状态（其中 $\mathbf{y}^0 = \psi$（$\mathbf{x}$）），对应于 G 中的路径 $\mathbf{y}_0 \to \mathbf{y}_1 \to \cdots$。由于状态空间是有限的（$|V| = |\Sigma|^n$），且遍历不会

终止，因此我们发现存在一个环 $\mathbf{y}_i \to \mathbf{y}_{i+1} \to \cdots \to \mathbf{y}_{j-1} \to \mathbf{y}^j = \mathbf{y}_i$。由于 $\mathbf{y}_0 \in S$ 蕴含 $d_{in}(\mathbf{y}_0) = 0$，因此我们发现 $\mathbf{y}_0$ 不在该环中。于是，该环无法从$\mathbf{y}_0$ 到达，因为环中所有节点的入度至多为 1，且$\mathbf{y}_0$ 不属于该环。这与环可通过路径 $\mathbf{y}_0 \to \mathbf{y}_1 \to \cdots \to \mathbf{y}_i$从 $\mathbf{y}_0$ 到达相矛盾。

□

定理 2. 算法 2 中的解码器是正确的。

证明。我们证明，对于任意 $\mathbf{x} \in \Sigma^k$，$g_k(f_k(\mathbf{x})) = \mathbf{x}$；设 $\mathbf{x} \in \Sigma^k$。令 $t(\mathbf{x})$ 表示在 $f_k(\mathbf{x})$ 中执行的迭代次数（由定理 1 可知其有限），并令 $\mathbf{y}_0 \to \cdots \to \mathbf{y}_t$ 表示编码器在 $G(\mathbf{y}_0 = \psi(\mathbf{x})$ 和 $\mathbf{y}_t = f_k(\mathbf{x}))$ 中所经过的路径。注意，在 $\mathbf{y}0, \ldots, \mathbf{y}_t$ 中，S 中唯一的节点是 $\mathbf{y}_0$，因为对于任意 $\mathbf{y} \in S$，$d_{in}(\mathbf{y}) = 0$。根据算法 2 的设计，我们发现，解码器从 $f_k(\mathbf{x}) = \mathbf{y}_t$ 开始遍历转置图 G^T，直到到达 S 中的一个节点。由于 G 中的最大入度为 1，因此 G^T 中的最大出度也为 1；所以，解码器所经过的路径是确定的，且与编码器所经过路径的反向路径一致。此外，由于只有 $\mathbf{y}_0$ 属于 S，因此解码器在 $\mathbf{y}_0$ 处终止 while 循环。于是，解码器返回 $\psi^{-1}(\mathbf{y}_0) = \psi^{-1}(\psi(\mathbf{x})) = \mathbf{x}$。

□

现在，我们将定理 1 和定理 2 的分析扩展，以证明编码器和解码器的平均时间复杂度为 $O(|\Sigma|^r \cdot f(n))$。

引理 1. 算法 1 中 while 循环的平均迭代次数至多为 $|\Sigma|^r$。

证明. 令 $t(\mathbf{x})$ 表示在 $f_k(\mathbf{x})$ 中执行的迭代次数。如定理 1 所示，算法 1 的一次执行等价于 G 上一条长度为 $t(\mathbf{x})$ 的路径。由于对所有$\mathbf{y} \in \Sigma^n$，$d_{in}(\mathbf{y}) \leq 1$，且 ψ 是单射，因此我们发现，由两个不同输入在 G 中所经过的任意两条路径互不相交。由于所有路径都位于 Σ^n 中，因此我们发现路径长度之和满足：

$$\sum_{\mathbf{x} \in \Sigma^k} t(\mathbf{x}) \leq |\Sigma|^n \implies \frac{1}{|\Sigma|^k} \cdot \sum_{\mathbf{x} \in \Sigma^k} t(\mathbf{x}) \leq |\Sigma|^{n-k} = |\Sigma|^r .$$

□

推论 1. 对于 $T(n)$，编码器和解码器的平均时间为 $O(|\Sigma|^r \cdot T(n))$，即在 $\varphi, \varphi^{-1}, \mathbf{1}_S, \mathbf{1}_{\mathcal{C}(n)}$ 中的最大时间复杂度。

2.4 一般扩展

本节介绍通用迭代算法的两种实用扩展，后续章节将会用到。我们先给出一种构造，它简化了构造 1 中针对单符号冗余的判定条件，

构造 2. 设 $\mathcal{C}(n) \subseteq \Sigma^n$ 是给定的约束，满足 $|\mathcal{C}(n)| \geq |\Sigma|^{n-1}$。给定一个单射函数 $\xi : \mathcal{C}(n) \to \Sigma^{n-1}$ 以及一个指示函数 $\mathbf{1}_{\mathcal{C}(n)} : \Sigma^n \to \{0, 1\}$，则存在一种高效的信道构造，使用一个冗余符号，且对于 $T(n)$，在 $\xi, \xi^{-1}, \mathbf{1}_{\mathcal{C}(n)}$ 中达到最大时间复杂度时，平均时间复杂度为 $O(T(n))$。

证明. 我们将通过构造 1 的一个特例展示所需的构造，其中取 $r = 1$ 和 $k = n - 1$，并选择特定的划分 S 和嵌入 ψ。我们作如下选择，

1. 子集 $S = \{\mathbf{x}1 \mid \mathbf{x} \in \Sigma^{n-1}\}$，满足条件 $|\mathcal{C}(n)| \geq |S| \geq |\Sigma|^{n-1}$ as $|\mathcal{C}(n)| \geq |\Sigma|^{n-1}$ and $|S| = |\Sigma|^{n-1}$。

2. 嵌入 ψ： $\Sigma^{n-1} \to S$ 使得 $\psi(\mathbf{x}) = \mathbf{x}1$，且 $\mathbf{1}_S(\mathbf{x}) = x_n$ 对于 $\mathbf{x} = (x_1, x_2, \ldots, x_n)$。

3. 单射函数 φ： $\mathcal{C}(n) \to \bar{S}$，使得 $\varphi(\mathbf{x}) = \xi(\mathbf{x})0$（单射性质由$\xi$ 是单射函数这一点得出），并使用给定函数 $\mathbf{1}_{\mathcal{C}(n)}$： $\Sigma^n \to \{0,1\}$。

因此，根据构造 1 的正确性，我们发现存在一个针对 $\mathcal{C}(n)$ 的信道构造，其 $r = 1$ 和 $O(|\Sigma| \cdot T(n)) = O(T(n))$ 的平均时间复杂度。 □

接下来，我们将展示一种由多个信道的交集导出的信道构造。对于传输消息需要同时满足多种不同约束的问题，这一点很有意义（例如，在DNA存储中，一种常见的组合是既满足近似平衡，又避免连续长度为 [EZ17]）。由于这些约束彼此相互依赖，朴素地尝试组合针对各项约束构造的方法并不可行，因此算法能否收敛并不明确。相反，遵循本文提出的通用框架，我们发现也可以相当容易地导出一种用于信道交集的构造，

构造3（信道交集）。设给定约束 $\mathcal{C}_1(n)\ \mathcal{C}_2(n) \ldots, \mathcal{C}_m(n) \subseteq \Sigma^n$，使得 $\forall i, \left|\mathcal{C}_i(n)\right| \leq |\Sigma|^{n-1-\lceil \log_{|\Sigma|}(m) \rceil}$。给定单射函数 $\forall i, \xi_i$： $\mathcal{C}_i(n) \to \Sigma^{n-\lceil \log_{|\Sigma|}(m) \rceil - 1}$以及指示函数 $\forall i, \mathbf{1}_{\mathcal{C}_i(n)}$： $\Sigma^n \to \{0,1\}$，存在一种高效的信道构造，用于$\mathcal{C}(n) = \bigcap_i \mathcal{C}_i(n)$，该构造仅需一个冗余符号，且平均时间复杂度为 $O(m \cdot T(n))$；对于 $T(n)$，其最大时间复杂度为 $\xi_1, \ldots, \xi_m, \xi_1^{-1}, \ldots, \xi_m^{-1}, \mathbf{1}_{\mathcal{C}_1(n)} \cdots, \mathbf{1}_{\mathcal{C}_m(n)}$ 中的最大值。

证明。 我们将证明，所需构造是构造 2 的一个特例：

1. 我们按如下方式验证 $|\mathcal{C}(n)| \geq \Sigma^{n-1}$，

$$|\mathcal{C}(n)| = \left|\bigcap_i \mathcal{C}_i(n)\right| \geq |\Sigma|^n - \sum_i \left|\overline{\mathcal{C}_i(n)}\right| \geq |\Sigma|^n - m \cdot |\Sigma|^{n-1-\lceil \log(m) \rceil} \geq |\Sigma|^{n-1}$$

2. 我们提出如下单射函数 ξ： $\mathcal{C}(n) \to \Sigma^{n-1}$，其中 $\xi(\mathbf{x})$ 的计算方式如下：令 i 为满足 $\mathbf{x} \in \mathcal{C}_i(n)$ 的最小索引（该索引存在，因为 $\mathbf{x} \in \mathcal{C}(n)$），则：

$$\xi(\mathbf{x}) = \xi_i(\mathbf{x})\mathbf{p}_i$$

其中 $\mathbf{p}_i$ 是使用 $\lceil \log(m) \rceil$ 个符号对 i 进行的编码。

3. 指示函数 $\mathbf{1}_{\mathcal{C}(n)}$： $\Sigma^n \to \{0,1\}$ 的计算方式为 $\mathbf{1}_{\mathcal{C}(n)}(\mathbf{x}) = \bigwedge_i \mathbf{1}_{\mathcal{C}_i(n)}(\mathbf{x})$.

因此，根据构造 2 的正确性，我们得知存在一个针对 $\mathcal{C}(n)$ 的信道构造，其 $r = 1$ 的平均时间复杂度为 $O(m \cdot T(n))$，对于 $T(n)$，最大时间复杂度为 $\xi_1, \ldots, \xi_m, \xi_1^{-1}, \ldots, \xi_m^{-1}, \mathbf{1}_{\mathcal{C}_1(n)} \cdots, \mathbf{1}_{\mathcal{C}_m(n)}$。 □

在第 3 节和第 4 节中，我们主要基于构造 2，展示通用框架对各种约束的有效性。虽然不可能列出构造 3 的所有约束组合，但我们将以一个约束组合为例，展示其如何解决一个此前尚未解决、对 DNA存储 至关重要的信道约束。

3 局部信道约束

本节探讨一种特殊情况，其中涉及适用于传输消息的子串的局部信道约束。也就是说，$\mathcal{C}$ 根据一个参数集定义，该参数集规定了不得出现在传输字中的子串。我们首先定义局部信道约束，接着将通用构造应用于局部信道约束，最后评估该构造在各种实际约束下的有效性。

3.1 定义与预备知识

我们首先定义如下子串避免（SA）受限信道，

定义 4 (SA). $\mathbf{w}(n)$-子串避免受限信道是参数化受限信道 $\mathcal{C}$，满足 $\mathcal{C}(n)$ 包含所有长度为 n 且不包含 $\mathbf{w}(n)$ 作为子串的词。

我们将定义扩展到多子串避免（MSA）向量，如下所示，

定义 5 (MSA). $W(n)$-多子串避免受限信道是参数化受限信道 $\mathcal{C}$，满足 $\mathcal{C}(n)$ 包含所有长度为 n 且不包含任何词$\mathbf{w} \in W(n)$ 作为子串的词。为简单起见，我们假设所有$\mathbf{w} \in W(n)$ 的长度均为 $\ell = \ell(n)$。[1]

本节探讨多子串避免受限信道，提出该信道的一种通用构造，并将各种实用的受限信道表示为多子串避免信道。

3.2 所提出的构造

我们为所有多子串避免码提出如下构造：

构造 4（MSA）。对于给定的 MSA约束 $W(n)$，以及（1）给定的单射函数χ：$W(n) \to \Sigma^{\ell'}$，满足 $\ell' = \ell - \lceil \log_{|\Sigma|}(n) \rceil - 1$；（2）给定的指示函数 $\mathbf{1}_{W(n)}$：$\Sigma^{\ell} \to \{0,1\}$，存在一种显式参数化构造，该构造仅需一个冗余符号，平均时间复杂度为 $O(n \cdot T(\ell))$，其中 $T(\ell)$ 是 χ、χ^{-1}、$\mathbf{1}_{W(n)}$ 中的最大时间复杂度。

证明。我们证明该构造是构造 2 的一个特例。我们按如下方式选择 $\xi : \mathcal{C}(n) \to \Sigma^{n-1}$：令 i 为 $\mathbf{x}$ 中第一个属于 $W(n)$ 的长度为 ℓ 的子串的索引（这样的子串必然存在，因为 $\mathbf{x} \in \mathcal{C}(n)$），则

$$\xi(\mathbf{x}) = \mathbf{x}_1^{i-1}\ \mathbf{x}_{i+\ell}^{n}\ \mathbf{p}_i\ \chi(\mathbf{x}_i^{i+\ell-1}),$$

其中，$\mathbf{x}_j^k$ 是 $\mathbf{x}$ 从 j 到 k（含端点）的子串，$\mathbf{p}_i$ 是使用 $\lceil \log(n) \rceil$ 个符号对 i 进行的编码。由于 χ：$W(n) \to \Sigma^{\ell'}$，因此 $|\xi(\mathbf{x})| = n - \ell + \lceil \log(n) \rceil + \ell' = n - 1$。我们发现，$\xi$ 是单射，因为窗口是以确定性方式选取的，且 χ 是单射。因此，ξ^{-1} 存在，并且可以通过对最后 ℓ' 个符号执行 χ^{-1}、从接下来的 $\lceil \log(n) \rceil$ 个符号中解码 $\mathbf{p}_i$，最后将窗口重新插入消息中来计算。此外，我们选取 $\mathbf{1}_{\mathcal{C}(n)}$：$\Sigma^n \to \{0,1\}$作为一个函数；当消息中至少有一个 $n-\ell+1$ 个窗口属于 $W(n)$（使用 $\mathbf{1}_{W(n)}$ 求值）时，该函数返回 1。我们发现，编码的平均时间复杂度为 $O(n \cdot T(\ell))$，解码的平均时间复杂度为 $O(T(\ell) + n)$，因为 ξ、$\mathbf{1}_{\mathcal{C}(n)}$可以在 $O(n \cdot T(\ell))$ 时间内完成，而 ξ^{-1}可以在 $O(T(\ell) + n)$ 时间内完成。

□

[1]否则，可将 $W(n)$ 中的较短的词用所有可能的组合填充至最大长度。

我们注意到，对于任何满足以下条件的 $W(n)$，该构造都能保证得到一个仅含单个冗余符号的码：

$$|W(n)| \leq |\Sigma|^{\ell-\log(n)-1} = \frac{1}{|\Sigma| \cdot n} \cdot |\Sigma|^{\ell}. \tag{2}$$

此外，如果提供了计算 χ 的高效算法，那么该码也具备高效的编码和解码能力。否则，注意到始终存在一种算法，其时间复杂度为 $T(\ell) = O(\ell \cdot \log(|W(n)|))$，空间复杂度为 $O(\ell \cdot |W(n)|)$（对 $W(n)$进行二分搜索）。

3.3 应用

本节回顾了种类繁多的局部信道约束，并展示了所提出的通用约束的有效性。我们从冗余 r 和所支持的最小 ℓ （支持更小的 ℓ 等同于采用更严格的约束，因而更理想）方面，将所提出的构造与以往最先进的算法进行比较。此外，我们解决了一个此前尚未解决的最小回文串问题；在第4节中，我们也将利用这一结果解决一个同样此前尚未解决的改进型反向互补约束问题。

3.3.1 最小汉明重量

最小汉明重量（MW）约束要求传输消息的所有 ℓ子串都具有某个最小汉明重量 $p(n)$，其中 $\Sigma = \{0,1\}$。[2] 以往最先进的工作 [LY18] 基于一种迭代方法，专门用于保证单调递进，需要$p(n)$位冗余，并支持最小 ℓ，其值为 $\lceil \log(n) \rceil + (p(n)-1) \cdot \lceil \log(\ell+2) \rceil + 2$。该算法首先在消息末尾填充 $p(n)$个1，然后迭代地将汉明重量小于 $p(n)$的窗口替换为其替代表示，同时额外填充10，以保证就已满足约束的窗口而言保持单调递进。

现在，我们将所提出的构造应用于该信道。首先，我们观察到该信道是一个按以下方式定义的MSA约束，

$$W_{\ell,p-MW}(n) = \left\{ \mathbf{w} \in \Sigma^{\ell} \mid w_H(\mathbf{w}) < p(n) \right\}. \tag{3}$$

我们按如下方式设计 χ： $W(n) \to \Sigma^{\ell'}$对于$\mathbf{w} \in W(n)$，我们定义 $\chi(\mathbf{w}) = \mathbf{p}^{i_1}\mathbf{p}^{i_2} \cdots \mathbf{p}^{i_{p(n)-1}}$，其中$i_1, \ldots, i_{p(n)-1}$是$\mathbf{w}$中非零符号的索引，$_i$则是$\mathbf{p}$的二进制编码，使用$\lceil \log(\ell+1) \rceil$ 位。因此，我们要求，

$$\ell' = (p(n)-1) \cdot \lceil \log(\ell+1) \rceil \leq \ell - \lceil \log(n) \rceil - 1$$

$$\iff \ell \geq \lceil \log(n) \rceil + (p(n)-1) \cdot \lceil \log(\ell+1) \rceil + 1.$$

因此，我们发现，所提出的构造同时将最小 ℓ 从 $\lceil \log(n) \rceil + (p(n)-1) \cdot \lceil \log(\ell+2) \rceil + 2$ 提升至 $\lceil \log(n) \rceil + (p(n)-1) \cdot \lceil \log(\ell+1) \rceil + 1$，更重要的是，还将冗余从 $p(n)$个符号减少到仅需一个冗余符号。

[2] $|\Sigma| > 2$ 的情形可直接类似地处理。[3]如果 $w_H(\mathbf{w}) < p(n) - 1$，那么我们附加一个值为 ℓ 的哑索引，以表示没有条目。

3.3.2 局部近似平衡汉明重量

现在，我们转而考虑局部近似平衡汉明重量（LAB）信道约束，该约束要求所有长度为 ℓ的子串的汉明重量都处于区间 $[p_1\ell, p_2\ell]$内。此前最先进的 [NCI20] 也采用迭代方法，针对这一特定信道设计了由三个阶段（初始阶段、替换阶段和扩展阶段）组成的流程，以保证单调收敛。他们首先证明，存在一个使用单个冗余符号、长度为 $\ell \geq \ln(n)/c^2$ 的码，其中 $c = \min\left\{\frac{1}{2} - p_1, p_2 - \frac{1}{2}\right\}$；然而，由于还需使用额外符号来保证收敛，他们最终仅成功实现了$\ell \geq \ln(n)/c^2 + 2$ 。相比之下，我们所提出的构造成功实现了 $\ell \geq \ln(n)/c^2$ ，且只需一个冗余符号。

我们首先根据以下内容将信道约束表示为MSA约束，

$$W_{\ell,p_1,p_2-LAB}(n) = \left\{\mathbf{w} \in \Sigma^{\ell} \mid w_H(\mathbf{w}) \notin [p_1\ell, p_2\ell]\right\}. \tag{4}$$

对于 χ，我们采用前人工作 [NCI20]中假设的相同函数 F 。我们发现，当 $\ell \geq \ln(n)/c^2$ 时，构造4的条件得到满足（见 [NCI20]），从而实现了 [NCI20]中证明在理论上可实现的结果。请注意，所提出的算法不需要存在 [NCI20], 中定义的额外函数 G ，而该函数对于 [NCI20]算法的扩展阶段至关重要。

总的来说，我们在冗余相同的情况下实现了略有改进的 ℓ ；由于无需进行扩展，我们还少假设存在一个函数。

3.3.3 最小周期性

我们首先明确地定义字符串的周期性，

定义 6。若对所有 $1 \leq i \leq \ell - p, w_i = w_{i+p}$，均有$\mathbf{w} \in \Sigma^{\ell}, 1 \leq p \leq \ell - 1$，则称其为$\mathbf{w}$ 的周期。

最小周期性（MP）信道约束要求，所有 ℓ-窗口都具有至少为 $p(n)$ 的最小周期性。请注意，这种信道约束是广为人知的游程长度受限（RLL）约束 [LY18] 的扩展；该约束限制由同一符号重复构成的长序列（$p(n) = 2$ 与 RLL 类似）。近期一项工作 [KLBLY22] 基于本文推广的方法展示了近乎最优的结果：当 ℓ 至少为 $\lceil \log(n) \rceil + p + 1$时，仅需一个冗余位；与此前最先进的结果 [SB19] 相比，其冗余和最小 ℓ均有显著改进。本文提出的广义构造，在按以下方式定义的 MSA 约束下，与这些最先进的结果一致，

$$W_{\ell,p-MP}(n) = \left\{\mathbf{w} \in \Sigma^{\ell} \mid \exists p' < p(n) \text{ such that } \mathbf{w} \text{ has minimal period } p'\right\}, \tag{5}$$

对于选作由$\mathbf{w}$的前 p'位、一个$\mathbf{w}$符号1以及 $p(n)$个符号0串接而成的 χ（同时用于编码 p' 和可据以恢复窗口的前 p' 个符号 [KLBLY22]）：

$$\chi(\mathbf{w}) = \mathbf{w}_1^{p'} 10^{p(n)-p'-1}.$$

因此，我们发现，所提出的算法成功地将 [KLBLY22] 推广到不仅适用于周期性的情形，同时在周期性情形下仍达到最先进的结果。

3.3.4 最小回文串

精确无回文约束（ENP）规定，长度恰为ℓ 的回文不能出现在长度为 n的消息中。该约束在DNA存储中有应用，因为回文会对DNA测序流程产生负面影响 [WSVDG+18]。然而，据我们所知，此约束此前尚未得到解决，这可能是因为序列替换技术中的扩展步骤较为复杂。我们将该约束表示为如下的MSA约束，

$$W_{\ell-ENP}(n)=\left\{\mathbf{w}\in\Sigma^{\ell}\;\middle|\;\mathbf{w}^{R}=\mathbf{w}\right\}, \tag{6}$$

其中 $(w_1,\ \ldots\ldots,\ w_\ell)^R=(w_\ell,\ \ldots\ldots,\ w_1)$。对于 $\chi:\ W(n)\to\Sigma^{\ell'}$，我们利用 $\chi(\mathbf{w})=\mathbf{w}_1^{\lceil\ell/2\rceil}$，因为字符串 $\mathbf{w}_{\lceil\ell/2\rceil+1}^{n}$ 的其余部分可以由 $\mathbf{w}_1^{\lceil\ell/2\rceil}$ 推导得出，即 $\mathbf{w}^R=\mathbf{w}$。因此，我们要求：

$$\ell'=\lceil\ell/2\rceil\le\ell-\lceil\log(n)\rceil-1\iff\lfloor\ell/2\rfloor=\ell-\lceil\ell/2\rceil\ge\lceil\log(n)\rceil+1 \tag{7}$$

因此，我们发现存在一种高效构造，只需使用单个冗余符号即可满足$\lfloor\ell/2\rfloor\ge\lceil\log(n)\rceil+1$。请注意，这也涵盖了长度为 $\ell+2k$的所有回文，因为任意长度为 $\ell+2k$ 的回文都包含一个长度为 ℓ的回文。为了解决一般最短回文约束（MPL），即限制长度为n的消息中任意回文的长度为 $\ge\ell$ 的约束，我们发现可以利用构造 3，构造一种适用于 MPL 的方案，该方案使用单个冗余符号和 $\ell=2\cdot\lceil\log(n)\rceil+4$（结合偶数和奇数 ℓ的 ENP 构造）。

4 全局信道约束

本节讨论两种作用于整个传输消息中子序列对的全局 信道约束。首先介绍无重复约束 [EGYM21]，它会消除传输消息中两个或更多位置出现的相同子序列；随后介绍二级结构约束（DNA反向互补），这是对无重复约束的推广。我们进一步改进了二级结构约束的构造，使其也能处理重叠窗口，从而解决了一个此前尚未解决、对DNA存储应用至关重要的问题 [WSVDG+18, NCK+23]。最后，我们解决全局近似平衡汉明重量信道 [NCI20] 在权重裕量 $\pm\sqrt{n}$ 下的开放问题，并展示了一种只需单个冗余符号的高效算法。

4.1 无重复

无重复约束要求，传输消息中的任意 ℓ 长子串在消息中恰好出现一次（即不存在长度为 ℓ 的重复子串）。在 [EGYM21]，中，作者全面研究了无重复约束，并提出了一种利用两个冗余符号 来支持 $\ell\ge2\cdot\lceil\log(n)\rceil+2$ 的构造。该构造基于采用消除阶段和扩展阶段的迭代方法（与此前其他工作类似），但作者还进一步限制算法，要求不存在 $\log(n)+1$ 个连续的零，因为这是扩展阶段正确运行所必需的。相比之下，我们通过应用构造 2，仅使用单个冗余符号 即可实现 $\ell\ge2\cdot\lceil\log(n)\rceil+1$。

我们首先明确地将重复自由（RF）约束定义如下：

$$\mathcal{C}_{\ell-RF}(n)=\left\{\mathbf{x}\in\Sigma^{n}\;\middle|\;\nexists1\le i<j\le n-\ell(n)+1\ :\ \mathbf{x}_i^{i+\ell(n)-1}=\mathbf{x}_j^{j+\ell(n)-1}\right\}, \tag{8}$$

对于 $\ell=\ell(n)$。我们将构造 2 应用于该问题，以找到，

构造5 (无重复)。 当 $\ell = 2 \cdot \lceil \log(n) \rceil + 1$ 时，存在一种针对 $\mathcal{C}_{\ell-RF}(n)$ 的高效构造，使用单个冗余符号，平均时间复杂度为 $O(n \log^2(n))$。

证明。 我们将通过将所需构造作为构造 2 的一个特例来加以证明。我们选择$\xi : \mathcal{C}_{\ell-RF}(n) \to \Sigma^{n-1}$ 如下：令 $1 \le i < j \le n - \ell + 1$ 为满足$\mathbf{x}_i^{i+\ell-1} = \mathbf{x}_j^{j+\ell-1}$ 的最小[4] 索引（必定存在，因为$\mathbf{x} \in \mathcal{C}_{\ell-RF}(n)$），然后我们将 $\xi(\mathbf{x})$ 定义为：移除位于 j 的窗口后，再附加 i 和 j的编码，

$$\xi(\mathbf{x}) = \mathbf{x}_1^{j-1}\mathbf{x}_{j+\ell}^{n}\mathbf{p}_i\mathbf{p}_j$$

其中$\mathbf{p}_m$是使用 $\lceil \log(n) \rceil$ 个符号对 m 进行的编码。我们发现 $|\xi(\mathbf{x})| = n - \ell + 2 \cdot \lceil \log(n) \rceil = n - 1$，符合要求。我们通过给出如下 ξ^{-1}来证明 ξ是单射： $\xi^{-1}(\mathbf{y})$ 首先移除并解码$\mathbf{y}$ 的最后 $2 \cdot \lceil \log(n) \rceil$ 个符号，以找出 i和 j，然后：

- 如果 $j \ge i+\ell$ （无重叠），那么 ξ^{-1} 会复制$\mathbf{y}$ 中索引为 i 的窗口，并将其插入位置j。在这种情况下，我们发现，所提出的 ξ^{-1} 能够成功反演 ξ，因为$\mathbf{x}$中位于 i和 j 的窗口是相同的。

- 如果 $j < i + \ell$ （重叠），那么我们发现$\mathbf{w} = \mathbf{x}_i^{i+\ell-1}$具有周期 $j - i$，因为$\mathbf{w}^{\ell-(j-i)+1} = \mathbf{w}_{\ell-(j-i)}1$。因此， ξ^{-1} 从$\mathbf{y}_i^{i+j-i-1} = ()\mathbf{x}_i^{i+j-i-1} = ()\mathbf{w}_1^{j-i}$ 中重构出$\mathbf{w}$ （利用$\mathbf{w}$ 具有 $j - i$ 周期这一事实），然后将$\mathbf{w}$ 插入$\mathbf{y}$中的索引 j 处。所提出的ξ^{-1} 能够成功反演 ξ，因为它恢复了位于 j 的窗口，然后重新插入该窗口。

我们选择 $\mathbf{1}_{\mathcal{C}_{\ell-RF}(n)}$ 作为一种算法，用于检查是否存在两个重复的 ℓ-子串，时间复杂度为$O(n \log(n) \cdot \ell) = O(n \log^2(n))$（例如，使用基于树的集合）。根据构造 2，我们发现编码和解码的总体平均时间复杂度均为 $O(n\log^2(n))$。 □

我们注意到，构造5可以用相同的参数推广到符号级无重复（SRF）约束，其定义如下：

$$\mathcal{C}_{\ell,\alpha-SRF}(n) = \left\{ \mathbf{x} \in \Sigma^n \;\middle|\; \nexists 1 \le i < j \le n - \ell + 1 \;:\; \alpha(\mathbf{x}_i^{i+\ell-1}) = \mathbf{x}_j^{j+\ell-1} \right\} \tag{9}$$

对于 α： $\Sigma^\ell \to \Sigma^\ell$一个符号级函数（即， $\alpha(x_1, \ldots\ldots, x_\ell) = (\beta_1(x_1)\ldots\ldots, \beta_\ell(x_\ell))$，对于 $\forall i$， β_i： $\Sigma \to \Sigma$）。

4.2 二级结构（DNA反向互补）

二级结构信道约束源于 DNA存储 [MK06], 领域，其中数据使用字母表 $\Sigma = \{A, T, C, G\}$编码在合成DNA链中。为了防止 DNA 链自身折叠，链中不能存在两个互为反向互补（RC）的窗口[5]：

$$RC(x_1, \ldots, x_n) = (c(x_n), \ldots, c(x_1)), \quad \text{where} \quad c(\alpha) = \begin{cases} A & \alpha = T \\ T & \alpha = A \\ C & \alpha = G \\ G & \alpha = C \end{cases} \tag{10}$$

[4]该顺序由 i 上的主序和 j 上的次序定义。[5]注意，支持任意满足 $c = c^{-1}$ 的 c：$\Sigma \to \Sigma$。

目前最先进的构造 [NCK+23] 解决的是一个宽松问题，其中只考虑不重叠的[6]窗口，并基于一种包含消除阶段和扩展阶段的迭代构造。它们通过填充AC的形式进行扩展$^{(n-N_0)/2}$，这要求进一步约束消息，使其不包含任何形式为 $(\alpha_1\alpha_2)^{(3\log n+2)/2}$的模式，其中 $\alpha_1, \alpha_2 \in \Sigma$。总的来说，对于 $\ell \geq 3\cdot\lceil\log(n)\rceil + 4$，它们提出了一种只需单个冗余符号的构造。我们证明，所提出的构造在满足这一宽松约束的同时，可以达到单符号冗余，并将最小 ℓ 从 $3\cdot\lceil\log(n)\rceil + 4$ 降至 $2\cdot\lceil\log(n)\rceil + 1$。

W我们首先明确地定义宽松二级结构（RSS）约束 ，

$$\mathcal{C}_{\ell-RSS}(n) = \left\{\mathbf{x}\in\Sigma^n \;\middle|\; \nexists 1\leq i < n-\ell+1, i+\ell\leq j\leq n-\ell+1 \;:\; RC(\mathbf{x}_i^{i+\ell-1}) = \mathbf{x}_j^{j+\ell-1}\right\}. \tag{11}$$

与构造5类似，我们发现，将构造 2 应用于该问题可得到以下结果：

构造 6（宽松二级结构）。当 $\ell = 2\cdot\lceil\log(n)\rceil + 1$ 时，对于 $\mathcal{C}_{\ell-RSS}(n)$，存在一种仅使用单个冗余符号的高效构造，其平均时间复杂度为 $O(n\log^2(n))$。

证明。与构造 5 的证明类似，无需处理重叠窗口，并使用反向互补函数代替相等关系。

□

为了同时处理重叠子串，我们注意到，由于它们彼此重叠，重叠子串构成一种局部约束。具体而言，我们发现，一个 ℓ 子串与其反向互补的重叠会确保消息中存在长度为 $2\cdot\lfloor\ell/2\rfloor + 2$ 的回文。因此，利用构造 3 和第3.3.4节，我们得到以下构造，

$$\mathcal{C}_{\ell-SS}(n) = \left\{\mathbf{x}\in\Sigma^n \;\middle|\; \nexists 1\leq i < j\leq n-\ell+1 \;:\; RC(\mathbf{x}_i^{i+\ell-1}) = \mathbf{x}_j^{j+\ell-1}\right\}. \tag{12}$$

构造 7（二级结构）。当 $\ell = 2\cdot\lceil\log(n)\rceil + 2$ 时，存在一种高效构造，可为 $\mathcal{C}_{\ell-SS}(n)$ 使用单个冗余符号，其平均时间复杂度为 $O(n\log^2(n))$。

4.3 近似平衡汉明重量

我们在此考虑全局近似平衡约束，它将众所周知的平衡 Knuth 码 [Knu86] 推广为要求整个消息的汉明重量大约为 $n/2$。这一构造对 DNA存储 等应用具有广泛意义，因为这些应用需要平衡的 GC 含量 [BGH+16]。组合分析表明，对于由所有长度为 n、汉明重量为 $[n/2-\sqrt{n}/2, n/2+\sqrt{n}/2]$ 的码字组成的信道，当 n[7]足够大时，存在一种使用单个冗余符号的构造。尽管如此，这个问题仍未解决：当前最先进的构造 [NCI20] 仅针对 $[np_1, np_2]$ 的线性近似平衡版本，且仅适用于 $p_1 < 1/2 < p_2$。下文将展示一种基于构造 3 和算术编码 [RL79] 技术、使用单个冗余符号的 $[n/2-\sqrt{n}, n/2+\sqrt{n}]$ 高效构造。我们首先明确地定义该约束如下：

$$\mathcal{C}_{AB}(n) = \left\{\mathbf{x}\in\Sigma^n \;\middle|\; w_H(\mathbf{x})\in\left[\frac{n}{2}-\sqrt{n}, \frac{n}{2}+\sqrt{n}\right]\right\}. \tag{13}$$

[6]由于反向互补不再意味着周期性，重叠窗口更难处理。[7]这可以从以下事实看出：当 $n\to\infty$ 时，二项分布趋近于正态分布，因此至少一半的空间包含在 $[\mu-\sigma, \mu+\sigma] = [n/2-\sqrt{n}/2, n/2+\sqrt{n}/2]$ 中。

我们还为整体构造定义了以下两个辅助约束 n,

$$\mathcal{C}_{AB-L}(n) = \left\{ \mathbf{x} \in \Sigma^n \;\middle|\; w_H(\mathbf{x}) \leq \frac{n}{2} + \sqrt{n} \right\}, \tag{14}$$

$$\mathcal{C}_{AB-H}(n) = \left\{ \mathbf{x} \in \Sigma^n \;\middle|\; w_H(\mathbf{x}) \geq \frac{n}{2} - \sqrt{n} \right\}. \tag{15}$$

注意到 $\mathcal{C}_{AB} = \mathcal{C}_{AB-L}(n) \cap \mathcal{C}_{AB-H}(n)$。现在我们提出如下整体构造,

构造 8（近似平衡）。当 $n > 4$ 时，存在一种高效构造，可在 $\mathcal{C}_{AB}(n)$ 中使用单个冗余符号。

证明。我们将此构造展示为构造 3 的一个特例，其中利用 $\mathcal{C}_{AB} = \mathcal{C}_{AB-L}(n) \cap \mathcal{C}_{AB-H}(n)$。不失一般性，我们将单射函数 $\xi_1 : \mathcal{C}_{AB-H} \to \Sigma^{n-2}$ 设计如下：给定一个字 $\mathbf{x} \in \mathcal{C}_{AB-H} \iff w_H(\mathbf{x}) < n/2 - \sqrt{n}$，我们利用 算术编码_[RL79][8]，通过 $p = 1/2 + 1/\sqrt{n}$ 生成长度为以下数值的区间,

$$\left(\frac{1}{2} - \frac{1}{\sqrt{n}}\right)^{w_H(\mathbf{x})} \cdot \left(\frac{1}{2} + \frac{1}{\sqrt{n}}\right)^{n - w_H(\mathbf{x})} \geq 1/\left|\Sigma\right|^{n-2}$$

对于 $w_H(\mathbf{x}) = n/2 - \sqrt{n} - 1$ 的最坏情况，只要 $n > 4$。因此，根据算术编码的正确性，我们发现已成功将 ξ_1 定义为从$\mathcal{C}_{AB-H}$ 到 Σ^{n-2}的单射函数。

因此，我们发现存在所需的 $\xi_1 : \mathcal{C}_{AB-H} \to \Sigma^{n-2}$ 和 $\xi_2 : \mathcal{C}_{AB-L} \to \Sigma^{n-2}$。因此，根据构造 3，我们发现存在一种高效构造，可用于 $\mathcal{C}_{AB} = \mathcal{C}_{AB-L}(n) \cap \mathcal{C}_{AB-H}(n)$，且冗余仅为 1 位。

□

5 结论

参数化受限编码将广为人知的固定约束编码领域加以推广，用于处理随消息长度增长、并体现某种一般性质的约束（例如，避免出现连续 $\log(n)$ 个零的窗口）。对于任何固定 受限编码任务，都存在通用解；但参数化 约束缺少此类解法，因此人们不得不针对每种约束设计定制算法。具体而言，多数方法通过迭代替换传输消息中的无效子串来实现编码，但这需要复杂的定制转移函数，以实现单调递进并保证收敛。本文提出一种适用于所有参数化受限编码问题的新型通用框架。令人意外的是，我们通过证明指出，要实现高效收敛，并不需要单调递进，因此该框架能够推广到所有约束。我们将这一通用约束应用于局部和全局参数化约束，并针对多种问题显著改进了最先进的构造，大幅简化了相关算法。此外，我们还首次为若干此前尚未解决的约束给出了构造。因此，这一通用框架也可轻松应用于未来，以应对其他此前尚未解决的受限编码问题。

[8]算术编码是一种迭代方法，每次迭代都将实数区间 $[0, 1)$ 划分为 $[0, p)$ 和 $[p, 1)$，其中区间的选择取决于当前输入位。通过证明所选词对应的区间长度至少为 $\ell > 0$，可以用 $\lceil \log(1/\ell) \rceil$ 个符号枚举这些词。

参考文献

[ACH83] Roy Adler、Don Coppersmith 和 Martin Hassner。滑动分组码算法——符号动力系统在信息论中的应用。《IEEE信息论汇刊》，29(1):5–22，1983年。[BB11] Amit Berman 和 Yitzhak Birk。受限闪存编程。载于《2011年IEEE国际信息论研讨会论文集》，第2128–2132页，2011年。[BGH+16] Meinolf Blawat、Klaus Gaedke、Ingo Huetter、Xiao-Ming Chen、Brian Turczyk、Samuel Inverso、Benjamin W Pruitt 和 George M Church。DNA数据存储的前向纠错。Procedia计算机科学，80:1011–1022，2016年。[BLC+16] James Bornholt、Randolph Lopez、Douglas M Carmean、Luis Ceze、Georg Seelig 和 Karin Strauss。基于DNA的档案存储系统。载于第二十一届编程语言与操作系统体系结构支持国际会议论文集，第637–649页，2016年。[EGYM21] Ohad Elishco、Ryan Gabrys、Eitan Yaakobi 和 Muriel M´edard。无重复码。《IEEE信息论汇刊》，67(9):5749–5764，2021年。[EZ17] Yaniv Erlich 和 Dina Zielinski。DNA喷泉实现稳健高效的存储架构。《科学》，355(6328):950–954，2017年。[HMG19] Reinhard Heckel、Gediminas Mikutis 和 Robert N Grass。DNA数据存储信道的特性。《科学报告》，9(1):1–12，2019年。[IC17] Kees A Schouhamer Immink 和 Kui Cai。面向基于DNA的存储系统的近容量受限码设计。《IEEE通信快报》，22(2):224–227，2017年。[IC20] Kees A Schouhamer Immink 和 Kui Cai。面向基于DNA的数据存储的受限码的性质与构造。IEEE Access，8:49523–49531，2020年。[Imm90] Kees A Schouhamer Immink。游程长度受限序列。《IEEE会刊》，78(11):1745–1759，1990年。[Imm04] Kees A Schouhamer Immink。海量数据存储系统编码。Shannon Foundation Publisher，2004年。[Imm22] Kees A Schouhamer Immink。受限编码的创新。《IEEE通信杂志》，60(10):20–24，2022年。[KLBLY22] Adir Kobovich、Orian Leitersdorf、Daniella Bar-Lev 和 Eitan Yaakobi。受限周期性编码。载于IEEE/IEICE信息论及其应用国际研讨会（ISITA），2022年。[Knu86] D. Knuth。高效平衡码。《IEEE信息论汇刊》，32(1):51–53，1986年。[LY18] Maya Levy 和 Eitan Yaakobi。用于DNA存储的互不相关码。《IEEE信息论汇刊》，65(6):3671–3691，2018年。

[Min88] H Minc.《非负矩阵》。Wiley-Interscience 出版物，1988年。[MK06] Olgica Milenkovic 和 Navin Kashyap。DNA计算编码的设计。载于《编码与密码学：国际研讨会，WCC 2005》，挪威卑尔根，2005年3月14日至18日。修订精选论文，第100–119页。Springer，2006年。[MRS01] Brian H Marcus、Ron M Roth 和 Paul H Siegel。《受约束系统编码导论》。讲义，2001年。[NCI20] Tuan Thanh Nguyen、Kui Cai 和 Kees A Schouhamer Immink。二进制子块能量受限码：Knuth平衡法与序列替换技术。载于2020年IEEE信息论国际研讨会（ISIT），第37–41页。IEEE，2020年。[NCK+23] Tuan Thanh Nguyen、Kui Cai、Han Mao Kiah、Duc Tu Dao 和 Kees A Schouhamer Immink。DNA计算编码的设计：二级结构规避码。arXiv预印本 arXiv:2302.13714，2023年。[OAC+18] Lee Organick、Siena Dumas Ang、Yuan-Jyue Chen、Randolph Lopez、Sergey Yekhanin、Konstantin Makarychev、Miklos Z Racz、Govinda Kamath、Parikshit Gopalan、Bichlien Nguyen等。大规模DNA数据存储中的随机访问。《自然·生物技术》，36(3)：242–248，2018年。[QYS14] Minghai Qin、Eitan Yaakobi 和 Paul H Siegel。可缓解闪存读写周期中单元间干扰的受限码。《IEEE通信领域精选期刊》，32(5)：836–846，2014年。[RL79] Jorma Rissanen 和 Glen G Langdon。算术编码。《IBM研究与开发期刊》，23(2)：149–162，1979年。[SB19] Jin Sima 和 Jehoshua Bruck。纠正多磁头赛道存储器中的删除。载于2019年IEEE信息论国际研讨会（ISIT），第1367–1371页，2019年。[SWZGY17] Clayton Schoeny、Antonia Wachter-Zeh、Ryan Gabrys 和 Eitan Yaakobi。纠正突发删除或插入的码。《IEEE信息论汇刊》，63(4)：1971–1985，2017年。[VW98] AJ Van Wijngaarden。《帧同步技术：Rahmensynchronisationsverfahren》。博士论文，杜伊斯堡-埃森大学，1998年。[VWI97] AJ Van Wijngaarden 和 Kees A Schouhamer Immink。采用序列替换技术构造受限码。载于《IEEE信息论国际研讨会论文集》，第144页。IEEE，1997年。[VWI10] AJ Van Wijngaarden 和 Kees A Schouhamer Immink。采用序列替换技术构造最大游程受限码。《IEEE通信领域精选期刊》，28(2)：200–207，2010年。[WSVDG+18] Sven Warris、Elio Schijlen、Henri Van De Geest、Rahulsimham Vegesna、Thamara Hesselink、Bas te Lintel Hekkert、Gabino Sanchez Perez、Paul Medvedev、Kateryna D Makova 和 Dick De Ridder。全基因组扩增后长读长序列中的回文纠正。《BMC基因组学》，19：1–12，2018年。