

# 基于 Harary 图生成树的部分重复码构造<sup>①</sup>



张鑫楠, 沈克勤, 孙 伟, 何亚锦

(长安大学 信息工程学院, 西安 710061)  
通讯作者: 张鑫楠, E-mail: 893789337@qq.com

**摘 要:** 针对部分重复码的有效修复问题, 本文基于 Harary 图生成树构造出了一种新型的部分重复 (Fractional Repetition based on Spanning trees of Harary graph, FRSH) 码. 实验结果表明, 相较于现有的里所 (Read-Solomon, RS) 码和简单再生码 (Simple Regeneration Codes, SRC), FRSH 码在修复带宽开销、修复局部性等方面得到了更低的开销, 且改善了修复效率, 并将故障节点的修复时间缩短.

**关键词:** 部分重复码; Harary 图; 生成树; 离心率

引用格式: 张鑫楠, 沈克勤, 孙伟, 何亚锦. 基于 Harary 图生成树的部分重复码构造. 计算机系统应用, 2021, 30(4): 241-246. <http://www.c-s-a.org.cn/1003-3254/7848.html>

## Fractional Repetition Codes Construction Based on Spanning Trees of Harary Graph

ZHANG Xin-Nan, SHEN Ke-Qin, SUN Wei, HE Ya-Jin

(School of Information Engineering, Chang'an University, Xi'an 710064, China)

**Abstract:** Aiming at the low efficiency in repairing failed nodes of fractional repetition codes, this study proposed a construction algorithm of Fractional Repetition codes based on Spanning trees of Harary graph (FRSH). As a result, FRSH codes have lower computational overhead in repairing bandwidth overhead and locality than RS codes and SRC. Besides, FRSH codes are more efficient and spend less time in repairing failed nodes, compared with the other two codes.

**Key words:** fractional repetition codes; Harary graph; spanning trees; eccentricity

目前, 将海量数据存储分布在分布式存储系统中的不同存储节点上的数据存储方式已在实际系统中得到了广泛应用, 如 Google 文件系统<sup>[1]</sup>、Hadoop 文件系统等. 为确保分布式存储系统中数据的可用性和可靠性, 通常采用诸如复制策略<sup>[2]</sup>或纠删码策略<sup>[3,4]</sup>的数据冗余策略. 以复制策略中的三副本复制为例, 三副本复制需要存储大量副本数据以确保系统较高的可靠性, 存储代价过高; 纠删码策略的提出使得修复造成的存储开销显著降低, 但其过大的修复带宽开销也成为了限制它的瓶颈.

2007 年, Dimakis 等人指出存储开销和修复带宽开销之间存在某种平衡, 平衡曲线上的点可通过再生码来实现<sup>[5]</sup>. 再生码基于网络编码的概念, 其故障节点

可通过连接一定数目的存活节点完成修复, 相比于纠删码降低了修复带宽开销. 目前再生码的研究主要集中在最小存储再生码和最小带宽再生码<sup>[6]</sup>.

El Rouayheb 和 Ramchandran 为进一步降低修复过程中的运算复杂和带宽开销, 提出了一种基于最小带宽再生点的精确修复编码-部分重复 (Fractional Repetition, FR) 码<sup>[7]</sup>. 部分重复码结合再生码和复制策略的优点, 有效减少了修复带宽开销和磁盘 I/O 开销<sup>[8]</sup>, 并实现精确的无编码修复.

目前, FR 码主要采用分组设计<sup>[9]</sup>、可分解设计<sup>[10,11]</sup>等方法进行构造. 朱兵等人基于分组设计提出一种重复度异构的 FR 码的构造<sup>[12]</sup>, 使得常用数据的备份更加充分, 且参数选取范围较大, 但同时也造成了较大的存

① 收稿时间: 2020-08-10; 修改时间: 2020-08-28; 采用时间: 2020-09-01; csa 在线出版时间: 2021-03-30

储开销; Natalia Silberstein 和 Tuvi Etzion 等人基于组合设计和正则图, 提出了一种达到最大码率的 FR 码的构造方法<sup>[13]</sup>, 但其重复度受 FR 码结构的约束, 不能适用于任意参数; Harout Aydinian 和 Holger Boche 等人则基于部分有序集构造了一种普遍好的 FR 码<sup>[14]</sup>, 这种构造方法虽然简便, 但其重复度却也十分受限. 为此, 本文基于 Harary 图生成树构造出了一种新型的部分重复 (Fractional Repetition based on Spanning trees of Harary graph, FRSH) 码, 可以在很大范围内选择构造参数和数据块的重复度, 还能修复多个故障节点. 通过调整 Harary 图的构造参数, 可以构造出不同重复度的 FR 码. 相比现有的 FR 码, 采用 Harary 图生成树设计 FR 码更加简洁直观. 相较于 RS 码和 SRC<sup>[15]</sup>, FRSH 码在修复带宽开销、修复复杂度以及修复局部性等方面得到了更低的开销, 且改善了修复效率, 并将故障节点修复时间缩短.

## 1 预备知识

### 1.1 Harary 图的基础知识

Harary 图是一种正则图, 定义为  $H_{k,m}$ , 其中  $k$  为每个节点所邻接的节点个数, 即顶点的度;  $m$  为顶点个数. 根据  $k$  和  $m$  的取值, 可分为 3 种情况构造 Harary 图.

(1)  $k$  是偶数

设  $k = 2r$ , 则  $H_{2r,m}$  构造如下: 先给出它的顶点  $0, 1, 2, \dots, i, \dots, j, \dots, m-1$ , 然后连接所有满足  $|i - j| \leq r$  的顶点, 即可完成  $H_{2r,m}$  的构造.

(2)  $k$  是奇数,  $m$  是偶数

设  $k = 2r + 1$ , 则  $H_{2r+1,m}$  构造如下: 首先构造出  $H_{2r,m}$ , 为满足  $k = 2r + 1$ , 还需添加一些边: 这些边连接了每个顶点  $i$  与顶点  $i + \frac{m}{2}$  ( $0 \leq i < \frac{m}{2}$ ), 连接完即可得到  $H_{2r+1,m}$ .

(3)  $k$  是奇数,  $m$  是奇数

设  $k = 2r + 1$ , 此时的  $H_{2r+1,m}$  构造如下: 首先构造出  $H_{2r,m}$ , 为满足  $k = 2r + 1$ , 需要添加一些边: 这些边连接了顶点  $0$  与顶点  $\frac{m-1}{2}$  和  $\frac{m+1}{2}$ , 再将每个顶点  $i$  连到顶点  $i + \frac{m+1}{2}$  ( $0 \leq i < \frac{m-1}{2}$ ), 如此连接即可完成构造.

图 1 所示是构造的  $k=4, m=8$  的 Harary 图  $H_{4,8}$ .

### 1.2 生成树的基础知识

无圈图是指不包含圈的图, 连通的无圈图定义为树. 无圈图的生成树是指顶点与无圈图相同、边集是无圈图的边集子集, 且含边数最少的连通子图. 图 2 中

$T_1$  是完全图  $K_8$  的生成树, 它包含  $K_8$  中的所有 9 个顶点, 边集是完全图  $K_8$  边集的子集, 且包含了最小数目的边数.

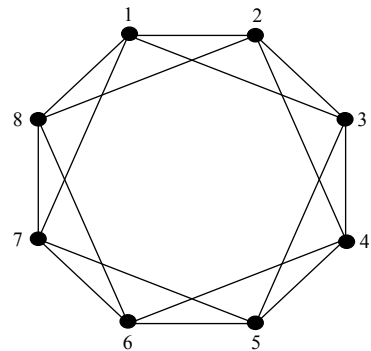


图 1 Harary 图  $H_{4,8}$

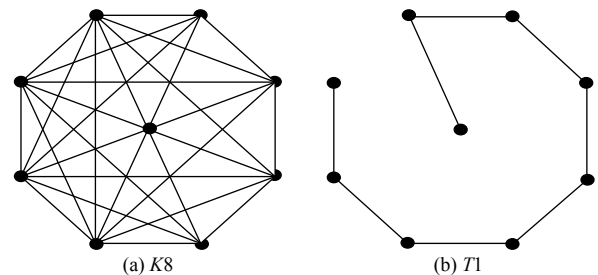


图 2 完全图  $K_8$  和它的生成树  $T_1$

由构造的 Harary 图  $G = H_{k,m}$ , 以顶点 1 为起始顶点得出生成树. 具体构造步骤如下<sup>[16]</sup>:

步骤 1. 画出边  $(1, v+1)$  和  $(1, m-v+1)$ ,  $v=1, 2, \dots, \frac{k}{2}$ .

步骤 2. 令  $p = \frac{k}{2} + 1, j = \frac{k}{2}$ , 画出边  $(m-p+1, m-p-j+1)$ ,  $(m-p-j+1, m-p-2j+1)$ ,  $\dots$ , 直到形成从顶点  $m-p+1$  到顶点 1 的路径时停止.

步骤 3. 令  $q=p+1$ , 重复步骤 2 画出边  $(m-q+1, m-q-j+1)$ ,  $(m-q-j+1, m-q-2j+1)$ ,  $\dots$ , 当存在从顶点  $m-p+1$  到顶点 1 的路径时停止, 否则转至步骤 1.

图 3 为基于  $H_{4,8}$ , 且以顶点 1 为起始顶点构造的生成树.

### 1.3 离心率的基础知识

对于连通图  $G$  的一个顶点  $v$ ,  $v$  的离心率定义为  $v$  到  $G$  中除  $v$  外所有顶点的距离的最大值. 例如图 4 中顶点 1 的离心率值为 3, 由于距离顶点 1 最长的顶点是顶点 6, 而从顶点 1 到顶点 6 的路径中共经过 3 个顶点, 即距离为 3, 故顶点 1 的离心率, 其它顶点同理.  $G$  的半径定义为  $G$  中所有顶点离心率的最小值, 直径则

定义为  $G$  的最大离心率, 分别记为  $\text{rad}(G)$  和  $\text{diam}(G)$ . 在图 4 中顶点最小离心率为 2, 最大离心率为 4, 所以有  $\text{rad}(G)=2, \text{diam}(G)=4$ .

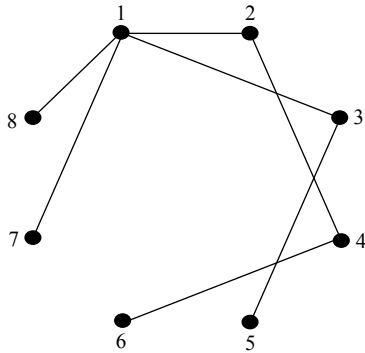


图 3 基于  $H_{4,8}$  的生成树

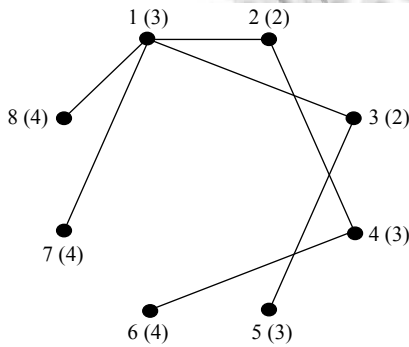


图 4 离心率参考图

## 2 基于 Harary 图生成树的部分重复码构造

### 2.1 构造方法

基于 Harary 图生成树与离心率, 给出重复度为  $\rho$  的 FRSH 码的构造算法, 具体步骤如算法 1.

算法 1. 基于 Harary 图生成树的部分重复码构造

- 1) 给定一个 Harary 图的设计参数  $k$  和  $m$ , 其中  $k$  为每个节点所连接的节点个数, 即节点的度,  $m$  为顶点个数,  $m > 0$ . 本文暂且只考虑  $k$  为偶数的情况.
- 2) 根据给出的设计参数  $k$  和  $m$  构造 Harary 图.
- 3) 选定 Harary 图上任一顶点作为顶点 1, 并从顶点 1 开始给 Harary 图的顶点顺时针编号, 每个顶点存储与其编号相同的数据块.
- 4) 由构造的 Harary 图以顶点 1 为起始顶点构造生成树.
- 5) 对得到的生成树的顶点按离心率分组, 将第一个生成树  $G$  的顶点按离心率分为  $n (n \geq 1)$  组, 分组后每组顶点的离心率相同, 并将各组顶点按编号从小到大的顺序分别写入  $n$  个节点中, 得到第一组节点.
- 6) 按照所需构造部分重复码的重复度  $\rho (\rho \geq 1)$  更换起始顶点, 重复步骤 4) 和步骤 5) 再得到  $\rho-1$  个生成树, 并将它们的顶点按离心率分组后分别写入  $n * (\rho-1)$  个新的节点中.

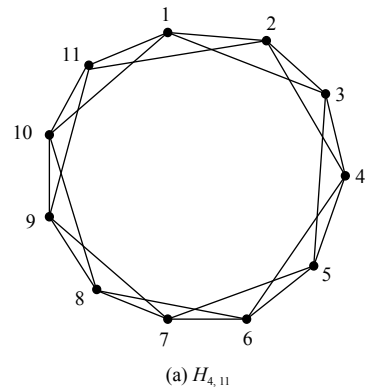
7) 由生成树顶点按离心率的分组得到关联矩阵  $M=(m_{s,t})(1 \leq s \leq n\rho, 1 \leq t \leq m)$ . 关联矩阵构造公式如下 ( $v_t$  代表生成树中的任一顶点,  $e_s$  代表任一离心率取值):

$$m_{s,t} = \begin{cases} 1, & \text{若 } v_t \text{ 与 } e_s \text{ 相关联} \\ 0, & \text{其它情况} \end{cases}$$

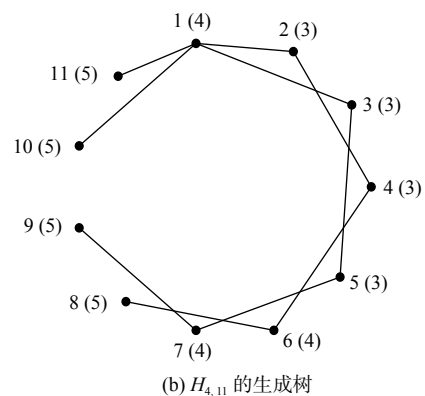
再将生成树的关联矩阵等价于 FR 码的关联矩阵, 关联矩阵的行向量对应 FR 码的存储节点, 列向量对应 FR 码的编码块. 行向量的重表示表示节点存储容量  $d$ , 列向量的重表示编码块重复度  $\rho$ . 由此即可得到重复度为  $\rho$  的 FR 码.

综上, 重复度为  $\rho$  的 FRSH 码的构造过程中, 其包含的不同数据块个数即为 Harary 图的顶点数  $m$ ; 重复度  $\rho$  由生成树个数决定; 包含的节点个数为  $n\rho$ .

具体地, 我们以构造  $\rho = 3$  的 FR 码的过程为例说明构造方法. 令  $k=4, m=11$ , 构造 Harary 图  $H_{4,11}$ , 并给其顶点编号, 如图 5(a) 所示. 再由构造的 Harary 图以顶点 1 为起始顶点得出第一个生成树, 如图 5(b) 所示.



(a)  $H_{4,11}$



(b)  $H_{4,11}$  的生成树

图 5  $H_{4,11}$  和它的生成树

将图 5(b) 中给出的生成树的顶点按离心率分组, 相同离心率顶点对应的数据块写入同一分组. 为满足重复度  $\rho = 3$ , 需重复步骤 4) 和步骤 5), 分别以顶点 3 和顶点 5 为起始顶点再得到第 2 个生成树和第 3 个

生成树(此处图略). 进一步将第2个生成树和第3个生成树的顶点存储数据按离心率分组, 构造关联矩阵  $M$ . 以图 5(b) 中的生成树为例, 本例中所有生成树顶点的离心率  $e_s$  共有  $n=3$  个取值:  $e_1=3$ 、 $e_2=4$ 、 $e_3=5$ . 图 5(b) 中顶点 2、3、4、5 的离心率都为  $e_1=3$ , 即  $v_2$ 、 $v_3$ 、 $v_4$ 、 $v_5$  与  $e_1$  相关联, 故有  $m_{1,2}=m_{1,3}=m_{1,4}=m_{1,5}=1$ , 以此类推便得到关联矩阵  $M$ .

$$M = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

根据关联矩阵  $M$ , 构造重复度  $\rho=3$  的 FR 码, 构成的 FR 码如图 6.

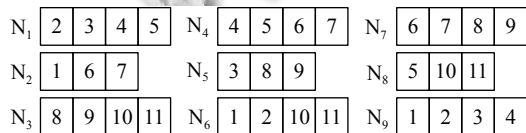


图 6 FR 码

### 2.2 故障节点修复

本论文构造的任意 FRSH 码都包含  $\rho$  个平行类, 且最多可容忍  $\rho-1$  个节点故障, 具体修复方案如下:

(1) 当单个节点失效时, 新生节点从存活平行类中下载失效节点对应数据块, 即可完成修复. 如节点  $N_1$  故障时, 损坏的数据块 2、3、4、5 可从剩余两个平行类  $N_4$ 、 $N_5$ 、 $N_6$  或  $N_7$ 、 $N_8$ 、 $N_9$  中下载. 这里选择连接存活节点  $N_4$ 、 $N_9$  并分别下载 4、5、2、3, 即可修复故障节点  $N_1$ .

(2) 当多个节点同时出现故障, 由于仍至少存在一个平行类包含全部数据块, 而本方法构造的 FR 码的任一平行类中包含的节点个数等于离心率分组数  $n$ , 故至多连接  $n$  个节点即可完成修复. 具体地, 当多个故障节点所包含的数据块个数  $d < \alpha_{\min} + \alpha_{\max}$  (这里  $\alpha_{\min}$  和  $\alpha_{\max}$  分别表示 FRSH 码中节点存储的最小数据块个数和最大数据块个数) 时, 仅需连接  $n-1$  个故障节点即可完成修复. 例如当  $N_2$  和  $N_5$  同时失效时, 故障节点  $N_2$  和  $N_5$  共包含  $d=6$  个数据块, 满足  $d < \alpha_{\min} + \alpha_{\max}$ . 此时, 修复故障节点  $N_2$  和  $N_5$  需要连接  $n-1=2$  个存活节点

完成修复. 这里选择存活节点  $N_7$  和  $N_9$ , 并分别从存活节点  $N_7$  和  $N_9$  下载数据块 6、7、8、9 和 1、3, 实现故障节点  $N_2$  和  $N_5$  的修复.

(3) 当多个节点同时出现故障, 且多个故障节点所包含的数据块个数  $d \geq \alpha_{\min} + \alpha_{\max}$  时, 需连接  $n-1$  或  $n$  个不同节点即可完成故障节点修复. 同样先考虑图 6 中的节点  $N_4$  和  $N_5$  发生故障, 此时故障节点  $N_4$  和  $N_5$  包含  $d=7$  个数据块, 满足  $d \geq \alpha_{\min} + \alpha_{\max}$ . 为修复故障节点  $N_4$  和  $N_5$  需要从  $n-1=2$  个存活节点中下载对应数据块, 这里选择存活节点  $N_1$  和  $N_7$ , 并分别从  $N_1$  和  $N_7$  下载数据块 3、4、5 和 6、7、8、9, 即可修复故障节点  $N_4$  和  $N_5$ . 另当  $N_2$  和  $N_7$  发生故障时, 此时故障节点  $N_2$  和  $N_7$  包含  $d=7$  个数据块, 同样满足  $d \geq \alpha_{\min} + \alpha_{\max}$ . 而此时修复  $N_2$  和  $N_7$  需连接  $n=3$  个存活节点  $N_4$ 、 $N_5$  和  $N_6$ , 并分别下载 6、7、8、9 和 1 便可修复  $N_2$  和  $N_7$ .

综上, 对于重复度为  $\rho$  的 FRSH 码, 修复单节点故障时在剩余平行类中连接对应存活节点 (连接的节点数至多为  $n-1$  个) 下载所需数据块即可; 修复多节点故障时, 当多个故障节点所包含的数据块个数  $d < \alpha_{\min} + \alpha_{\max}$  时, 仅需连接  $n-1$  个故障节点即可完成修复; 当多个故障节点所包含的数据块个数  $d \geq \alpha_{\min} + \alpha_{\max}$  时, 需连接  $n-1$  或  $n$  个不同节点即可完成故障节点修复.

### 3 性能分析

基于 Harary 图生成树构造的 FR 码的性能分析主要集中在修复局部性、修复带宽开销和运算复杂度这几方面, 并将其与传统的 RS 码和简单再生码 (Simple Regenerating Codes, SRC) 进行性能比较. 表 1 给出了 SRC、RS 码与 FRSH 码在一般情况下的节点存储开销、修复带宽开销以及修复局部性的计算公式, 之后我们会在给定具体文件大小和编码数据等条件下用柱状图的形式把上文构造的  $\rho=3$  的 FR 码与 SRC、RS 码的各种性能逐一进行比较.

表 1 几种编码方案的性能分析

编码方案	SRC	(n, k)RS码	FRSH码
节点存储开销	$(f+1)M/k$	$M/k$	$2M/k$
修复带宽开销	单节点故障	$M$	$2M/k$
	两节点故障	$2(f+1)M/k$ 或 $M/k$	$M$
修复局部性	单节点故障	$k$	2
	两节点故障	$k$	$\leq 3$

### 3.1 修复局部性

修复局部性是衡量构造出的部分重复码性能的重要指标之一,其定义为节点故障修复过程中连接的存活节点数目,即磁盘 I/O 开销.此处为方便比较,我们假设原文件大小  $M=1000$  Mb,存储节点数为  $n=11$ ,SRC 子文件数  $f=3$ ,RS 码和 SRC 的原文件重构度为  $k=8$ ,FRSH 码外部采用 (11,8)MDS 编码.本小节仅考虑两种节点故障情况:单节点故障与两节点故障.

当单节点出现故障时,SRC 在修复失效节点时需要连接  $2f$  个节点,这里取  $f=3$ ,则 SRC 的修复局部性为 6;若采用 (11,8)RS 码,需要连接  $k=8$  个节点以恢复出完整的原文件,再由原文件修复故障节点,故其修复局部性为 8;而本文构造的 FRSH 码需要连接 2 个节点来修复故障节点,修复局部性为 2.

在两节点故障的情况下,SRC 与 (11,8)RS 码都需要连接  $k=8$  个存活节点以修复故障节点,故它们的修复局部性都为 8;而基于 Harary 图生成树构造的 FRSH 码的修复局部性为 2 或 3,为便于对比,此处取恒为 3 作为比较.由图 7 可见,在单节点故障与两节点故障情况下,本文构造的 FRSH 码的修复局部性都优于 SRC 和 RS 码.

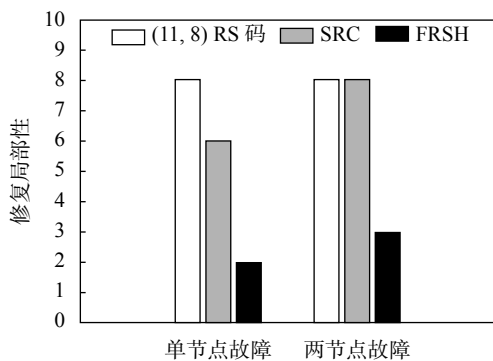


图 7 修复局部性

另与基于图因子构造的部分重复码相比,在单节点故障时两种部分重复码都需要连接两个节点进行修复;在两节点故障时,基于图因子的部分重复码需要连接 4 个节点进行修复,而 FRSH 码至多需连接 3 个节点进行修复,可见在两节点故障时 FRSH 码具有显著优势.

### 3.2 修复带宽开销

在单节点故障的情况下,SRC 需下载  $f$  个数据块以完成修复,而每个数据块的大小为  $M/fk$ ,因此 SRC 的

带宽开销为  $(f+1)M/k$ ;由于 RS 码在单节点故障的情况下完成修复需下载整个原文件,故 RS 码的带宽开销为原文件大小  $M$ ;而本文构造的 FRSH 码在单节点故障的情况下需要通过连接 2 个存活节点完成修复,因此 FRSH 码的带宽开销为  $2M/k$ .

当两节点同时故障时,RS 码与 SRC 的带宽开销均为  $M$ .FRSH 码的带宽开销为  $3M/k$ .

假设原文件大小为  $M=1000$  Mb,存储节点数  $n=11$ ,SRC 子文件数  $f=3$ ,RS 码和 SRC 的原文件重构度为  $k=8$ ,FRSH 码外部采用 (11,8)MDS 编码.当单节点故障时,RS 码的带宽开销为 1000 Mb,SRC 的带宽开销为 500 Mb,FRSH 码的带宽开销为 250 Mb;当两个节点故障时,(11,8)RS 码修复带宽开销为 1000 Mb,SRC 的带宽开销同样为 1000 Mb,为方便对比 FRSH 码的带宽开销取较大值  $3M/k=375$  Mb.如图 8 所示,无论单节点还是两节点故障,FRSH 码的修复带宽开销都相对较低.

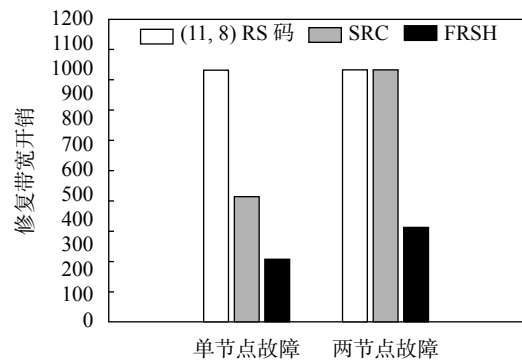


图 8 修复带宽开销对比

### 3.3 构造算法运算复杂度

除去修复局部性与修复带宽开销这两种直观性能,构造算法复杂度体现了算法在执行时的难易程度,是衡量一个算法好坏的重要指标.所谓运算复杂度,即将算法写成程序在实际的计算机系统中运行时涉及的计算量.本文构造的 FRSH 码构造时不需要任何计算量,只需调节 Harary 图与其生成树的参数即可.相较于可达到同等性能的 FRC,减少了大量的运算时间.

将基于 Harary 图生成树的 FRSH 与运用图因子分解构造的 FRC<sup>[17]</sup>在构造算法运算复杂度进行对比分析,在文献 [17] 中,构造算法时运用到了加法,乘法运算和基于一次方程上的运算共 3 种运算方式,相较于本文仅需调整 Harary 图参数,可知本文的构造算法运算复杂度更优.

## 4 结论与展望

针对部分重复码的有效修复问题,本文基于 Harary 图生成树构造出了一种新型的部分重复码. 这种 FRSH 码在构造方面的优势在于参数选择范围较广且运算复杂度极低. 在性能方面,实验结果表明,相较于现有的 RS 码和 SRC, FRSH 码在修复带宽开销、修复局部性等方面得到了更低的开销,且改善了修复效率,并将故障节点的修复时间缩短;又与可达到类似性能的图因子分解的部分重复码在运算复杂度方面进行比较,得出本文的构造算法运算复杂度更优.

### 参考文献

- 1 Shafer J, Rixner S, Cox AL. The hadoop distributed filesystem: Balancing portability and performance. 2010 IEEE International Symposium on Performance Analysis of Systems & Software (ISPASS). White Plains, NY, USA. 2010. 122–133.
- 2 Liu Y, Vlassov V. Replication in distributed storage systems: State of the art, possible directions, and open issues. 2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. Beijing, China. 2013. 225–232.
- 3 Silberstein M, Ganesh L, Wang Y, *et al.* Lazy means smart: Reducing repair bandwidth costs in erasure-coded distributed storage. Proceedings of International Conference on Systems and Storage. New York City, NY, USA. 2014. 1–7.
- 4 Papailiopoulos DS, Dimakis AG, Cadambe VR. Repair optimal erasure codes through hadamard designs. IEEE Transactions on Information Theory, 2013, 59(5): 3021–3037. [doi: 10.1109/TIT.2013.2241819]
- 5 Dimakis AG, Godfrey PB, Wainwright MJ, *et al.* Network coding for distributed storage systems. 26th IEEE International Conference on Computer Communications (INFOCOM 2007). Spain. 2007. 2000–2008.
- 6 Rawat AS, Silberstein N, Koyluoglu OO, *et al.* Optimal locally repairable codes with local minimum storage regeneration via rank-metric codes. 2013 Information Theory and Applications Workshop (ITA). San Diego, CA, USA. 2013. 1–8.
- 7 El Rouayheb S, Ramchandran K. Fractional repetition codes for repair in distributed storage systems. 2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton). Allerton, IL, USA. 2010. 1510–1517.
- 8 Rai BK, Dhoorjati V, Saini L, *et al.* On adaptive distributed storage systems. 2015 IEEE International Symposium on Information Theory (ISIT). Hong Kong, China. 2015. 1482–1486.
- 9 Wang J, Wang TT, Liu XY, *et al.* Locally repairable codes based on fractional repetition codes in distributed storage systems. 14th International Conference on Wireless Communications, Networking and Mobile Computing. Lancaster, UK. 2018. 578–587.
- 10 Olmez O, Ramamoorthy A. Repairable replication-based storage systems using resolvable designs. 2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton). Monticello, IL, USA. 2012. 1174–1181.
- 11 Olmez O, Ramamoorthy A. Constructions of fractional repetition codes from combinatorial designs. 2013 Asilomar Conference on Signals, Systems and Computers. Pacific Grove, CA, USA. 2013. 647–651.
- 12 Zhu B, Li H, Hou HX, *et al.* Replication-based distributed storage systems with variable repetition degrees. 2014 Twentieth National Conference on Communications (NCC). Kanpur, India. 2014. 1–5.
- 13 Silberstein N, Etzion T. Optimal fractional repetition codes based on graphs and designs. IEEE Transactions on Information Theory, 2015, 61(8): 4164–4180. [doi: 10.1109/TIT.2015.2442231]
- 14 Aydinian H, Boche H. Fractional repetition codes based on partially ordered sets. 2017 IEEE Information Theory Workshop (ITW). Kaohsiung, China. 2017. 51–55.
- 15 Papailiopoulos DS, Luo JQ, Dimakis AG, *et al.* Simple regenerating codes: Network coding for cloud storage. 2012 Proceedings IEEE INFOCOM. Orlando, FL, USA. 2012. 2801–2805.
- 16 Kashyap N, Mukherjee M, Sankarasubramanian Y. On the secret key capacity of the Harary graph PIN model. 2013 National Conference on Communications (NCC). New Delhi, India. 2013. 1–5.
- 17 余春雷, 王静, 王秘, 等. 图因子分解的部分重复码构造. 中国科技论文, 2019, 14(11): 1260–1264.