An Easy-To-Implement Construction of Ruzsa–Szemerédi Graph

Chenyang Cao[†]

Project advisor: Michael Kapralov

Abstract. The researchers call a graph G an (r, t)-Ruzsa-Szemerédi graph if its edge set can be partitioned into t edge-disjoint induced matchings, and each matching has size r. In this paper, I will summarize some construction methods of (r, t)-Ruzsa-Szemerédi graph, implement them into algorithms and evaluate their complexity. Then I will give a new construction method of bipartite Ruzsa-Szemerédi graph that is easy to implement with parameters $r = \Theta(\log N)$ and $t = N/2$.

1. Introduction.

1.1. Background. Ruzsa–Szemerédi graph was firstly introduced by Ruzsa and Szemerédi in their paper [13]. They use this graph to solve the $(6, 3)$-problem: What's the maximum number of edges in a graph such that each edge in the graph belongs to a unique triangle. Equivalently, such a problem can be also shown as: What's the maximum number of edges in a bipartite graph such that the edges can be partitioned into several induced matchings. The definition of Ruzsa–Szemerédi graph in recent research is based on the latter problem: A Ruzsa–Szemerédi graph is a graph with an edge set which can be partitioned into several induced matchings. To simplify, I will use RS graph instead of Ruzsa-Szemeredi graph in the rest of this paper.

After the graph is introduced to solve the $(6, 3)$-problem, researchers found it can also be used in many other problems. [4] uses the RS graph to solve the problem of communicating over a shared directional multi-channel. [8], [10] and [3] uses such graph to prove the bound of streaming complexity of maximum matching problem. And [9] connect the RS graph with Linearity Testing. Besides, there are still many applications of RS graphs in many different areas such as Combinatorial Mathematics, Information Theory, and Computer Science.

Although the application of RS graph is varied, there exists a core problem about such a graph: Does the RS graph with specific parameters really exists? By its definition, one can't directly show the existence of the RS graph with some given parameters r and t. Thus one of the main routes of the research about RS graphs in the past decades is to construct dense RS graphs with specific parameters. In section 3, I will talk about these construction methods.

What's more, although there are many different constructions of RS graphs, most of them only prove the existence of such graphs. Many proofs use some combinatorial conclusions, which are NP-hard for the computer to solve. Assume we can solve the problem in some good ways, the algorithm to construct such a graph may have very high time and space complexity that we can not afford. However, some test sets are necessary for the simulation and experiment of the algorithms related to RS graph. Thus building an easy-to-implement construction of the RS graph is important but poorly researched. In this paper, I will do some analysis of the complexity of the implements, and try to build a good construction based on

[†]School of Computer and Communication Sciences, The École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland (chenyang.cao@epfl.ch)

this criterion.

1.2. Structure of this paper. This paper is organized as follows. In section 2, I will show some notations and definitions to be used in this paper. In section 3, there will be a short survey of current construction methods of RS graph and some algorithms to generate RS graphs based on these construction methods. Then I will introduce my new construction method in section 4. And finally there will be some conclusions in section 5.

2. Preliminaries. In this section, I will introduce some notations and definitions used in the rest of this paper. The most important one is the definition of RS graph, or the (r, t)-RS graph. Like most papers do, I will use $G = (V, E)$ to describe a graph, in which V is the vertex set of the graph, and E is its edge set. Because the edge direction, the repeat edge as well as the self-loop are not necessary for the definition of a RS graph, I will assume all graphs in this paper are undirected, and have no repeat edge and self-loop. In order to give a precise definition to the Ruzsa-Szemerédi graph, let's first define what is an induced matching:

Definition 2.1 (Matching). *Given a graph $G = (V, E)$, a **matching** is an edge subset $M \subseteq E$ such that each vertex belongs to at most one edge in M.*

Definition 2.2 (Induced Matching). *Given a graph $G = (V, E)$, an **induced matching** is a matching M such that the rest of edges $E \setminus M$ contains no edge connecting the vertices in the matching M.*

As mentioned in the introduction section, the RS graph can be decomposed into several induced matchings. However, such decomposition can be trivial: each edge set only containing a single edge is an induced matching. So to avoid the trivial decomposition, we use some parameters to define the RS graph:

Definition 2.3 ((r, t)-Ruzsa–Szemerédi Graph). *A graph $G = (V, E)$ is a (r, t)-Ruzsa–Szemerédi Graph if it can be partition into t induced edge-disjoint matchings $M_1, M_2, \cdots, M_t$ such that each matching has at least r edges.*

There are some comments on the definition of (r, t)-RS graph:

- **The number of vertex** n: It seems to be a very important parameter that can influence the density of a graph directly. However, most papers about the construction of RS graph talks about the asymptotic behavior of the graph as $n \to \infty$, so the value of r and t is in the format of a bound related to n. In this way, the definition of RS graph does not need to take n as a direct parameter.
- **The bipartite graph**: Based on the application of the target RS graph, some papers ([6] and [8]) choose to construct the bipartite Ruzsa–Szemerédi graph instead of the ordinary ones. One can still uses such (r, t) parameter to describe a bipartite RS graph except for a small modification: the meaning of n may be the size of the disjoint vertex subsets, but not the whole vertex set.
- **Other parameters**: Besides the edge number and the matching number, there are also some other parameters can be used to show how dense the Ruzsa–Szemerédi graph is. The edge density $\epsilon = r/n$ of each matching and the total edge number rt are the other widely used parameters. One can easily transform the parameter (r, t) into the target format and make the alternative definition based on such parameters.

In the construction, we may also talk about some sequences and subsequences. Here we define two useful sets of n integers:

$$[n] = \{1, 2, \cdots, n\}$$
$$\langle n \rangle = \{0, 1, \cdots, n-1\}$$

Then we can define a sequence with length m as:

$$[n]^m = \{(x_1, \cdots, x_m) : x_i \in [n], i \in [m]\}$$
$$\langle n \rangle^m = \{(x_1, \cdots, x_m) : x_i \in \langle n \rangle, i \in [m]\}$$

Given a sequence x and one of its subsequence $x' \subseteq x$, we can define a mask to show which elements in the sequence are included in the subsequence:

Definition 2.4 (Mask). *Given a sequence x of length m, the mask u of the subsequence $x' \subseteq x$ is defined as a binary sequence u of length m such that for any $i \in [m]$:*

$$u_i = \begin{cases} 1 & x_i \in x' \\ 0 & x_i \notin x' \end{cases}$$

Then use a mask vector, we can describe some property of the corresponding subsequence (or subset). Here we use the Hamming distance and Hamming weight to describe the size of a subsequence or the intersection of two subsequences.

Definition 2.5 (Hamming Distance). *Given two vectors $x, y \in \langle q \rangle^n$, the Hamming distance between them is defined as:*

$$d_H(x, y) = |\{i \in [n] : x_i \neq y_i\}|$$

Definition 2.6 (Hamming Weight). *The Hamming weight of a vector $x \in \langle q \rangle^n$ is the Hamming distance between it and the all-zero vector, denoted as $w(x) = d_H(x, \mathbf{0})$*

Given x is a sequence of size n and $x' \subseteq x$ is a subsequence with mask $u \in \{0, 1\}^n$. By counting the 1's in the mask, i.e. the Hamming weight of the mask, we can get the size of the subsequence x':

$$|x'| = w(u)$$

Consider the intersection of two subsequences $y, z \subseteq x$ and their corresponding masks $u, v \in \{0, 1\}^n$:

$$|y| + |z| = |y \setminus z| + |z \setminus y| + 2|y \cap z|$$

By the definition of Hamming distance, we have:

$$w(u) = |y|$$
$$w(v) = |z|$$
$$d_H(u, v) = |y \setminus z| + |z \setminus y|$$

And thus:

$$|y \cap z| = \frac{1}{2}(w(u) + w(v) - 2d_H(u, v))$$

3. Construction Methods.

3.1. Summary of the construction methods. After reviewing some papers of the related research, I collect some construction methods for RS graph and their corresponding parameters, which are shown in Table 1.

Table 1
Summary of the constructions

Paper	**Matching size** r	**Matching Number** t	**Edge Number** rt
Ruzsa, Szemerédi (1978)[13]	$n/\exp\{O(\sqrt{\log n})\}$	$n/3$	$*$
Birk, Linial, Meshulam (1993)[4]	$(\log n)^{\Omega(\log\log n/\log\log\log n^2)}$	$*$	$n^2/24$
Fischer et. al. (2002)‡[6]	$n/3 - o(n)$	$n^{\Omega(1/\log\log n)}$	$n^{1+\Omega(1/\log\log n)}$
Goel, Kapralov, Khanna(2012)‡[8]	$n/2 - o(n)$	$n^{\Omega(1/\log\log n)}$	$n^{1+\Omega(1/\log\log n)}$
Kapralov(2013)‡[10]	$n/2 - o(n)$	$n^{\Omega(1/\log\log n)}$	$n^{1+\Omega(1/\log\log n)}$
Alon, Moitra, Sudakov (2013)[1]	$n^{1-o(1)}$	$*$	$(1-o(1))\binom{n}{2}$
Jacob, Hao, Benny (2017)[7]	$n/4 - o(n)$	$O(n/\log n)$	$*$
Alon, Shraibman (2020)[2]	$*$	$n^{1+O(1/\log\log n)}$	$\Omega(n^2/\log n)$

$*$ means the construction doesn't take the parameter into consideration directly
‡ means the bipartite RS graph construction

3.2. Original construction of Ruzsa and Szemerédi [13].

3.2.1. 3-Term Arithmetic Progression. Before we start the construction, let's first talk about 3-Term Arithmetic Progression.

Definition 3.1 (3-Term Arithmetic Progression). *A 3-Term Arithmetic Progression is a sequence consists of 3 numbers (x, y, z) such that $y - x = z - y$, or equivalently $x + z = 2y$.*

Donote 3-Term Arithmetic Progression as 3AP in the rest of paper for simplify. Then define the 3AP-Free set:

Definition 3.2 (3AP-Free Set). *Given a positive integer n, the set $S \subseteq [n]$ is said to be 3AP-free if any three different numbers in S are not 3AP.*

The maximum size of 3AP-Free set on $[n]$ is a hard problem that has been discussed a lot. And here is a useful bound for the size of 3AP-Free set:

Theorem 3.3 (Behrend (1946)). *Let S be a 3AP-Free set on $[n]$ and $v(n)$ be the maximum number of distinct integers in S, then $v(n) = n/e^{O(\sqrt{\log n})}$*

3.2.2. Construction. Based on the 3AP-Free set, Ruzsa and Szemerédi construct the graph in which every edge belongs to a unique triangle, and such a graph is equivalent to a RS graph. Here I choose to show a construction in [11] which uses the same idea as Ruzsa and Szemerédi and directly outputs the RS graph.

Then prove the follow Theorem to show that Algorithm 3.1 is correct:

Algorithm 3.1 Construction by Ruzsa and Szemerédi

Input a integer number n
Input the 3-term arithmetic progression free set $T \subseteq [n]$
Define the vertex set $A := [3n]$, $B := [3n]$
Define the collection of matchings $\mathcal{M} := \{\}$
for all $x \in [n]$ **do**
  Define a matching $M_x = \{\}$
  for all $a \in T$ **do**
    Update $M_x := E \cup \{(x + a, x + 2a)\}$
  end for
  Update $\mathcal{M} := \mathcal{M} \cup \{M_x\}$
end for
Compute the edge set $E = \cup_{x \in [n]} M_x$
return $G := (A, B, E)$ and $\mathcal{M}$

Theorem 3.4 (Correctness of Algorithm 3.1). *The matchings in $\mathcal{M}$ returned by the Algorithm 3.1 are induced. The graph generated by Algorithm 3.1 is indeed RS graph.*

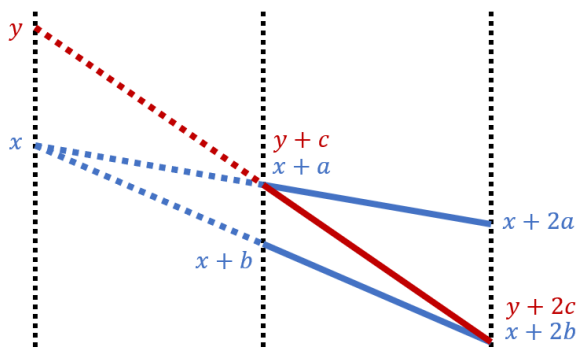

Figure 1. *Main idea of proving Theorem 3.4*

Proof. Assume there exists $x \in [n]$ such that M_x is not induced.That means there exists $a, b \in T$ such that edge $(x + a, x + 2b)$ belongs to another matching M_y. Thus there exists $c \in T$ such that $x + a = y + c$ and $x + 2b = y + 2c$. Combine the two equation, we have $x - y = c - a = 2c - 2b$ and then we have $a + c = 2b$. As a result (a, b, c) is a 3-term arithmetic progression, which contradicts the fact that T is 3-term arithmetic progression free. ∎

From the Algorithm 3.1, we can get a bipartite graph with $3n$ vertices in each vertex set and can be decomposed into n matchings with $|T| = n/e^{O(\sqrt{\log n})}$ edges in each matching. In other words, it's a bipartite $(N/\exp\{O(\sqrt{\log N})\}, N/3)$-RS graph.

3.2.3. Analysis about the implement. By Algorithm 3.1, if we have a 3AP-free set, we can easily get a RS graph. However, a big 3AP-free set is hard to build. [5] shows a table of 3AP-free sequence with $n \leq 123$. For some bigger n, although there are some algorithms to

generate a 3AP-free set, most of them are very time consuming. Because the size of vertex set is directly related to the parameter n, we can not generate a large graph based on this algorithm.

3.3. Coloring constructions. [6][8][10] Because of the similarity of this family of construction methods, I will give a formal description of them and then show their different configuration.

3.3.1. Fixed-weight Hamming Code. This family of construction is based on the existence of a collection of subsets with good property. Consider a given set $[n]$, there are 2^n subsets of it in total. Given integers $t < w < n$, one can select a collection $\mathcal{C}$ of subsets such that:

- Any subset $S \in \mathcal{C}$ has fixed size: $|S| = w$
- Any two subsets $S_1, S_2 \in \mathcal{C}$ has an intersection set with small size $|S_1 \cap S_2| \leq t$

Then use the binary mask to describe the subsets above, we can define such collection with good property as fixed-weight Hamming code:

Definition 3.5 (Fixed-Weight Hamming Code). *Given vector length n, weight w and distance d, then (n, w, d) fixed-weight Hamming code is defined as a collection of vectors $\mathcal{C} \subseteq \{0,1\}^n$ such that:*

- *For any $u \in \mathcal{C}$, $w(u) = w$*
- *For any different $u, v \in \mathcal{C}$, $d_H(u, v) \geq d$*

Here we use the Hamming distance to describe the intersection size:

$$d = 2\,(w - t)$$

By Gilbert-Varshamov bound and its corollary in [12], we have a fact about the size of the code:

Theorem 3.6 (Levenvstein(1971)). *For (n, w, d)-Hamming code with fixed weight, if $d \leq \frac{2w(n-w)}{n}$ holds, the code size $|C|$ has:*

$$\frac{1}{n}\log|C| \geq H(\frac{w}{n}) - \frac{w}{n}H(\frac{d}{2w}) - (1 - \frac{w}{n})H(\frac{d}{2(n-w)}) - o(1)$$

Here $H(x)$ denotes the binary entropy of x:

$$H(x) = -x\log_2(x) - (1-x)\log_2(1-x)$$

3.3.2. Formal Description of the Construction. This family of methods construct bipartite graph between two vertex sets $X = Y = [m]^n$, in which m is a integer we will discuss later. In order to formally describe this family of construction, I will define some useful notations at first:

Definition 3.7 (Coloring operation χ). *The coloring operation χ is defined as a mapping: $[m]^n \times \{0,1\}^n \to \{blue, white, red\}$. Such operation can assign colors to a vertex based on the given binary mask.*

Definition 3.8 (Transform operation τ). *The transform operation τ is defined as a mapping: $[m]^n \times \{0,1\}^n \to [m]^n$. Such operation can map a vertex to another one based on the given binary mask.*

Definition 3.9 (Inverse Transform operation τ^{-1}). *The inverse transform operation τ^{-1} is defined as a mapping: $[m]^n \times \{0,1\}^n \to [m]^n$ such that for any $x \in [m]^n$ and $u \in \{0,1\}^n$, it has $\tau(\tau^{-1}(x,u),u) = \tau^{-1}(\tau(x,u),u) = x$*

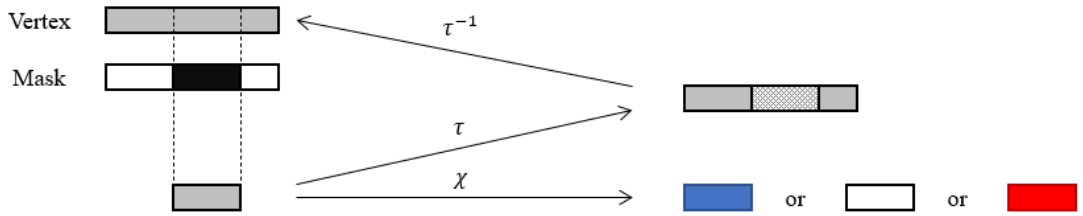

Figure 2. *The operations in the coloring construction*

There are many possible operations that can be used including algebraic calculations, combinations, permutations, and some complex operations. However, in order to keep the induced property in the construction process, some conditions must hold:

Definition 3.10 (Induced Conditions). *The pair of operations (χ,τ) is said to hold the induced conditions on set $U \subseteq \{0,1\}^n$ if:*

- *The inverse transform operation τ^{-1} exists*
- *For any $x \in [m]^n$, $u \in U$ such that $\chi(x,u) = blue$, it has $\chi(\tau(x,u),u) = red$*
- *$x \in [m]^n$, $u \in U$ such that $\chi(x,u) = blue$, then for any $u' \in U$, either $\chi(x,u') \neq blue$ or $\chi(\tau(x,u),u') \neq red$*

Then a specific construction method based on a pair of operations (χ,τ) that hold the induced conditions on set U can be shown as

Theorem 3.11 (Coloring Construction). *If operations (χ,τ) that holds the induced condtions on set $U \subseteq \{0,1\}^n$, for all $u \in U$, define the matching between two vertex sets $X = Y = [m]^n$:*

$$M_u = \{(x,\tau(x,u)) : x \in [m]^n, \chi(x,u) = blue\}$$

Then each matching in the matching set:

$$\mathcal{M} = \{M_u : u \in U\}$$

is induced.

Proof. First prove that M_u is really a matching. From the first induced condition, the inverse transform operation exists. So if there exists $x \neq y$ such that $\tau(x,u) = \tau(y,u)$, then:

$$x = \tau^{-1}(\tau(x,u),u) = \tau^{-1}(\tau(y,u),u) = y$$

contradicts the assumption that $x \neq y$. Meanwhile the transform operation τ can only map x to one vertex $\tau(x,u)$ given u. Thus M_u is indeed a matching.

Then prove that the matchings are induced. Assume edge $(x, \tau(x,u))$ is induced in matching $M_{u'}$. Then it must have $\chi(x,u') = blue$ and $\chi(\tau(x,u),u') = red$, which contradicts the third induced condition. Thus the matchings in $\mathcal{M}$ are induced. ■

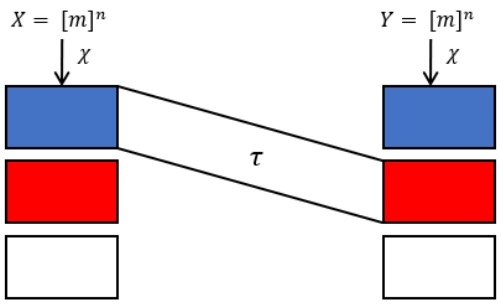

Figure 3. *The Matching constructed based on a mask*

Finally consider the parameter of the RS graph constructed by the method above. Because for each $u \in U$, we can construct a induced matching M_u. The total number of matchings is $t = |U|$. In each matching, the number of matching is equal to the number of blue points:

$$r = \min_{u \in U} |\{x : \chi(x,u) = blue\}|$$
$$t = |U|$$

3.3.3. Basic Configuration. [6] Given integer n as a multiple of 3. Then define $w = p = n/3$, $t = n/7$ and then $d = 2(w-t) = 8n/21$. Choose the operations as below:

$$\chi_0(x,u) = \begin{cases} blue & (\sum_{i:u_i=1} x_i)/p \mod 3 = 0 \\ red & (\sum_{i:u_i=1} x_i)/p \mod 3 = 1 \\ white & (\sum_{i:u_i=1} x_i)/p \mod 3 = 2 \end{cases}$$
$$\tau_0(x,u) = x - 2u$$

Then check the induced conditions on a (n,w,d) fixed-weight Hamming code denoted as U. For the first condition, the inverse transform operation can be easily shown as $\tau^{-1}(x,u) = x + 2u$ and the proof is trivial. For the second condition, given $u \in U$, if $\chi_0(x,u) = blue$, then it has:

$$(\sum_{i:u_i=1} x_i)/p \mod 3 = 0$$

After the transform operation, the sum of the subset:

$$\sum_{i:u_i=1} (\tau(x,u))_i = \sum_{i:u_i=1} (x_i - 2u_i) = \sum_{i:u_i=1} x_i - 2w$$

Because $w = p = n/3$, then:

$$(\sum_{i:u_i=1} \tau(x,u)_i)/p \mod 3 = ((\sum_{i:u_i=1} x_i)/p - 2) \mod 3 = 1$$

thus $\chi_0(\tau(x,u),u) = red$. However, such operation may get a result out of bounder. If there exists $i \in [n]$ such that $u_i = 1$ and $x_i < 2$, then $\tau(x,u)_i < 0 \notin [m]$. In order to fix this, the vertices that may lead to out-of-bound issue are colored white instead of blue or red:

$$\chi(x,u) = \begin{cases} \chi_0(x,u) & \text{for all i s.t. } u_i = 1,\ x_i \notin \{1,2,m-1,m\} \\ white & \text{otherwise} \end{cases}$$

$$\tau(x,u) = x - 2u \mod m$$

$$\tau^{-1}(x,u) = x + 2u \mod m$$

And the condition also holds for the new operations.

For the third condition, given $x \in [m]^n$ and $u \in U$ such that $\chi(x,u) = blue$. Then for any $u' \in U$, if both $\chi(x,u') = blue$ and $\chi(\tau(x,u),u') = red$, consider the difference of the sum of the subset represented by u':

$$\sum_{i:u'_i=1} (x - \tau(x,u)) = 2|\{i : u_i = u'_i = 1\}| < \frac{2n}{7} < \frac{n}{3}$$

However, because $\chi(x,u') = blue$ and $\chi(\tau(x,u),u') = red$, meanwhile $\sum_{i:u'_i=1} \tau(x,u)_i \leq \sum_{i:u'_i=1} x_i$, there must be a gap of length $n/3$ (the white vertices) between them and then:

$$\sum_{i:u'_i=1} (x - \tau(x,u)) \geq \frac{n}{3}$$

Now the two equations hold a contradiction, and thus the third condition holds. Then by following the template in the previous section, we can prove that the matchings are induced.

Finally let's confirm the parameters of such RS graph. Let $N = m^n$ be the size of vertex set. For each $u \in U$, by symmetric property of the coloring operation $\chi_0(x,u)$, the number of *blue* vertex is $\frac{N}{3}$. Then some of them will become white because of containing $\{0,1,m-2,m-1\}$, then the matching size, or the number of blue points corresponding to $\chi(x,n)$ is:

$$|M_u| \geq \frac{N}{3} - \frac{n}{3} \cdot 4 \cdot m^{n-1} = \frac{N}{3}(1 - \frac{4n}{m})$$

So set $m = n^2$ and then $|M_u| = \frac{N}{3} - o(N)$.

By Theorem 3.6, plug in the parameter $w = n/3$, $t = n/7$, the size of code is about:

$$\log|U| \geq \left(H(\frac{1}{3}) - \frac{1}{3}H(\frac{4}{7}) - \frac{2}{3}H(\frac{2}{7}) - o(1)\right) n = \Omega(n)$$

Because $N = n^{2n}$, we have $|U| = 2^{\Omega(n)} = N^{\Omega(1/\log\log N)}$ matchings in total. As a result, the output of the construction method is a $(N/3 - o(N), N^{\Omega(1/\log\log N)})$-RS graph.

3.3.4. Improved Configuration. [8] Based on the configuration above, we can find that the number of matchings t is related to the Theorem 3.6 and hard to optimize. Thus the main idea of improving the method is to increase the matching size, i.e. the number of blue vertices for a given $u \in U$. Besides, the white vertices are never used by neither the construction nor the conditions. So modifying the χ operation and take more previous white vertices into consideration is one possible solution.

Let $w = (\epsilon/6)m$, $p = 2(1 + 2/\epsilon)w$, $t = (5/4)(\epsilon/6)^2 m$ and then:

$$d = 2(w - t) = \left(\frac{\epsilon}{3} - \frac{5}{2}\left(\frac{\epsilon}{6}\right)^2\right) m$$

Then define the operations:

$$\chi_0(x, u) = \begin{cases} red & (\sum_{i:u_i=1} x_i)/p \in [0, (2/\epsilon)w) \\ blue & (\sum_{i:u_i=1} x_i)/p \in [(1 + 2/\epsilon)w, (1 + 4/\epsilon)w) \\ white & \text{otherwise} \end{cases}$$

$$\tau_0(x, u) = x - \left(1 + \frac{2}{\epsilon}\right) u$$

In order to solve the out-of-bound problem, adjust the operations:

$$\chi(x, u) = \begin{cases} \chi_0(x, u) & x_i \geq (1 + 2/\epsilon) \text{ for all } i \text{ s.t. } u_i = 1 \\ white & \text{otherwise} \end{cases}$$

$$\tau(x, u) = x - \left(1 + \frac{2}{\epsilon}\right) u \mod m$$

$$\tau^{-1}(x, u) = x + \left(1 + \frac{2}{\epsilon}\right) u \mod m$$

Similarly, check the induced condition of such operations like what we have done in previous section. And finally, we can construct a $(N/3 - o(N), N^{\Omega(1/\log\log N)})$-RS graph.

3.3.5. Implement of the Construction. Based on the construction in Theorem 3.11, we can directly design an algorithm to generate RS graph, as shown in Algorithm 3.2.

This algorithm is very awful. Because it uses two levels of for loops and iterates all the m^n vertices for $|U|$ times. Then the running time of the algorithm will increase terribly as n gets bigger. For example, assume now we are using the basic configuration such that $m = n^2$ and $|U| = 2^{\Omega(n)}$, the total number of iterations is about $n^{2n}2^{\Omega(n)}$.In each iteration, one needs to compute the sum of w values, which can't be computed immediately. In order to reduce the time cost of the algorithm, and find a new way to deal with the for loops is necessary.

The main idea of my algorithm is to iterate all blue vertices and directly add the related edge, instead of iterating all the vertices and then checking their color. Now the question is how to find all the blue vertices.

Luckily the operation χ is irrelevant to the positions of the value that is masked by u. That means, given fixed w numbers, for any $u \in U$, write the selected numbers at the position masked by u and choose arbitrary values at other positions. As a result, the color of the vector (or vertex) is fixed, no matter what the mask u is. Thus we can find all the blue points in three steps:

Algorithm 3.2 Coloring Construction

Input parameters n, w, d, m
Input a (n, w, d) fixed-weight Hamming code U
Input the operations (χ, τ) that hold the induced conditions on U
Define the vertex set $A := [m]^n$, $B := [m]^n$
Define the collection of matchings $\mathcal{M} := \{\}$
for all $u \in U$ **do**
  Define a matching $M_u := \{\}$
  for all $a \in A$ **do**
    if $\chi(a, u) = blue$ **then**
      Update $M_u := M_u \cup \{(a, \tau(a, u))\}$
    end if
  end for
  Update $\mathcal{M} := \mathcal{M} \cup \{M_u\}$
end for
Compute the edge set $E := \cup_{x \in [n]} M_u$
return $G := (A, B, E)$ and $\mathcal{M}$

1. Find all possible subsequences of size w that can make the vertex be colored by blue
2. Given a mask $u \in U$, filling in the subsequence at the positions masked by u
3. Assign all possible values to other positions

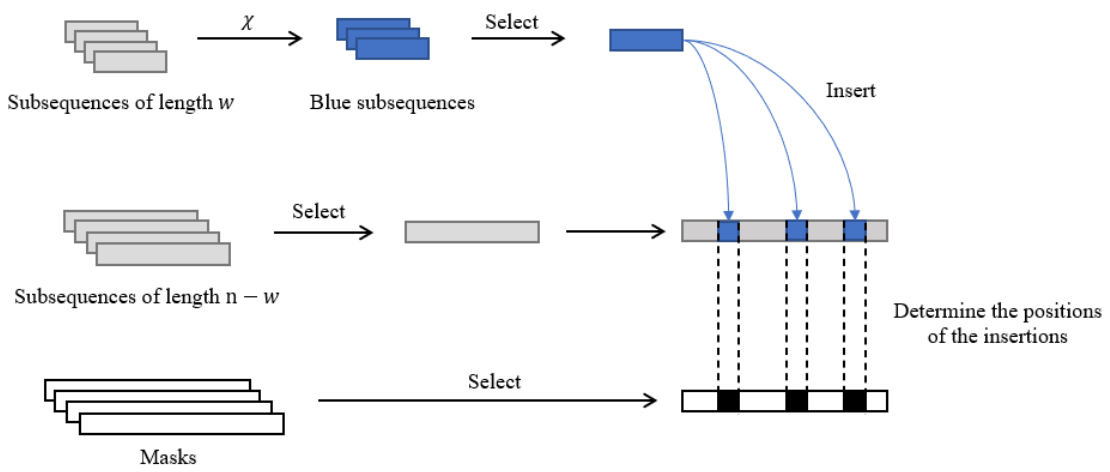

Figure 4. *The construction based on insertion*

Another possible optimization is to use some exchange operations. First, put the subsequence at the first w positions, and then exchange the entries of the vectors based on the mask u. Based on these two optimizations, a better algorithm can be built as Algorithm 3.3,

3.4, 3.5:

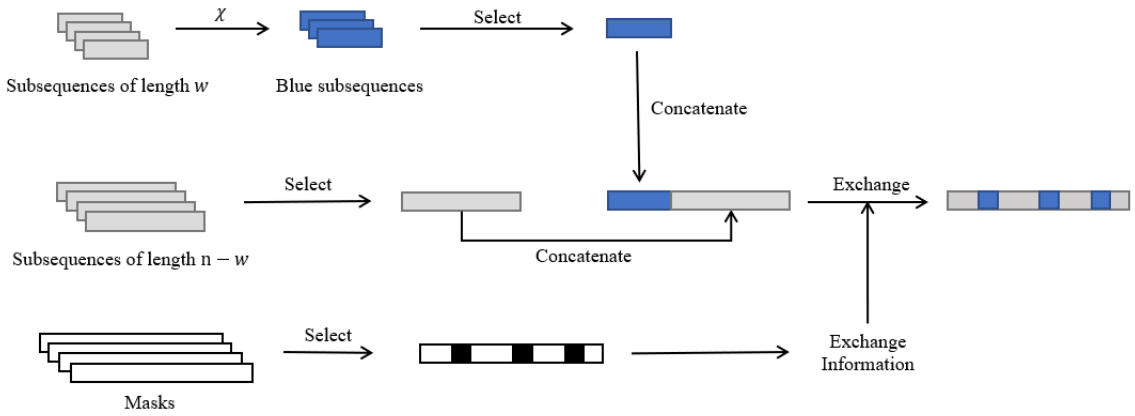

Figure 5. *The construction based on exchange*

Algorithm 3.3 Blue Subsequence Generation

Input parameters w, m
Input the operations χ
Define the collection of blue subsequence $\mathcal{B} := \{\}$
for all $a \in [m]^w$ **do**
  if $\chi(a, \mathbf{1}^w) = blue$ **then**
    Update $\mathcal{B} := \mathcal{B} \cup \{a\}$
  end if
end for
return $\mathcal{B}$

Actually, the number of iterations will not change a lot in this improved algorithm because we need to iterate all possible vertices. However, in this algorithm, a subsequence will only be computed once to confirm the color while generating the blue subsequence. What's more, it also follows the idea of pre-computation. By some pre-computed set such as the collection of blue subsequences and base sequences, the final construction will be faster because it only exchanges two entries of the vector for w times without any other computation.

However, the disadvantage of the algorithms based on the coloring constructions above is clear: both generating the fixed-weight Hamming code and the iterations are time-consuming.

3.4. Construction on Nearly Complete Graph.

3.4.1. Geometric based construction. The geometric construction in [1] is based on a theorem about the cover of a graph by induced matchings:

Theorem 3.12 (Cover by Induced Matchings(Lemma 2.5 in [1])). *Let H be a graph with maximum degree d. Then H can be covered by $O(d^2)$ induced matchings.*

Algorithm 3.4 Base Sequence Generation

Input parameters n, w, m, d
Input the blue subsequence $\mathcal{B}$
Define the base sequence $\mathcal{S} := \{\}$
for all $b \in \mathcal{B}$ **do**
  for all $a \in [m]^{n-w}$ **do**
    Update $\mathcal{S} := \mathcal{S} \cup \{\text{concat}(b, a)\}$
  end for
end for
return $\mathcal{S}$

Algorithm 3.5 Improved Coloring Construction

Input parameters n, w, d, m
Input a (n, w, d) fixed-weight Hamming code U
Input the operations (χ, τ) that hold the induced conditions on U
Input the base sequence $\mathcal{S}$
Define the vertex set $A := [m]^n$, $B := [m]^n$
Define the collection of matchings $\mathcal{M} := \{\}$
for all $u \in U$ **do**
  Define a matching $M_u := \{\}$
  Define $v := \{i : u_i = 1\}$
  for all $s \in S$ **do**
    Define $t := s$
    for all $i \in [w]$ **do**
      $t := \text{exchange}(t, i, v_i)$
    end for
    Update $M_u := M_u \cup \{(t, \tau(t, u))\}$
  end for
  Update $\mathcal{M} := \mathcal{M} \cup \{M_u\}$
end for
Compute the edge set $E := \cup_{x \in [n]} M_u$
return $G := (A, B, E)$ and $\mathcal{M}$

Thus if one can construct a graph with the limited degree, then from the fact above, it can be shown as a RS graph with $t = O(d^2)$. Meanwhile, if the maximum degree of a graph is so big, then the number of matchings will increase rapidly and thus the construction may become trivial. So it's necessary to reduce the maximum degree of the graph. However, the total number of edges will decrease rapidly if we make the maximum degree too small. So the problem is to find a balance point between fewer matchings and fewer edges. The paper uses a two-step construction. First cover a graph with many edges with a series of subgraphs with low maximum degree. Then cover the subgraphs with not too many induced matchings. Finally collect all these matchings and remove the repeated edges.

At the first step, construct the nearly complete graph on the vertex set $[m]^n$ as:

Definition 3.13 (Nearly complete graph G_{NC}^{norm}). *For each pair of vertices $x, y \in [m]^n$, they are adjacent if and only if*

$$\left| \|x - y\|_2^2 - \mu \right| \leq n$$

where μ denotes the expectation of the norm between vertices x, y sampled from $[m]^n$ uniformly at random:

$$\mu = \mathbb{E}_{x,y}(\|x - y\|_2^2)$$

By Hoeffding's Inequality, the probability that any two random vertices are not adjacent is:

$$\Pr\left(\left| \|x - y\|_2^2 - \mu \right| \leq n\right) \leq 2e^{-n/2m^4}$$

And thus the total edge number of the graph:

$$|E| \geq \binom{N}{2}(1 - o(1))$$

Then define the subgraph with the limited maximum degree:

Definition 3.14 (Subgraph G_z). *Given $z \in [m]^n$, define the vertex set V_z:*

$$V_z = \left\{ x : \left| \|x - z\|_2^2 - \frac{\mu}{4} \right| \leq \frac{3n}{4} \right\}$$

And the subgraph G_z is the induced graph of G_{NC}^{norm} on vertex set V_z.

By the property of the norm $\|x - y\|_2^2$, it can be proven that:

Theorem 3.15 (G_{NC}^{norm} can be covered by G_z). *If $n \geq 2m$, then for each edge (x, y) in G_{NC}^{norm}, there exists $z \in [m]^n$ such that $x, y \in V_z$.*

Theorem 3.16 (G_z has limited degree). *The maximum degree of each G_z is less than $(10.5)^n$*

Thus the graph G_{NC}^{norm} can be covered by $m^n(10.5)^n$ induced matchings. Let $N = m^n$ and tune the parameters, we can get the RS graph with $N^{1-o(1)}$ matchings and $\binom{N}{2}(1 - o(1))$ edges in total.

However, this construction method is so hard to implement because the Theorem 3.12 only proves the existence of the induced matchings, but does not give us any idea to find out these matchings. As a result, we can generate the graph G_{NC}^{norm} efficiently by its definition, but we can't know how to decomposite them into induced matchings.

3.4.2. Hamming code based construction. Similarly, there is another construction that follows the same process: removing some edges from the complete graph, and then finding some induced matchings to cover the nearly complete graph. In the previous section, we remove the edges from the complete graph based on the 2-norm between the vertices. Then Hamming distance is another way to describe the distance of two vectors, so we can also try to construct a nearly complete graph:

Definition 3.17 (Nearly complete graph $G_{NC}^{Hamming}$). *For each pair of vertices $x, y \in [m]^n$, they are adjacent if and only if*

$$d_H(x, y) > n - d$$

where $d < n$ is a fixed integer.

Then by Gilbert-Varshamov bound, one can show that if $d/n \geq 2/(m-1)$, the number of missing edges is at most N^e where:

$$e = 1 + \frac{H(d/n) + (1 - d/n)\log_2(m-1)}{\log_2 m} + o(1)$$

The next step is to cover the nearly complete graph with some induced matchings. Here the construction uses the Hamming code to build the matchings:

Definition 3.18 (Hamming Code). *Given vector length n, subspace length k and distance d, then $[n, k, d]$ Hamming code is defined as a collection of vectors $\mathcal{C} \subseteq \{0,1\}^n$ such that:*

- $|\mathcal{C}| = 2^k$
- *For any different $u, v \in \mathcal{C}$, $d_H(u, v) \geq d$*

Definition 3.19 (Flip Operation). *Given a pair of vertices (a, b) in graph $G_{NC}^{Hamming}$, denote:*

$$S = \{i : a_i = b_i\}$$

Let $r = |S|$ and the rest indices:

$$[n] \setminus S = \{i_1, \cdots, i_{n-r}\}, i_1 < \cdots < i_{n-r}$$

For vector $x \in \{0,1\}^{n-r}$, define the x-flip of (a, b) as (c, d) by:

- *For $i \in S$, $c_i = d_i = a_i = b_i$*
- *For $i = i_j \notin S$ and $x_j = 0$, $c_i = a_i, d_i = b_i$*
- *For $i = i_j \notin S$ and $x_j = 1$, $c_i = b_i, d_i = a_i$*

Based on the flip operation, define the relationship:

Definition 3.20 ($\sim$ Relation). *For two pairs of vertices (a, b) and (c, d), we say $(a, b) \sim (c, d)$ if and only if $S = \{i : a_i = b_i\} = \{i : c_i = d_i\}$ and $|S| < d$. Furthurmore, there exists a codeword $x \in \mathcal{C}$ such that (c, d) is the x-flip of (a, b).*

Finally consider the property of such relation of vertex pairs. Especially the relationship between $\sim$ relation and Hamming code. Here is the core theorem in the construction:

Theorem 3.21. *Given a Hamming code $\mathcal{C}_{n-r}$ containing all-one vector, then the relation $\sim$ is an equivalence relation over unordered vertex pair (a, b) which has $d_H(a, b) > n - d$*

Thus if we represent the unordered vertex pair (a, b) as an edge, and collect all the equivalence edges as a matching, by the property of Hamming code, each equivalence class is an induced matching consisting of 2^{k-1} edges. In other words, such graph $G_{NC}^{Hamming}$ can be partitioned into several induced matchings with matching size 2^{k-1}.

3.5. Protocol Based Construction.

3.5.1. Number-On-The-Forehead Protocol. This construction method is based on the Number-On-The-Forehead (NOF) protocol:

Definition 3.22 (Number-On-The-Forehead Protocol). *Given a function:*

$$f : X_1 \times X_2 \times \cdots \times X_k \to \{0, 1\}$$

Assume there are k players and each player has a corresponding variable $x_i \in X_i$. The i-th player can see all the variables except x_i. The Number-On-The-Forehead Protocol is a protocol for the k players to communicate on a blackboard and compute the value of $f(x_1, \cdots, x_k)$. Meanwhile, based on the protocol the binary string written on the common blackboard by the players is called the Transcript of the input x, denoted as $T(x)$. Because all the players can write on the blackboard, we use $T_i(x)$ to show the binary string written by the i-th player. Then given a fixed transcript T, a cylinder intersection $S(T)$ is defined as the subset of entries such that $T(x) = T$ and $f(x) = 1$.

In the construction, we focus more on the function with the sub-permutation property:

Definition 3.23 (Sub-Permutation). *Given a function:*

$$f : [n]^{k-1} \times [N] \to \{0, 1\}$$

We say f is a sub-permutation when:

- *every line in the k-th dimension contains a single 1*
- *every other line contains at most one 1*

And f is a weak sub-permutation when every line contains at most one 1

Now we need a "bridge" to connect the protocol and the graph, here is a theorem to do this job:

Theorem 3.24. *Let $f : [n] \times [n] \times [N] \to \{0, 1\}$ be a weak sub-permutation, and S be a symmetric cylinder intersection. Let $H = ([n], F)$ be the subgraph of G_S induced on V_A such that:*

$$F = \{(x, y) : \exists z \in V_B \text{ s.t. } (x, y, z) \in S\}$$

Then the edges of F can be partitioned into N induced matchings.

Thus if we can construct a protocol that holds the conditions above, we can also construct a RS graph.

3.5.2. Construction. Given a function $f : [n]^{k-1} \times [N] \to \{0, 1\}$, a NOF protocol for f, a transcript T of the last player, denote:

$$S_k(T) = \{(x_1, \cdots, x_k) \in [n]^{k-1} \times [N] : T_k(x_1, \cdots, x_k) = T, f(x_1, \cdots, x_k) = 1\}$$

Then the process to do the construction can be summarized as:

1. Choose a weak sub-permutation $f : [n]^{k-1} \times [N] \to \{0, 1\}$

2. Construct a NOF protocol P for computing f
3. Find a transcript T of the last player such that $S_k(T)$ is symmetric
4. Apply $S = S_k(T)$ and use the Theorem 3.24

The paper gives an example of constructing RS graph with $\Omega(n^2/\log n)$ edges and $n^{1+O(1/\log\log n)}$ induced matchings. Let $q, d > 1$ be natural numbers and $n = q^d$. Define:

$$Z_{q,d} = \left\{ \frac{1}{2}(x+y) : x, y \in [q]^d \right\}$$

and the function:

$$g_{q,d} : ([q]^d)^2 \times Z_{q,d} \to \{0,1\}$$

such that $g_{q,d}(x,y,z) = 1$ if and only if $x + y = 2z$. It can be proven that $g_{q,d}$ is a weak sub-permutation. Then construct the NOF protocol as Algorithm 3.6.

Algorithm 3.6 NOF Protocol

The z-player computes $\|x-y\|_2^2$ and write it on the board
The y-player writes 1 if and only if $\|x-y\|_2^2 = 4\|x-z\|_2^2$
The x-player writes 1 if and only if $\|x-y\|_2^2 = 4\|y-z\|_2^2$
if Both x-player and y-player write 1 **then**
  return 1
else
  return 0
end if

By Chernoff-Hoeffding's Inequality,

$$\Pr\left(\left|\|x-y\|_2^2 - \mathbb{E}(\|x-y\|_2^2)\right| \geq t\right) \leq 2e^{-\frac{2t^2}{dq^4}}$$

So with constant probability, let $d = q^4$ one can get there exists T such that $|S_3(T)| \geq \Omega(n^2/\log n)$ and $S_3(T)$ is symmetric. And finally we can construct the RS graph by using the Theorem 3.24.

4. Easy-to-Implement Construction.

4.1. Basic Idea. My new construction is based on the coloring construction method. In section 3.3.5 I have discussed the implementation of the method and there are three main reasons that make it hard to implement:

- Generating fixed-weight Hamming code is hard
- The size of the graph increases rapidly and the running time becomes unaffordable
- The computation of operations (χ, τ) in each iteration will influence the total cost of time

So my construction will focus on these points to reduce the cost of the implement.

- Design a new collection of subset U that is not hard to generate with good property
- Construct RS graph on smaller vertex set
- Reduce the cost of operations (χ, τ)

4.2. Configuration. My construction starts from the XOR operation $\oplus$. Because XOR operation is one of the most efficient operations for the computer, and I want to introduce XOR operations to the coloring construction method.

A suitable position for the XOR operation is the τ operation. Because it's easy to find that given a mask u and a binary vector x, flipping the masked bits only needs to do the XOR operations, and its inverse operation is itself:

$$\tau(x,u) = x \oplus u$$
$$\tau^{-1}(x,u) = x \oplus u$$

Then is the χ operation. When we flip the masked bits, some property of the subsequence will change, such as the number of 0's and 1's of the masked bits. Here I choose to use the parity of the number of 1's of the masked bits:

$$\chi(x,u) = \begin{cases} blue & \oplus_{i:u_i=1} x_i = 1 \\ red & \oplus_{i:u_i=1} x_i = 0 \end{cases}$$

The next step is to check the induced conditions.

Theorem 4.1 (Induced Condition for XOR Based Operation). *Assume U is a collection of masks of length n such that:*

- *For any $u \in U$, $w(u) \mod 2 = 1$, i.e. the size of subsequence is odd*
- *For any $u, u' \in U$, $u \neq u'$, then $w(u \cap u') \mod 2 = 0$, i.e. the intersection size of two different subsequences is even*

Then the XOR based operation holds the induced conditions.

Proof. For the first condition, we have shown that the inverse operation $\tau^{-1}(x,u) = \tau(x,u) = x \oplus u$ exists.

For the second condition, if the vertex $x \in \{0,1\}^n$ is colored blue:

$$\bigoplus_{i:u_i=1} x_i = 1$$

Then after the flip, we have:

$$\bigoplus_{i:u_i=1} \tau(x,u)_i = \bigoplus_{i:u_i=0} x_i = \left(\bigoplus_{i\in[n]} x_i\right) \oplus \left(\bigoplus_{i:u_i=1} x_i\right) = 1 \oplus 1 = 0$$

Thus the vertex $\tau(x,u)$ is colored red. The second condition holds.

For the third condition, assume there exists different $u, u' \in U$ such that:

$$\chi(x,u) = blue$$
$$\chi(x,u') = blue$$
$$\chi(\tau(x,u),u') = red$$

Define the set of indices:

$$
\begin{aligned}
I_1 &= \{i : u_i = 1, u'_i = 0\} \\
I_2 &= \{i : u_i = 0, u'_i = 1\} \\
I_c &= \{i : u_i = 1, u'_i = 1\}
\end{aligned}
$$

Then represent the χ operation by these sets:

$$
\begin{aligned}
\left(\bigoplus_{i\in I_1} x_i\right) \oplus \left(\bigoplus_{i\in I_c} x_i\right) &= 1 \\
\left(\bigoplus_{i\in I_2} x_i\right) \oplus \left(\bigoplus_{i\in I_c} x_i\right) &= 1 \\
\left(\bigoplus_{i\in I_2} x_i\right) \oplus \left(\bigoplus_{i\in I_c} (x_i \oplus 1)\right) &= 0
\end{aligned}
$$

Combine the second and third equations, we have:

$$
\left(\bigoplus_{i\in I_c} (x_i \oplus 1)\right) \oplus \left(\bigoplus_{i\in I_c} x_i\right) = \bigoplus_{i\in I_c} 1 = |I_c| \mod 2 = 1 = 1 \oplus 0
$$

Thus $|I_c| \mod 2 = 1$, which contradicts the assumption such that the intersection size of two subsequences is even. Thus the third condition holds. The operation (χ, τ) holds the induced conditions. ∎

Finally, I find a construction of the set U such that holds the conditions mentioned in Theorem 4.1. Consider a vector of length 2^k, keep splitting the indices of the vector into two parts with the same size, until the length of each part is 2. In this way, we can get $1 + 2 + \cdots + 2^{k-1} = 2^k - 1$ subsets in total, denoted as $I_1, \cdots, I_{2^k-1}$. Then it's obvious that $|I_j|$ is a power of 2 and thus an even number for all $j \in [2^k - 1]$. What's more, for any two different subsets I_j and I_k, either $I_j \subset I_k$, or $I_k \subset I_j$, or $I_j \cap I_k = 0$. As a result, $|I_j \cap I_j|$ is always even and the second condition holds.

Now the rest of the work is to ensure the first condition without influencing the intersection of the subsets. Consider another $2^k - 1$ bits and assign each bit to each I_j above, then the size of I_j becomes odd but the intersection size of different subsets keeps even.

Here is an example. Let $k = 3$ and the mask length $n = 2^{k+1} - 1 = 15$:

Table 2
The Construction of the Masks

index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I_1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
I_2	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0
I_3	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0
I_4	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0
I_5	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0
I_6	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0
I_7	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1

Finally, let's discuss the parameter of construced RS graph. The vertex set is $\{0,1\}^n$ and thus the number of vertice is $N = 2^n = 2^{2^{k+1}-1}$. The size of U is about $|U| = 2^k - 1 = \Theta(\log N)$. And by symmetry for any given $u \in U$, the number of blue vertices and red vertices are the same. So the matching size is $N/2$. Thus by this method, we construct a $(N/2, \Theta(\log N))$ RS graph.

4.3. Implement. Based on the Algorithm 3.2, we can also implement such construction. What's more, such construction is far more efficient than the previous configurations. For the construction of U, For the number of vertices, the new construction only has 2^n vertices per set, which is significantly less than n^{2n} in previous constructions. For the complexity of the operations, XOR operations are one of the fastest operations for the computer and thus the χ and τ operations can be implemented in a fast way.

5. Conclusion. In this paper, we talked about different construction methods for RS graphs, and analyze the implementation of such methods. The core problem of such construction methods is that nearly all the constructions are based on some combinatorial problems such as 3AP-free set, fixed-weight Hamming code and so on. Such problems are very hard to implement to reach the theoretical bound. What's more, in order to construct the matchings based on the combinatorial structures, we may also need to do some complex operations which also leads to the complexity of the implementation. However, in the situation that we just want some real RS graph containing enough edge, instead of the extreme RS graphs that hold the bound tightly, such constructions may not so useful or efficient. Thus my work is focused on this condition and shows a new idea about the construction methods. Although the edge density of the generated RS graph decreases, the process of generation becomes very easy. By some easy-to-implement methods like XOR operations and deterministic generation algorithms, I finally find a construction method that can generate $(N/2, \Theta(\log N))$-RS graph. Compare to the previous coloring construction methods, it achieves the best matching size, but the bound of the number of matchings decreases. There is still some space to improve the number of matchings, but it may increase the cost of generating the masks and conflict with my goal of reducing the complexity of implementation.

Because of the lack of time, I can't implement all these methods and compare their performance in a more precise way. And the construction may only talk about the asymptotic bound, and the constant term is seldom considered. Thus there is still a long way to go before

we can build a good RS graph generator.

REFERENCES

[1] N. Alon, A. Moitra, and B. Sudakov, *Nearly complete graphs decomposable into large induced matchings and their applications*, Journal of the European Mathematical Society, 15 (2013), pp. 1575–1596.

[2] N. Alon and A. Shraibman, *Number on the forehead protocols yielding dense ruzsa–szemerédi graphs and hypergraphs*, Acta Mathematica Hungarica, 161 (2020), pp. 488–506.

[3] S. Assadi, *A two-pass (conditional) lower bound for semi-streaming maximum matching*, in Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2022, pp. 708–742.

[4] Y. Birk, N. Linial, and R. Meshulam, *On the uniform-traffic capacity of single-hop interconnections employing shared directional multichannels*, IEEE Transactions on Information Theory, 39 (1993), pp. 186–191.

[5] J. Dybizbański, *Sequences containing no 3-term arithmetic progressions*, the electronic journal of combinatorics, (2012), pp. P15–P15.

[6] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, and A. Samorodnitsky, *Monotonicity testing over general poset domains*, in Proceedings of the thiry-fourth annual ACM symposium on Theory of computing, 2002, pp. 474–483.

[7] J. Fox, H. Huang, and B. Sudakov, *On graphs decomposable into induced matchings of linear sizes*, Bulletin of the London Mathematical Society, 49 (2017), pp. 45–57.

[8] A. Goel, M. Kapralov, and S. Khanna, *On the communication and streaming complexity of maximum bipartite matching*, in Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms, SIAM, 2012, pp. 468–485.

[9] J. Håstad and A. Wigderson, *Simple analysis of graph tests for linearity and pcp*, Random Structures & Algorithms, 22 (2003), pp. 139–160.

[10] M. Kapralov, *Better bounds for matchings in the streaming model*, in Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms, SIAM, 2013, pp. 1679–1697.

[11] J. Komlós and M. Simonovits, *Szemerédi's regularity lemma and its applications in graph theory*, (1996).

[12] V. Levenštein, *Upper bounds for codes with a fixed weight of vectors*, Problemy Peredaci Informacii, 7 (1971), pp. 3–12.

[13] I. Z. Ruzsa and E. Szemerédi, *Triple systems with no six points carrying three triangles*, Combinatorics (Keszthely, 1976), Coll. Math. Soc. J. Bolyai, 18 (1978), pp. 939–945.