DEPARTMENT OF COMPUTER SCIENCE

Streaming Algorithms for Maximum Independent Sets

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Engineering in the Faculty of Engineering.

Thursday 5th May, 2022

Abstract

We study streaming algorithms for finding maximum independent sets (MIS); that is, finding a maximum cardinality subset of disjoint objects from a given collection. We focus specifically on the case of two dimensions where every object is a square, and aim to give an overview of the state of the problem and the open questions that remain. In the case where the squares are of uniform size, there is a 3-approximation streaming algorithm using space linear in the size of the output; we build on this by developing some preliminary results for squares of multiple sizes. The mentioned algorithm also allows us to achieve an upper bound of 3 for approximating the size of the MIS. However, through a reduction from a multi-party communication problem, we demonstrate that there exists a lower bound of $\frac{5}{2}$, and we discuss this gap. Finally, for squares of two different sizes, this lower bound increases to 3, and we cover the proof and implications of this result [2].

Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Taught Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, this work is my own work. Work done in collaboration with, or with the assistance of others, is indicated as such. I have identified all material in this dissertation which is not my own work through appropriate referencing and acknowledgement. Where I have quoted or otherwise incorporated material which is the work of others, I have included the source in the references. Any views expressed in the dissertation, other than referenced material, are those of the author.

[redacted] Thursday 5th May, 2022

Contents

List of Figures

List of Tables

Ethics Statement

This project did not require ethical review, as determined by my supervisor, Dr. Christian Konrad.

Chapter 1

Introduction and Background

In this project we work with the streaming model of algorithms. In the streaming setting, the algorithm receives the input data as a stream of objects, rather than all at once (as it would in the classical setting). A streaming algorithm can also be thought of as an algorithm which makes only one pass over the input data, as opposed to having free access to all of it. In this way, the space used by a streaming algorithm is an important consideration. Very large data sets are becoming more and more common, and the motivation for streaming algorithms stems from a need to work with and manage these data sets. For example, an algorithm could be required to compute some function of a data set, and output a reasonable solution (i.e. with a good approximation ratio) despite memory limitations preventing storage of the whole input. It should be obvious that if a streaming algorithm requires space linear in the size of the input to achieve a given approximation ratio, then we effectively have a lower bound for the problem, and this method is used throughout the literature.

The streaming model of algorithms is related to communication complexity [4]. In particular, many problems studied in the streaming setting have lower bounds proved via reductions to various communication problems, and we will explore two applications of this method ourselves later in the project.

Within this streaming setting the particular problem that we will study is approximating maximum independent sets (MIS) and approximating their sizes. We define the problem as follows: given a collection of objects (e.g., intervals on the real line, shapes in two dimensions), find a maximum cardinality disjoint subset[1]. This primary applications of this problem are in proving computational complexity of many theoretical problems [8], and as models for real world optimisation problems [5].

The overall aim of this project is to give an overview of the results which already exist for a particular case of this problem; in particular, the case of a unit square stream. Along the way we will touch on problems of intervals and unit intervals and their relation to the two-dimensional case. Finally, we will develop some preliminary results for the problem of squares of arbitrary sizes, building on methods used for the case of unit squares. It is our hope that this project provides context and clarity for the state of this problem, as well as highlighting possible open questions for any interested party.

We will now take a look at some of the existing results, presented in the below table:

Space Bound	**Approx. MIS** $\tilde{O}(\alpha(G))$	**Approx. size of MIS** $\text{poly}(\log n, \epsilon^{-1})$	**Approx. size of MIS** $\Omega(n)$
Unit Interval	$3/2$ [3]	$3/2 + \epsilon$ [1]	$< 3/2$ [3]
Interval	2 [3]	$2 + \epsilon$ [1]	< 2 [3]
Unit Square	3 [2]	$3 + \epsilon$ [2]	$< 5/2$ [2]
Squares Size $x \in [w, v]$	$\frac{6w^2}{(2w-v)(3w-v)}$	$\frac{6w^2}{(2w-v)(3w-v)} + \epsilon$	< 3 [2]

Table 1.1: The first column relates to the approximation factor when estimating the MIS. The second and third columns relate to the upper and lower bounds respectively for the approximation factor when estimating the size of the MIS. All uncited results are from this paper.

In the one dimensional case, the problem is known as "interval selection" and consists of finding

[1] It is possible to represent these collections of objects as either graphs (streams of vertices and edges), or with a geometric representation. In this project, we will work with the latter.

disjoint subsets of intervals on the real line. The results for interval selection come from the paper "Space-Constrained Interval Selection" [3], and "Interval Selection in the Streaming Model" [1]. The first develops a deterministic 2-approximation streaming algorithm for the problem, along with an algorithm for the special case of proper intervals, which gives an improved approximation ratio of $\frac{3}{2}$. This is followed by proofs that these bounds are essentially the best possible when working in the streaming setting, as any better approximation factor would require space linear in the size of the input, which, of course, defeats the purpose of a streaming algorithm. The second contains results for approximating the size of the MIS in interval selection.

The third and fourth rows contain results for squares in two dimensions. The results for unit squares come from the paper "Independent Sets in Vertex-Arrival Streams" [2], and include a 3-approximation algorithm for finding an MIS on a unit square stream, as well as upper and lower bounds for estimating the size of the MIS. It can be seen that, unlike for intervals and unit intervals, for unit squares there is a noticeable gap between the upper and lower bounds, which will be explored later in this paper.

Finally, the last row holds results for squares of multiple sizes, which were developed as part of this project. Note that these results hold only when the squares all have side lengths between v and w, where $w \leq v \leq \frac{3}{2}w$; in other words the squares must all be within a certain size of one another for the result to hold, and this restriction is a consequence of the construction used in the proof. We will discuss this in detail in the next chapter, but the essential meaning is this: when allowed squares of arbitrary size, the approximation factor that we can achieve using the same method for unit squares depends only on the size of the largest square.

Chapter 2

The Upper Bound

2.1 The Algorithm

We will begin by describing in detail an algorithm for computing MIS on a unit square stream. This algorithm is taken from [2] and produces a 3-approximation for the MIS using space linear in the size of the output. The algorithm is a generalisation of a similar algorithm for unit interval streams [1] which decomposes the real line into length 3 segments. This algorithm instead decomposes the 2-dimensional plane into 2-by-3 strips.

Theorem 1. [2] *For a unit square stream, we can 3-approximate MIS.*

Proof. We consider the half-open strip [0, 3) x [0, 2). At most 2 non-overlapping closed unit squares can fit fully inside the region. In order for these squares to not overlap, one must be to the left of the other (they cannot be one on top of the other due to the vertical dimensions of the strip) (Figure 2.1). Then, to determine an MIS on this strip, we need store only 2 squares; the leftmost and rightmost.

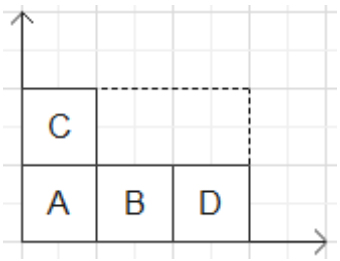


Figure 2.1: Squares A and B are within the half-open strip, squares C and D are not.

Now, we consider the entire 2-dimensional plain, partitioned into the same 3-by-2 half-open strips. Further, we consider only squares which fall entirely within one of these strips (Figure 2.2). We can solve MIS on each strip as before and take the union, which determines exactly MIS on the set of squares which fall entirely within a strip.

Finally, we consider that we can shift the partition uniformly and randomly by up to 2 units in the vertical axis and up to 3 units in the horizontal axis. We ask ourselves the probability that any given square falls fully within a half-open strip once we have shifted the partition. In the vertical axis, this probability is $\frac{1}{2}$ and in the horizontal axis this probability is $\frac{2}{3}$, and multiplying gives us an overall probability of $\frac{1}{3}$. Therefore, if we consider placing the partition randomly on the 2-dimensional plane, any square will be fully contained in a strip at least 1/3 of the time (Figure 2.3). Hence, if we then compute MIS on these various possible partitions, the largest of these outputs must be a 3-approximation.

□

So we have a 3-approximation streaming algorithm for computing MIS on a unit square stream. A natural question concerns the space used by the algorithm. This is connected to the fact that, in reality, we cannot partition the space in an infinite number of ways. In fact, the space used will depend on the number of different ways that we partition the 2-dimensional plane. As an example, consider the case

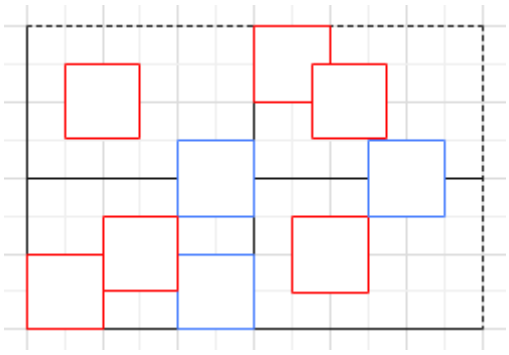


Figure 2.2: In this example, the red squares are contained fully within a strip and the blue squares are not. Therefore, this step computes MIS on the set of red squares.

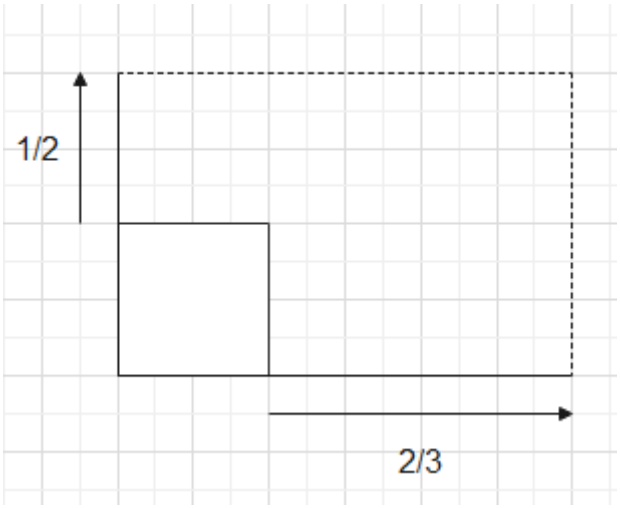


Figure 2.3: The probabilities mentioned.

where we shift the partition by either 0, 1 or 2 units horizontally, and by either 0 or 1 units vertically. This gives us 6 different ways of partitioning the space. In this case we would be computing MIS on 6 different substreams of squares. As mentioned above, computing MIS on a strip requires us to store 2 squares; the leftmost and rightmost within the strip. Therefore, the algorithm stores at most 12 squares per square in the solution and, in general, the algorithm uses space linear in the size of its output.

One may ask whether or not a better approximation factor could be achieved by using strips with different dimensions than 3-by-2. The answer is no, since we require that MIS is solvable exactly on each half-open strip, and strips of larger size do not allow this.

To show this we must first consider the well-known two-party communication problem INDEX, which we define below:

Definition (INDEX). *Alice holds an n-bit string $X \in \{0,1\}^n$ and Bob holds an index $\sigma \in [n]$. Alice sends a single message to Bob, who, upon receiving the message, outputs $X[\sigma]$.*

It is well-known that Alice must essentially send Bob all n bits:

Theorem 2. [7] *The randomised constant error communication complexity of INDEX is $\Omega(n)$.*

Now we show that the construction is optimal via a reduction from INDEX. Party one will construct a set of squares based on the bit string X that they hold. Party two will then add more squares to this construction depending on the answer bit. The size of the MIS of the construction will change depending on the value of the answer bit. Formally:

Theorem 3. [2] *Given a stream of w-by-w squares contained in a $(2+\delta)w$-by-$(2+\delta)w$ region, achieving a $(\frac{3}{2}-\epsilon)$-approximation to the size of the MIS, with constant probability of success, for any $\epsilon, \delta > 0$ requires $\Omega(n)$ space.*

Proof. We fix an instance of INDEX with bit string $X \in \{0,1\}^n$ and query index $\sigma \in [n]$. Fix $\epsilon, \delta > 0$ and let $w = \frac{4n}{\delta}$. Party one will construct the following set of squares: for each $i \in [n]$ with $X_i = 1$ include the square centred on $(\frac{2n}{\delta} + 2i, \frac{2n}{\delta} + 2n + 2 - 2i)$. Observe that for all of the squares added to the construction by party one, any two given squares overlap (refer to 2.4). Therefore, at most one of the squares that party one adds can be a part of the MIS. Observe that, the parties could use a $(\frac{3}{2} - \epsilon)$-approximation

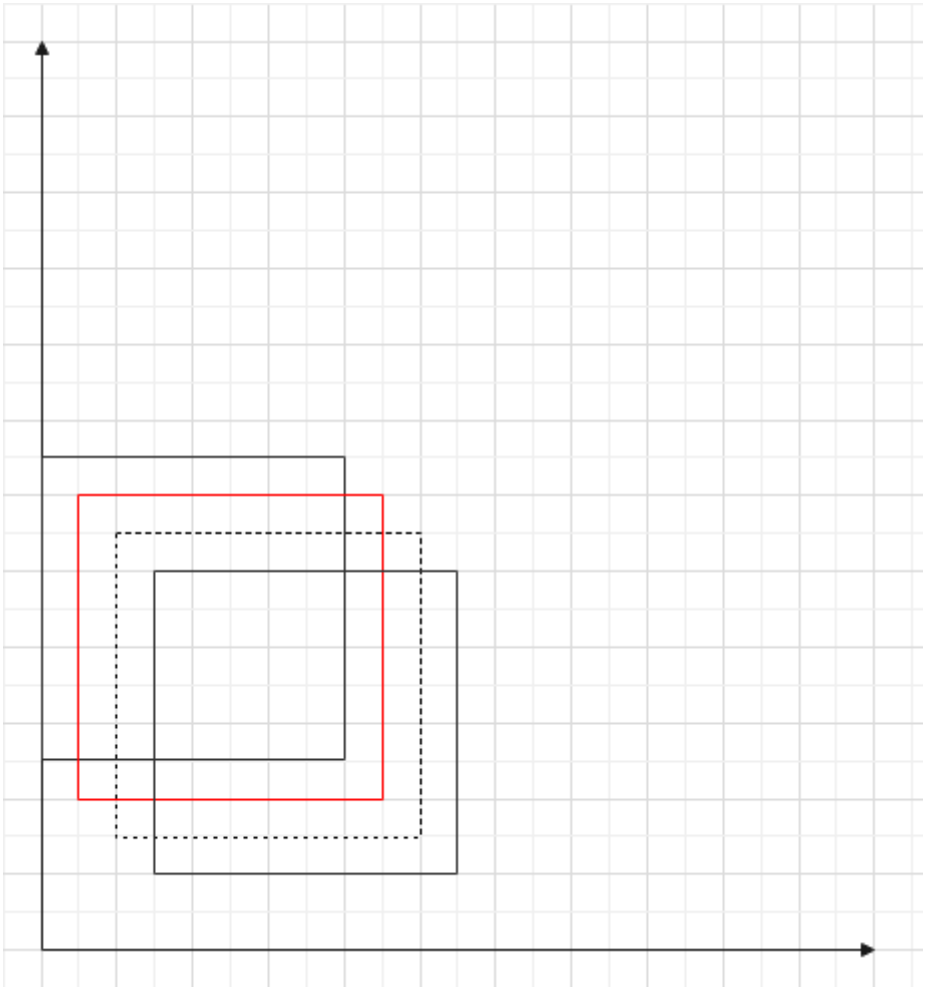


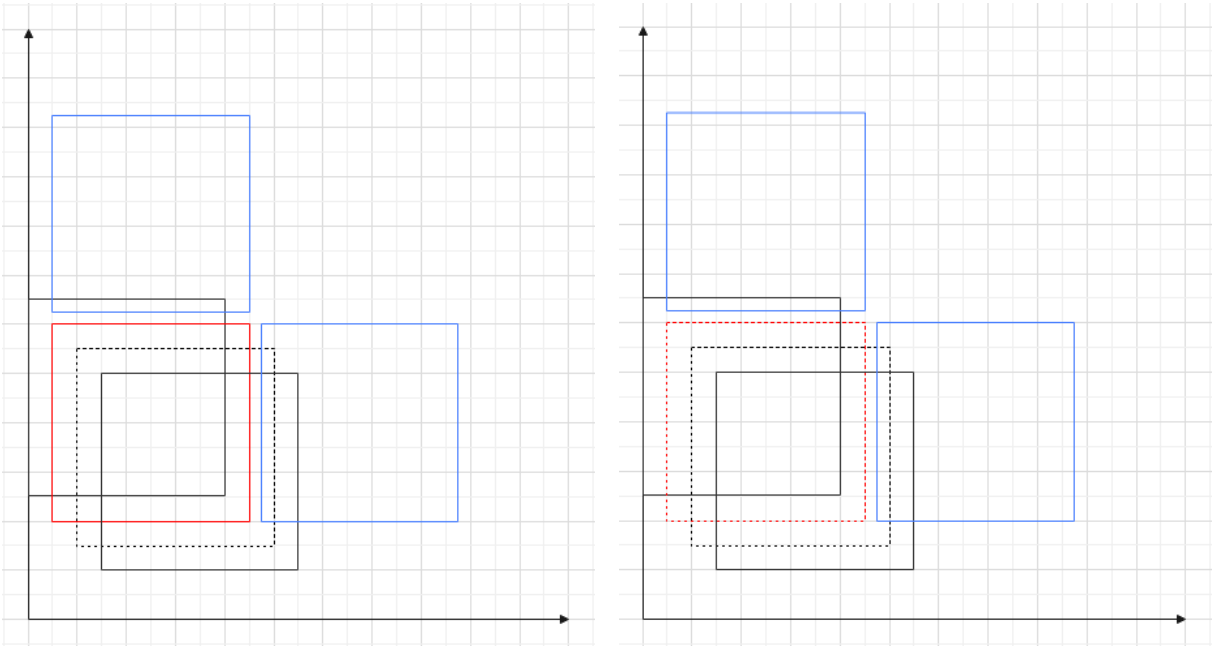
Figure 2.4: Example: $X = (1,1,0,1), \sigma = 1$. The red square corresponds to the answer bit. The dashed square corresponds to the bit which is 0; it is not a part of the construction, but is included here for clarity.

algorithm to allow party two to determine X_σ, because this arrangement of squares is essentially a set of unit intervals, and we know such an algorithm exists [3]. Party two then appends squares centred on $(\frac{6n}{\delta} + 2\sigma + 1, \frac{2n}{\delta} + 2n + 2 - 2\sigma)$ and $(\frac{2n}{\delta} + 2\sigma, \frac{6n}{\delta} + 2n + 3 - 2\sigma)$ (refer to 2.5a, 2.5b). If $X_\sigma = 1$, then there is a square between the two new squares and the size of the MIS as 3. Otherwise, the size of the MIS is 2.

□

2.2 Extension of the Algorithm

One of the problems that naturally follows from the unit square stream is the case where we have squares of arbitrary size. In particular, we would like to develop an algorithm which maintains a good



(a) $X_\sigma = 1$. The blue squares have been added by party two. Along with the red square, they form an MIS of size 3.

(b) $X_\sigma = 0$. The red square is no longer a part of the construction. The MIS consists of only the blue squares and has size 2.

approximation ratio, while still maintaining a relatively good space complexity. We will first consider the problem of squares of two sizes. First observe that there is a trivial 6-approximation using the 3-approximation algorithm given above; we simply run the algorithm first on the squares of one size and then on the squares of the other size, and take the largest output. If we have squares of 3 different sizes, then we have a 9-approximation using the same method, and so on. However, these approximation factors are very quickly becoming unreasonable, and are not appropriate for the overall problem of allowing as many sizes of square as we would desire. It turns out that, so long as the radii of the largest square is less than $\frac{3}{2}$ times that of the smallest square, we can use the same construction as above to approximate MIS on as many sizes of square as we want. Note that:

- The above condition is necessary to ensure that two of the largest size of square will fit side by side in the construction.
- The approximation factor will be inversely proportional to the size of the largest square

The formal proof is as follows:

Theorem 4. *Consider a stream of squares of multiple sizes, and let the smallest square be of size w. Suppose every other square is of a size $x \in [w, \frac{3}{2}w)$, and that the largest square has size v. We have an algorithm for approximating MIS with an approximation factor of $\frac{6w^2}{(3w-v)(2w-v)}$.*

Proof. We will consider a half-open strip with dimensions $[0, 3w)$ x $[0, 2w)$. We can find MIS exactly on the space by storing the leftmost and rightmost squares of each size. By partitioning the entire space as before and considering only the squares falling exactly within a half-open strip, we can determine MIS exactly on this substream of squares. We now consider that we can shift the partition by up to $3w$ units horizontally and $2w$ units vertically, and ask ourselves the probability that any given square of either size falls exactly within a half-open strip. For a square of size w, the probability will be $\frac{1}{3}$ (refer to Theorem 1). For a square of size v, the probability that the square is fully contained in the horizontal axis is $\frac{3w-v}{3w}$, and in the vertical axis it is $\frac{2w-v}{2w}$. By multiplication the final probability is $\frac{(3w-v)(2w-v)}{6w^2}$, meaning that by taking the partition that produces the largest MIS, we obtain an approximation factor of $\frac{6w^2}{(3w-v)(2w-v)}$. □

We by no means suggest that this algorithm is optimal; in fact, this algorithm has a problem with regards to space complexity. Similarly to the algorithm above, the space used depends on the number of

ways the space is partitioned, but additionally in this case, the space used grows linearly with the number of different sizes of square in the input. Considering that we can in theory have an infinite number of different sizes of square, this quickly becomes impractical. However, we have at least shown that for a relatively small number of sizes, we can obtain a decent approximation using the same method.

2.3 The Upper Bound

Finally, we return to the case of unit squares and outline a proof for the upper bound, using our 3-approximation algorithm, although the full details are determined to be beyond the scope of this paper:

Theorem 5. [2] *For a unit square stream, we can* $(3+\epsilon)$*-approximate the* $\alpha(G)$ *with constant probability using* $O(\epsilon^{-2} log\ \epsilon^{-1} + log\ n)$ *space.*

Proof. Recall that when we run the algorithm, we produce many MIS's based on the way that we partition the space. Then observe that if we can get a $(1+\epsilon)$-approximation to the size of each of these MIS's, then we are done, as the largest of these MIS's is the required 3-approximation.

Each strip can contain either 0, 1, or 2 disjoint squares in it. To approximate the size of an MIS, we estimate γ, the number of non-empty strips for a given partition, and δ, the average number of disjoint squares in non-empty strips. Then the MIS has size $\approx \gamma\delta$.

Approximating γ is a distinct elements problem, which we can $(1+\epsilon)$-approximate with constant probability in $O(\epsilon^{-2} + \log n)$ space [6].

Then we can approximate δ by using nearly-uniform permutations to keep a nearly-uniform sample of the non-empty strips [1]. □

In summary, we have a 3-approximation algorithm for MIS on unit square streams and this algorithm leads to an upper bound of 3 for estimating the size of the MIS. Furthermore, when given squares of arbitrary size as discussed in Section 2.2, the same method will give an upper bound equal to the approximation factor in a similar way. The proof is essentially the same as that of Theorem 5 above and so is omitted. The next chapter explores the lower bound.

Chapter 3

The Lower Bound

In this chapter we will discuss the lower bound for the problem of approximating the size of the MIS on a unit square stream. It is proved via a reduction from a multi-party communication problem known as CHAIN that we will introduce now.

3.1 The Communication Problem CHAIN

CHAIN is a generalisation of the 2-party communication problem INDEX that we introduced in Section 3.1. It works by chaining together multiple independent cases of the INDEX problem which have the same "answer bit" $a \in \{0, 1\}$. Each party aside from the last holds an n-bit vector which contains the answer bit somewhere with in it. Each party aside from the first holds an index giving the location of the answer bit in the previous party's vector. Party 1 sends a message to party 2, who upon receipt sends a message to party 3, and so on. We give the formal definition below:

Definition ($\mathbf{CHAIN}_k$). An instance of $CHAIN_k$ consists of $k-1$ n-bit binary vectors $X^{(i)} \in \{0,1\}^n$, $k-1$ indices $\sigma_i \in [n]$ and an answer bit $a \in \{0, 1\}$. The index σ_i corresponds to the location of the answer bit in $X^{(i)}$; in other words, we are guaranteed that for all i $X^{(i)}_{\sigma_i}$ is equal to the answer bit a. We proceed as follows:

- The first party P_1 knows $X^{(1)}$
- Each intermediate party P_i for $1 < i < k$ knows $X^{(i)}$ and σ_{i-1}
- The final party P_k knows just σ_{k-1}

P_1 sends a single message to P_2, then P_2 sends a message to P_3 and so on, until P_{k-1} sends a message to P_k. P_k must output the answer bit a correctly with probability at least $\frac{2}{3}$.

We also have the following:

Theorem 6. [2] *Any communication scheme which solves $CHAIN_k$ must communicate at least $\Omega(\frac{n}{k^2})$ bits.*

The proof for this theorem is via a reduction from a different multi-party communication problem, but the details of this reduction are considered to be outside the scope of this project, and as such, are omitted. Those interested may refer to section 3.1 of [2].

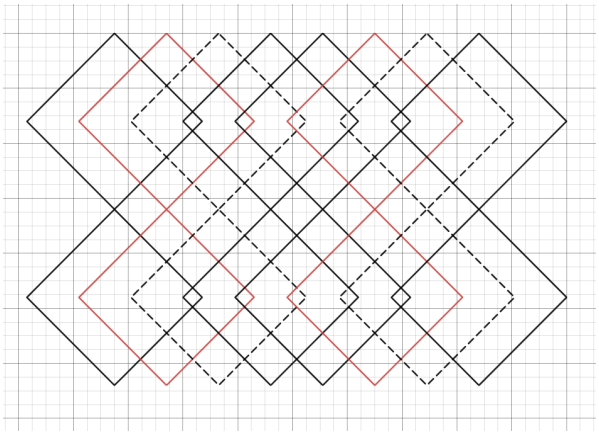
3.2 Proof for the Lower Bound

Finally, we have the proof for the lower bound for approximating the size of MIS on a unit square stream. The proof is via a reduction from the chained index communication problem introduced above and uses similar methods that that of Theorem 3. This gives us our lower bound of $\frac{5}{2}$.

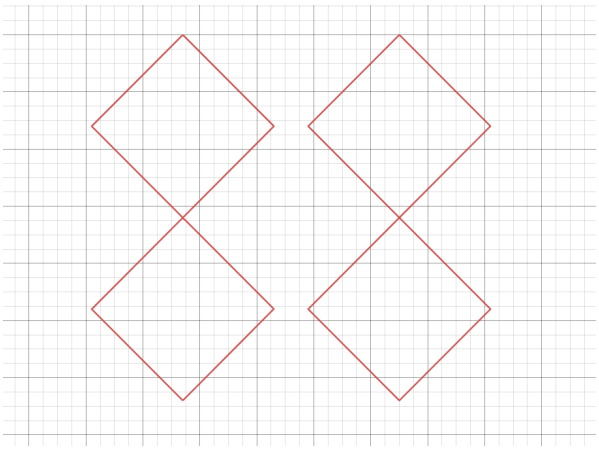
Theorem 7. [2] *Achieving a $(\frac{5}{2}-\epsilon)$-approximation to the size of the MIS on a unit square stream with constant probability of success requires $\Omega(n)$ space for any $\epsilon > 0$.*

Proof. Fix an instance of $CHAIN_3$ with n-bit vectors and an integer $k \in [n]$. Each of the 3 parties will construct a set of squares based on the vectors information they have (i.e. the vectors and indices) such that the MIS will be either large or small depending on the answer bit. Then, a streaming algorithm for approximating the size of the MIS can be used to solve the communication problem, which gives a space lower bound.

Party one adds the following squares to the construction, based on $X^{(1)}$: for every entry such that $X_i^{(1)} = 1$ and for every $j \in [k]$, add squares of radius $2n^2$ centered at $(i(4n+3)+(j+1)(4n^2+3n), 4n^2)$ and $(i(4n+3)+(j+1)(4n^2+3n), 8n^2)$. This makes two horizontal lines of squares stacked on top of each other (Figure 3.1a). The first n locations are associated with the n bits of $X^{(1)}$, with a squares placed for bits that are 1, and omitted for those that are 0. This is then repeated for a total of k times, and the second row is an exact copy of the first. Note that all n squares which correspond to any one given instance of $X^{(1)}$ overlap, and so the size of the MIS on this collection of squares is $2k$. In particular, the collection of squares associated with any index in $X_i^{(1)} = 1$ forms an MIS of size $2k$ (Figure 3.1b).



(a) Example: $X^{(1)} = 1101, \sigma_1 = 2, X^{(2)} = 1001, \sigma_2 = 4, k = 2$. This is the construction formed by party one. The red square corresponds to the answer bit; the dashed square corresponds to a square omitted due to a 0 in the bit string.



(b) We remove all the other squares to clearly see that the squares corresponding to the answer bit form an independent set.

Party two adds the following squares to the construction, based on $X^{(2)}$ and σ_1, such that the size of the MIS increases only when the answer bit $X_{\sigma_1}^{(1)} = 1$: for every entry such that $X_i^{(2)} = 1$ and for every $j \in [k]$, add a square of radius $2n^2$ centered at $(\sigma_1(4n+3)+(j+\frac{3}{2})(4n^2+3n), 6n^2-n+2i)$. This produces k columns of n balls lined up to fit in the gaps between the squares corresponding to the answer bit $X_{\sigma_1}^{(1)}$ (assuming those squares are present) (Figure 3.2). Observe that all squares in one column overlap, and so the size of the MIS on **only the squares added by party two** is k, and that once again, all squares associated with any index such that $X_i^{(1)} = 1$ form an MIS of size k. At this point refer to 3.3 and 3.4.

Party three adds the following squares to the construction, based on σ_2 and σ_1 (σ_1 can easily be sent to party three along with party two's message): for every $j \in [k]$, add squares of radius $2n^2$ centred at $(\sigma_1(4n+3)+(j+\frac{3}{2})(4n^2+3n), 10n^2-n+2\sigma_2+1)$ and $(\sigma_1(4n+3)+(j+\frac{3}{2})(4n^2+3n), 2n^2-n+2\sigma_2-1)$. These square at the top and bottom of each of the columns from party two surrounding the square corresponding to $X_{\sigma_2}^{(2)}$. In total, party three adds $2k$ squares to the construction, and as none of these squares overlap, the size of the MIS on only the squares added by party three is $2k$ (Figure 3.5).

We now wish to determine the size of the MIS for the entire construction. In the case that the answer bit is 1 ($X_{\sigma_1}^{(1)} = X_{\sigma_2}^{(2)} = 1$), we can take the MIS associated with $X_{\sigma 1}^{(1)}$, which is of size $2k$, and the MIS associated with $X_{\sigma_2}^{(2)}$, which is of size k, and all the squares added by party three to form an MIS of size $5k$ (Figure 3.6). In the case that the answer bit is 0 ($X_{\sigma_1}^{(1)} = X_{\sigma_2}^{(2)} = 0$), taking any square from party one (except the left most $\sigma_1 - 1$ squares in each row) excludes every square in a column from party two and a square from party three, and similar exclusions occur between the other pairs of constructions. So the best MIS that we can form comes from taking an MIS from party one associated with an index smaller than σ_1 (size $2k$) and one square each from parties two and three in the rightmost column, giving a total

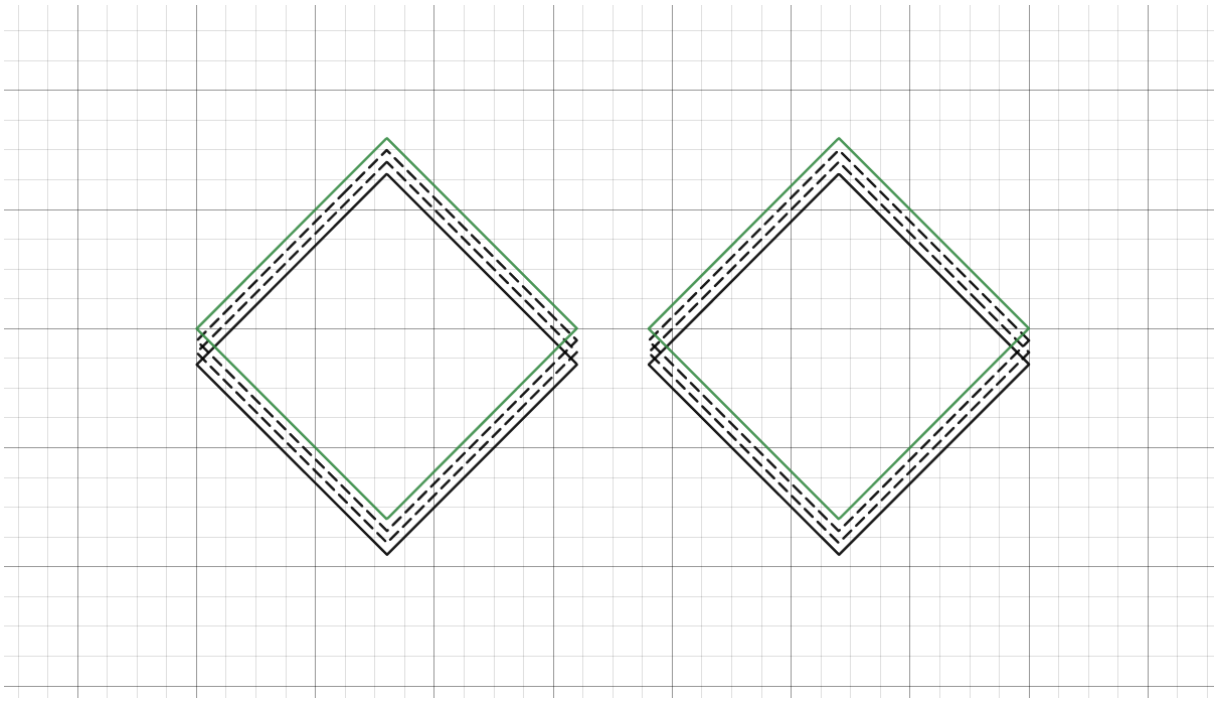


Figure 3.2: Continuation of the above example. Party two adds the above squares to the construction. The dashed squares are omitted, and the green squares correspond to the answer bit. Note that the green squares form an MIS of size k.

MIS of size $2k + 2$.

Thus, any streaming algorithm achieving an approximation factor better than $\frac{5k}{2k+2}$ must use $\Omega(n)$ space, for any constant k (just take n large enough to allow such a k). □

Of particular interest is the fact that, when allowed squares of any two different sizes, this lower bound immediately jumps to 3:

Theorem 8. [2] *Achieving a $(3 - \epsilon)$-approximation to the size of the MIS on a stream of squares of two different sizes with constant probability of success requires $\Omega(n)$ space for any $\epsilon > 0$.*

Proof. This construction is similar to the previous one. The first party inserts k rows of squares, rather than 2. The second party inserts columns between every neighbouring pair of rows from party one, and the third party puts smaller squares between the columns from party two such that they are disjoint from the squares corresponding to the answer bit but overlap the squares on either side. When the answer bit is 0, this gives an MIS of size $k^2 + k + 1$ and when the answer bit is 1 this gives us an MIS of size $3k^2$, from which the result follows. □

This result seems to suggest that, in the case of a unit square stream, our upper bound of 3 is in fact correct, and that our 3-approximation algorithm is optimal.

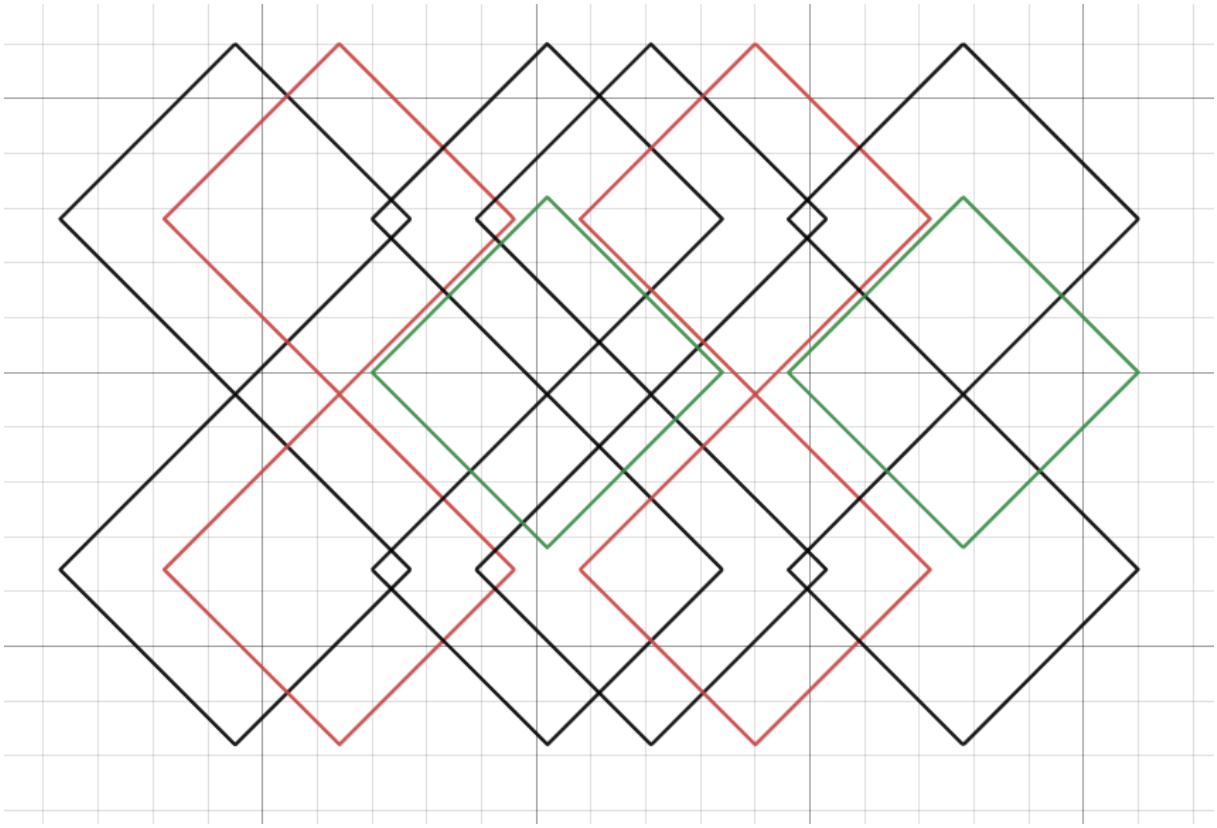


Figure 3.3: Above is how the answer bit squares from party two fit among the squares from party one. Notice that all non-answer bit squares from party one overlap with the answer bit squares from party two, with the exception of squares to the left of the first red squares.

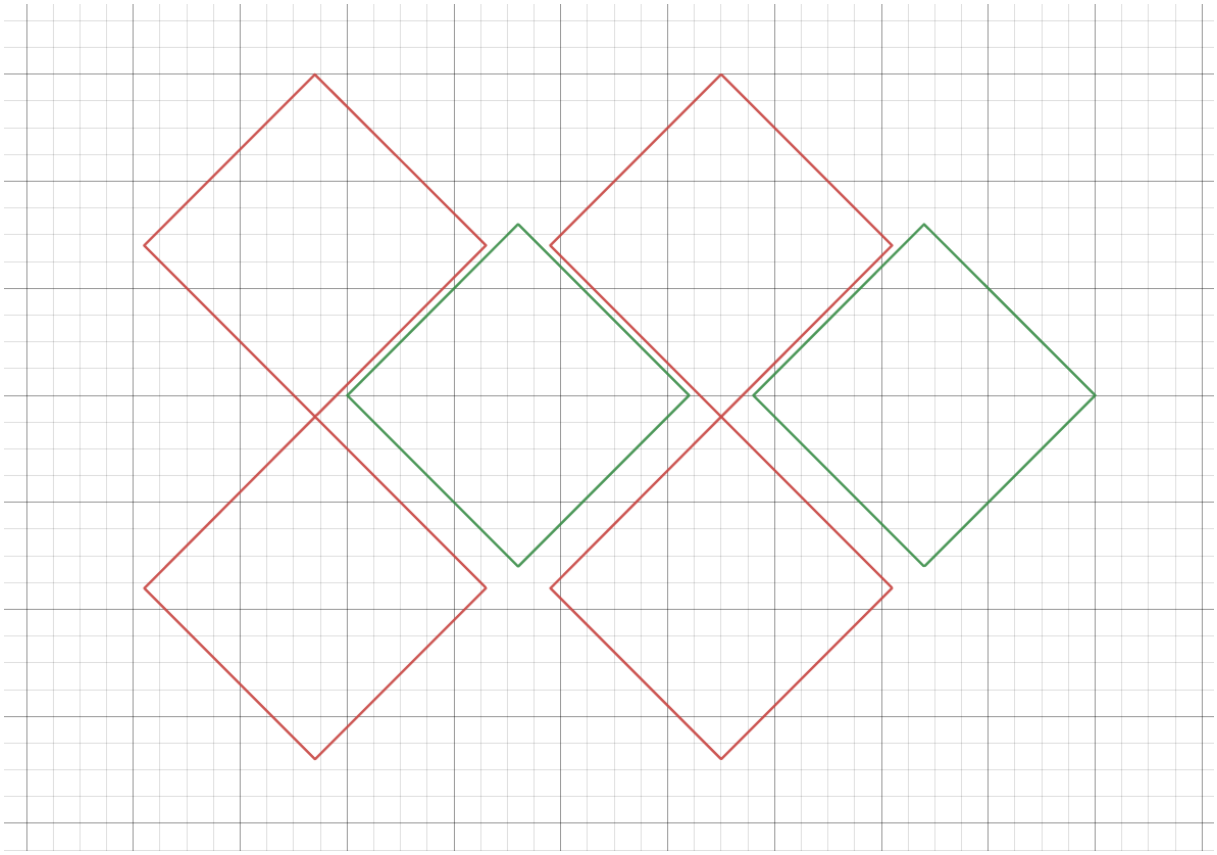


Figure 3.4: All the answer bit squares so far. They form an MIS of total size $3k$.

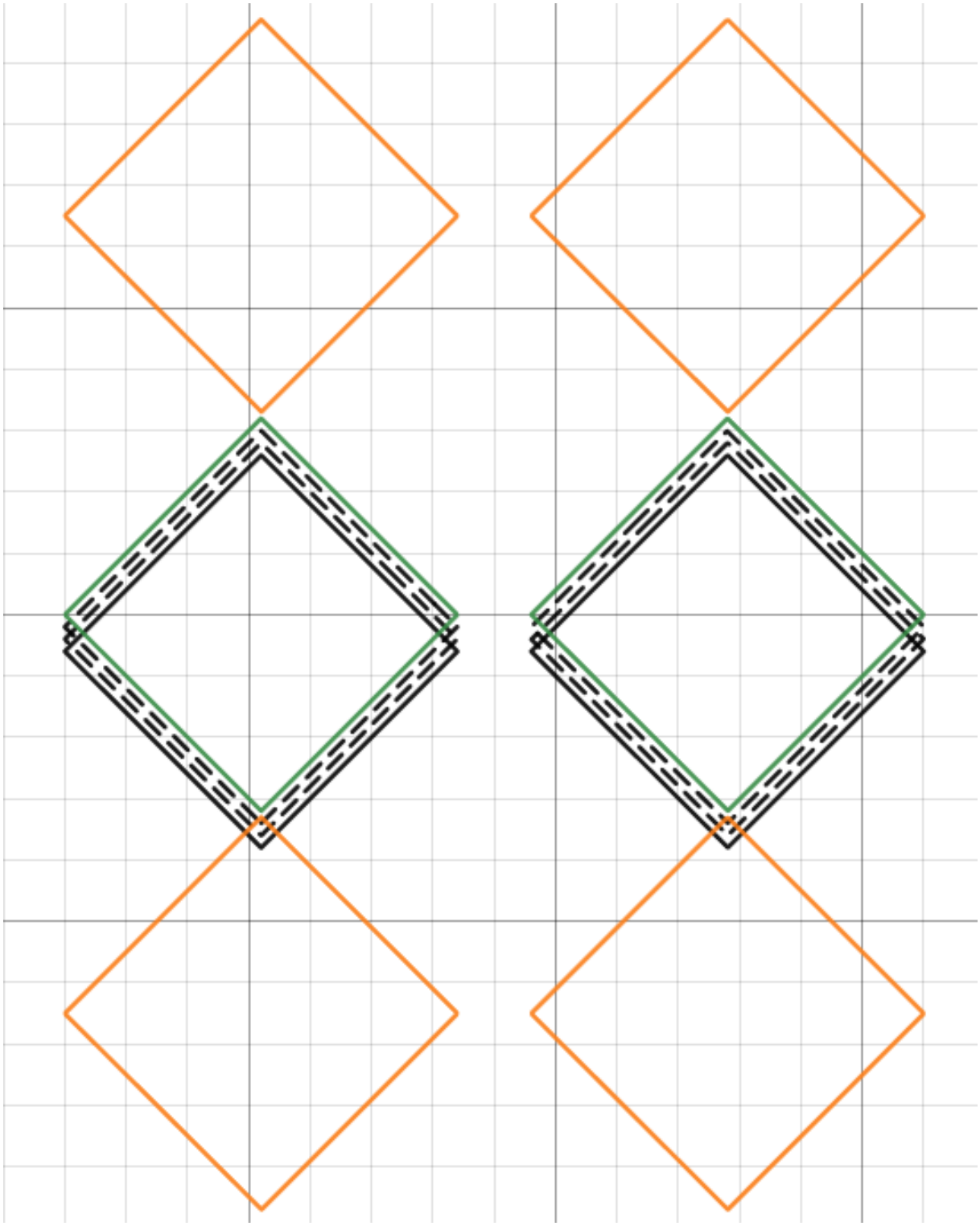


Figure 3.5: The orange squares are those added by party three, and obviously form an MIS of size $2k$. This is how they fit in relation to the squares added by party two.

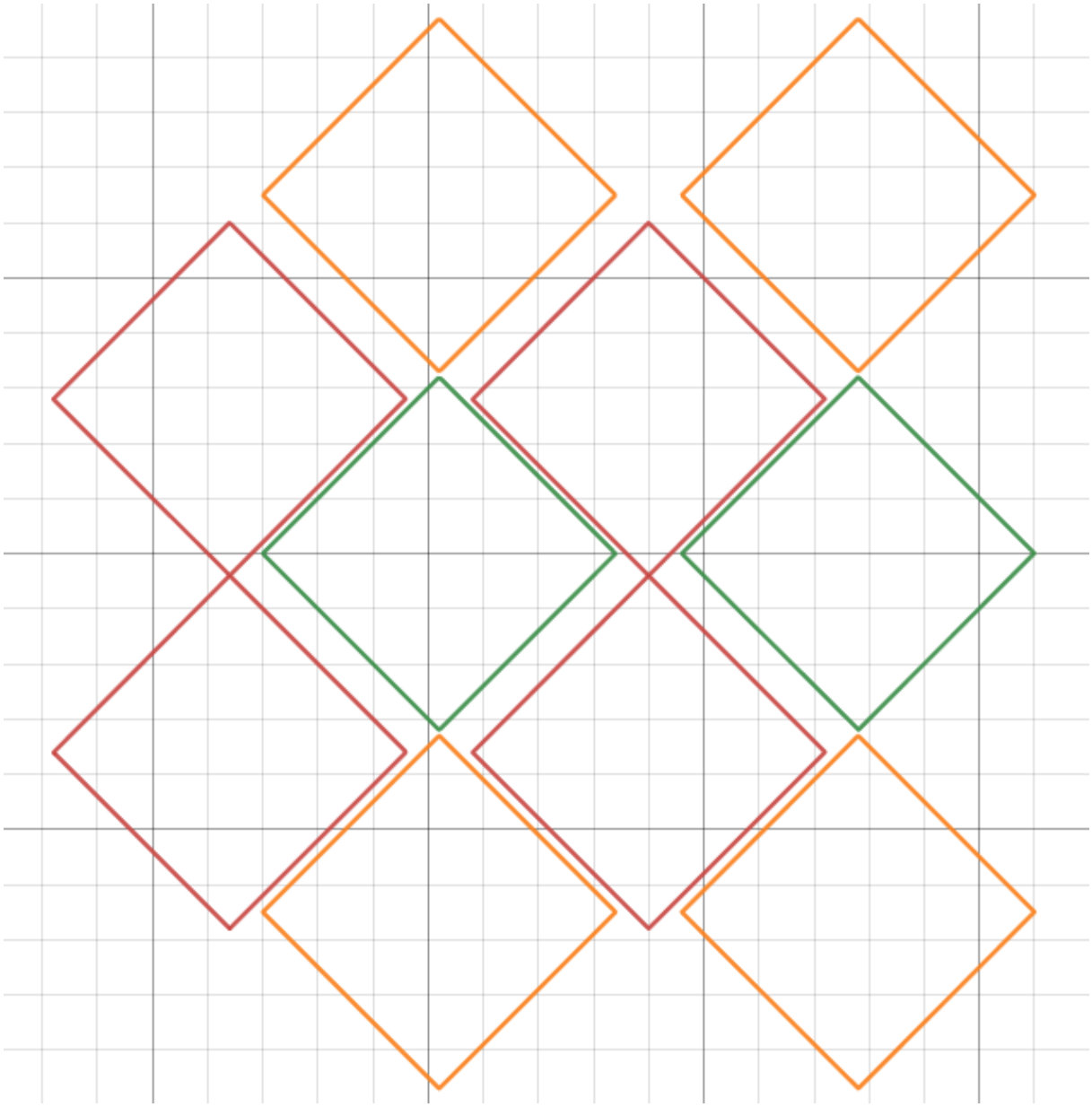


Figure 3.6: All the answer bit squares. They form an MIS of total size $5k$.

Chapter 4

Conclusion

Throughout this paper we have introduced and summarised the current state of the problem of MIS on squares in the streaming setting. We have 3-approximation algorithm for finding MIS on a unit square stream. On a stream of squares of arbitrary sizes, for squares all a relatively similar size, we can use a similar algorithm to compute MIS with a slightly worse approximation factor dependent on the size of the largest square. Finally, for the case of a unit square stream, we have upper and lower bounds for estimating the size of the MIS, which are 3 and $\frac{5}{2}$ respectively.

The biggest unanswered question which still remains is that of which of these bounds is correct, and there seems to be evidence for both. On the one hand, when allowed squares of more than one size, the lower bound increases to 3; this would seem to suggest that the upper bound is correct, and that our algorithm for finding MIS on a unit square stream is in fact optimal. To argue against, we will look briefly again at the one-dimensional case of interval selection. The approximation factor for unit intervals is $\frac{3}{2}$, which is better than the approximation factor for general sizes of interval which is 2. We may expect this logic to extend to the two-dimensional case; in other words, that the approximation factor for unit squares is better than for general sizes of square. Then, the lower bound for 2 sizes of square being 3 may suggest (though does not strictly imply), the existence of a streaming algorithm for unit squares with an approximation factor better than 3.

We find the second argument to be lacking, partially because problems with approximation factors of $\frac{5}{2}$ are very rare, but mostly because there is not much reason to believe that the approximation factors from the one-dimensional case extend to the two-dimensional case in this way. In fact, the current literature (including this paper) may appear to present the cases as more similar than they are due to the algorithms for streams of squares being created by re-working an algorithm for interval selection. Overall, we are of the opinion that it is more likely for the correct answer for the problem of unit squares is 3; however, a proof for this lower bound is yet to be developed.

In addition, the problem of finding an approximate MIS for arbitrary sizes of square remains open. Although we have proved that we can obtain non-trivial approximation factors for this problem, they are heavily conditional. In particular the methods presented here only work for squares within a very narrow size window of one another, and upon increasing the number of different sizes of square in the input, the space used eventually becomes unreasonable. We still lack an efficient algorithm for the general version of this problem.

Bibliography

[1] S. Cabello and P. Pérez-Lantero. Interval selection in the streaming model, 2015.

[2] G. Cormode, J. Dark, and C. Konrad. Independent sets in vertex-arrival streams, 2018.

[3] Y. Emek, Magnús M. Halldórson, and A. Rosén. Space-constrained interval selection, 2012.

[4] M. R. Henzinger, P. Raghavan, and S. Rajagopalan. Computing on data streams, 1998.

[5] A. Hossain, E. Lopez, and S. M. Halper. Automated design of thousands of nonrepetitive parts for engineering stable genetic systems, 2020.

[6] D. M. Kane, J. Nelson, and D. P. Woodruff. An optimal algorithm for the distinct elements problem, 2010.

[7] I. Kremer, N. Nisan, and D. Ron. On randomized one-round communication complexity, 1999.

[8] S. S. Skiena. The algorithm design manual, 2012.