The Four Corners Problem Over Larger Alphabets

Daniel Průša
Czech Technical University

Michael Wehar
Bryn Mawr College

Chen Xu
University at Buffalo

Developments in Language Theory (DLT 2026)

Outline

1. Background & Prior Works
2. Corner Pattern Cases
3. Conditional Lower Bounds
4. Upper Bound for Square Case
5. Conclusion

The Four Corners Problem

- We study a 2D combinatorial pattern matching problem.
- Fix an alphabet Σ and a corner pattern $\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$ where $a, b, c, d \in \Sigma$.
- **Input:** A matrix with entries over Σ.
- **Question:** Does there exist a rectanglar subblock within the input matrix where the upper left corner is a, upper right is b, lower left is c, and lower right is d?
- We denote this problem by $\texttt{Rect-}\Sigma^{m\times n}\texttt{-}\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$.

Corner Pattern Examples

- Consider two examples of corner patterns $\left[\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$:

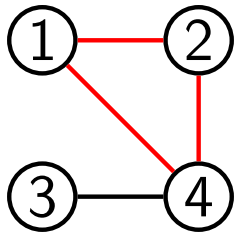
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Known Upper Bounds

- For any finite alphabet Σ and corner pattern $\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$, $\texttt{Rect-}\Sigma^{m\times n}\texttt{-}\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$ is an example of a local picture language (LOC) matching problem.
- We previously showed that all LOC matching problems are solvable in $O(mn \min\{m, n\})$ time (Mráz, Průša, Wehar; CIAA 2019).
- Furthermore, we showed that when $\Sigma = \{0, 1\}$, $\texttt{Rect-}\Sigma^{m\times n}\texttt{-}\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$ is solvable in $\tilde{O}(mn)$ time for all corner patterns (DLT 2020).
- **Question:** Can we solve $\texttt{Rect-}\Sigma^{m\times n}\texttt{-}\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$ in $\tilde{O}(mn)$ time when Σ contains more than two symbols?

Triangle Finding

- Given a graph $G = (V, E)$, do there exist vertices v_1, v_2, $v_3 \in V$ such that $\{v_1, v_2\}$, $\{v_2, v_3\}$, $\{v_3, v_1\} \in E$?



- Solvable in $O(n^\omega)$ time where $n = |V|$ and $\omega < 2.373$ denotes the matrix multiplication exponent.
- It is unknown if there is an algorithm solving it in $\tilde{O}(n^2)$ time.

Triangle Finding Hardness

- A problem is **Triangle Finding-hard** if it is at least as hard as Triangle Finding.
- In other words, a problem X is Triangle Finding-hard if there exists an efficient reduction from Triangle Finding to X.
- A reduction is a function r that translates graphs to instances of problem X such that G is a graph that contains a triangle if and only if $r(G)$ is a positive instance of problem X.
- Our reductions reduce graphs with n vertices to matrices with $\tilde{O}(n)$ rows and $\tilde{O}(n)$ columns in at most $\tilde{O}(n^2)$ time.

Outline

1. Background & Prior Works
2. Corner Pattern Cases
3. Conditional Lower Bounds
4. Upper Bound for Square Case
5. Conclusion

Corner Patterns Cases

- When $\Sigma = \{0, 1\}$, there are $2^4 = 16$ different possible cases.
- All 16 cases can be reduced to the following 4 cases by simple transformations (rotation, bit-flip): $\left[\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right]$, $\left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$, $\left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $\left[\begin{smallmatrix} 1 & 0 \\ 1 & 0 \end{smallmatrix}\right]$.
- When $\Sigma = \{0, 1, 2\}$, we can reduce down to six distinct cases. The two additional cases are: $\left[\begin{smallmatrix} 0 & 1 \\ 2 & 0 \end{smallmatrix}\right]$, $\left[\begin{smallmatrix} 0 & 0 \\ 1 & 2 \end{smallmatrix}\right]$.
- Let $\Sigma = \{0, 1, \ldots, C\}$ for some $C \geq 3$, we can reduce down to seven distinct cases. The one additional case is: $\left[\begin{smallmatrix} 0 & 1 \\ 2 & 3 \end{smallmatrix}\right]$.

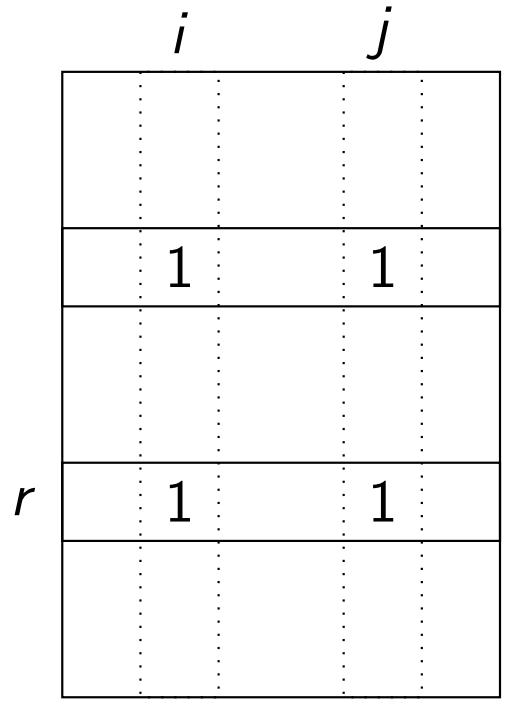
Prior Results

- All four cases over a binary alphabet are solvable in $\tilde{O}(mn)$ time.
  - $\left[\begin{smallmatrix}1 & 1\\ 1 & 1\end{smallmatrix}\right]$: Column Pairs Method (we'll see shortly)
  - $\left[\begin{smallmatrix}1 & 0\\ 1 & 1\end{smallmatrix}\right]$: Reducing Matches to Minimal Matches (DLT 2020)
  - $\left[\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right]$: Divide and Conquer (DLT 2020)
  - $\left[\begin{smallmatrix}1 & 0\\ 1 & 0\end{smallmatrix}\right]$: Column Pairs + Divide and Conquer (DLT 2020)
- One case over ternary alphabet was previously known to be hard.
  - $\left[\begin{smallmatrix}1 & 0\\ 1 & 1\end{smallmatrix}\right]$ with $\Sigma = \{0, 1, 2\}$ is Triangle Finding-Hard. (CIAA 2019)

Algorithm for Rect-$\{0,1\}^{m\times n}$-$\left[\begin{smallmatrix}1&1\\1&1\end{smallmatrix}\right]$

Input: $\mathrm{M} \in \{0,1\}^{m,n}$ (assume $m \geq n$)

```
1:  S := ∅
2:  for each row r of M do
3:      R := ∅
4:      for each i s.t. M(r, i) = 1 do
5:          add i to R
6:      end for
7:      for each i, j ∈ R s.t. i < j do
8:          if S contains (i, j) then
9:              return matching submatrix
10:         else
11:             add (i, j) to S
12:         end if
13:     end for
14: end for
```



$$O\left(n^2+mn\right) = O\left(mn\right)$$

New Results

- Cases over alphabet $\Sigma = \{0, 1, 2\}$:
  - $\left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$: Triangle Finding-Hard (previously known)
  - $\left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$: Triangle Finding-Hard
  - $\left[\begin{smallmatrix} 0 & 1 \\ 2 & 0 \end{smallmatrix}\right]$: Triangle Finding-Hard
  - $\left[\begin{smallmatrix} 0 & 0 \\ 1 & 2 \end{smallmatrix}\right]$: Triangle Finding-Hard
  - $\left[\begin{smallmatrix} 1 & 0 \\ 1 & 0 \end{smallmatrix}\right]$: Solvable in $O(mn + n^2)$ time using column pairs method, but cannot be solved in $\tilde{O}(mn)$ time unless 4-Clique is in $\tilde{O}(n^3)$ time!
- Case over alphabet $\Sigma = \{0, 1, 2, 3\}$:
  - $\left[\begin{smallmatrix} 0 & 1 \\ 2 & 3 \end{smallmatrix}\right]$: Triangle Finding-Hard
- Case over alphabet $\Sigma = \{0, 1, \ldots, C\}$ for $C \geq 2$:
  - $\left[\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right]$: Solvable in $O(mn)$ time using column pairs method.

Outline

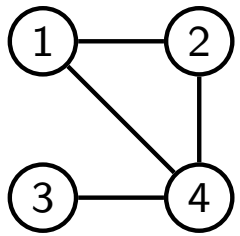
1. Background & Prior Works
2. Corner Pattern Cases
3. Conditional Lower Bounds
4. Upper Bound for Square Case
5. Conclusion

Rect-$\{0,1,2\}^{m\times n}$-$\left[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right]$ is Triangle Finding-hard

Theorem

Rect-$\{0,1,2\}^{m\times n}$-$\left[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right]$ cannot be solved in $\tilde{O}(mn)$ time unless Triangle Finding can be solved in $\tilde{O}(n^2)$ time.

G:



	1	2	3	4
1	2	1	2	1
2	1	2	2	1
3	2	2	2	1
4	1	1	1	2

$$\left[\begin{array}{cccc|cccc}
2 & \boxed{1} & 2 & 1 & 2 & 0 & 2 & \boxed{0} \\
1 & 2 & 2 & 1 & 0 & 2 & 2 & 0 \\
2 & 2 & 2 & 1 & 2 & 2 & 2 & 0 \\
1 & 1 & 1 & 2 & 0 & 0 & 0 & 2 \\
\hline
2 & 0 & 2 & 0 & 1 & 2 & 2 & 2 \\
0 & 2 & 2 & 0 & 2 & 1 & 2 & 2 \\
2 & 2 & 2 & 0 & 2 & 2 & 1 & 2 \\
0 & \boxed{0} & 0 & 2 & 2 & 2 & 2 & \boxed{1}
\end{array}\right]$$

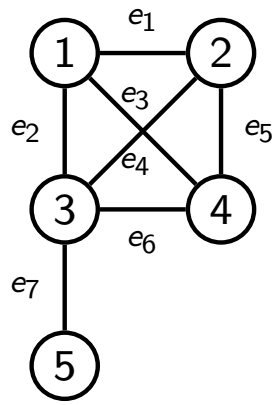
$G = (V, E)$ contains a triangle iff M contains submatrix $\left[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right]$.

Rect-$\{0,1,2\}^{m\times n}$-$\left[\begin{smallmatrix}1&0\\1&0\end{smallmatrix}\right]$ is 4-Clique-hard

Theorem

Rect-$\{0,1,2\}^{m\times n}$-$\left[\begin{smallmatrix}1&0\\1&0\end{smallmatrix}\right]$ cannot be solved in $\tilde{O}(mn)$ time unless 4-Clique is solvable in $\tilde{O}(n^3)$ time.

G:



$$\left[\begin{array}{ccccccc|ccccccc}
1 & \boxed{1} & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 0 & \boxed{0} & 0 & 2 \\
1 & 2 & 2 & 1 & 1 & 2 & 2 & 2 & 0 & 0 & 2 & 2 & 0 & 2 \\
2 & \boxed{1} & 2 & 1 & 2 & 1 & 1 & 0 & 2 & 0 & 2 & \boxed{0} & 2 & 2 \\
2 & 2 & 1 & 2 & 1 & 1 & 2 & 0 & 0 & 2 & 0 & 2 & 2 & 2 \\
2 & 2 & 2 & 2 & 2 & 2 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2
\end{array}\right]$$

dims $= |V| \times 2|E|$

$G = (V, E)$ contains a 4-Clique iff M contains submatrix $\left[\begin{smallmatrix}1&0\\1&0\end{smallmatrix}\right]$.

Outline

1. Background & Prior Works
2. Corner Pattern Cases
3. Conditional Lower Bounds
4. Upper Bound for Square Case
5. Conclusion

Square Case

- Fix an alphabet Σ and a corner pattern $\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$.
- Given an input matrix M over Σ, we consider whether M contains a subblock with equal width and height matching the corner pattern.
- We denote this problem by $\texttt{Square-}\Sigma^{m\times n}\texttt{-}\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$.
- We can simply solve this problem in $O(mn \min\{m, n\})$ time by iterating over each entry a (as the upper left corner) and iterating over each possible side length to check if it forms a matching square.
- **Question:** Can we do better? Can we solve $\texttt{Square-}\Sigma^{m\times n}\texttt{-}\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$ in strongly subcubic time?

A Small Algorithmic Improvement: Part 1

Theorem

Square-$\Sigma^{m\times n}$*-*$\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$ *is solvable in* $O(\frac{mn\min\{m,n\}}{\sqrt{\log(m+n)}})$ *time.*

- **Basic Idea:** Break your m by n matrix into blocks of size $\mathcal{C}\cdot\sqrt{\log(m+n)}$ by $\mathcal{C}\cdot\sqrt{\log(m+n)}$.
- Now, we have a matrix of $\frac{m}{\mathcal{C}\cdot\sqrt{\log(m+n)}}$ by $\frac{n}{\mathcal{C}\cdot\sqrt{\log(m+n)}}$ blocks.
- If there is a square matching the corner pattern, then there is a combination of at most 4 blocks containing the corners of the square.

A Small Algorithmic Improvement: Part 2

Theorem

Square-$\Sigma^{m\times n}$*-*$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ *is solvable in* $O(\frac{mn\min\{m,n\}}{\sqrt{\log(m+n)}})$ *time.*

- There are at most $\frac{mn\min\{m,n\}}{\mathcal{C}^3\cdot\log^{1.5}(m+n)}$ combinations to consider.
- However, there are at most $|\Sigma|^{4\cdot\mathcal{C}^2\cdot\log(m+n)}$ possible combinations so some will appear multiple times.
- Let us cache the results so that we don't need to redo work.
- If we pick $\mathcal{C}$ sufficiently small relative to the alphabet size, then we can precompute all of these combination cases in $\tilde{O}(mn)$ time storing the results in a table.
- Lookups in the table will take $O(\log(m+n))$ time. As a result, checking all cases will take $O(\frac{mn\min\{m,n\}}{\sqrt{\log(m+n)}})$ time.

Outline

1. Background & Prior Works
2. Corner Pattern Cases
3. Conditional Lower Bounds
4. Upper Bound for Square Case
5. Conclusion

Conclusion

We showed the following:

- Five cases of the four corners problem over larger than binary alphabets are Triangle Finding-Hard.
- One particular case (namely, $\left[\begin{smallmatrix}1 & 0\\ 1 & 0\end{smallmatrix}\right]$) is solvable in $O(mn + n^2)$ time, but is not solvable in $\tilde{O}(mn)$ time unless 4-Clique is in $\tilde{O}(n^3)$ time.
- All square cases are solvable slightly faster than cubic time.

Thank You!

Questions or Comments?