Analysis of Unique Solutions in Binary Linear Systems with Structured Matrices

Name: Nikita Shah
Enrollment ID: **202101209**
Email: 202101209@daiict.ac.in
B. Tech. Project (BTP) Report
BTP Mode: ***On Campus***
DA-IICT
Gandhinagar, India

Mentor: Prof. Arpita Mal
Email: arpita_mal@daiict.ac.in
DA-IICT
Gandhinagar, India

***Abstract*—This project investigates the uniqueness of solutions in binary linear systems characterized by a structured matrix, interacting with binary vectors from sets X and Y, where entries are limited to -1, 1 and vectors in Y have their final entry fixed at 1. The primary goal is to study the linear systems of the form $yTx = 1 \forall x \in X, y \in Y$, and to identify the structural properties of B that ensure a unique solution exists. Using techniques from combinatorial matrix theory, linear algebra, and coding theory, we develop mathematical criteria for uniqueness and investigate computational approaches for solution verification. The project provides new insights into binary matrix interactions and opens pathways for applications in coding theory and optimization.**

***Index Terms*—Combinatorial Matrix Theory, Linear Algebra, Coding Theory, Computational Mathematics, Binary Linear Systems, Structured Matrices, Invertibility Conditions, Computational Mathematics, Solution Space Analysis, Hadamard Matrices, Mask Filtering**

I. Introduction

This project investigates binary linear systems where the core unknown is a variable matrix $\boldsymbol{T} \in \mathbb{R}^{n \times n}$, and the system structure is governed by binary vectors from sets X and Y with entries in $\{-1, 1\}$. The main objective is to analyze the bilinear form $y^T Tx = 1 \ \forall x \in X$ and $\forall y \in Y$, and determine when the system admits a unique solution. To model the problem, a large structured matrix B is constructed, capturing all combinations of X and Y. Solving the problem reduces to selecting n^2 independent rows from B to form a system with a nonzero determinant. Due to the exponential growth of possibilities with n, optimization strategies such as dependency analysis, structured grouping of rows, and parallel computation in C++ were developed. This work contributes both theoretical insights into the uniqueness of binary matrix systems and computational methods to solve large instances efficiently.

II. Problem Formulation

For a given natural number n, we define:

- X: A $2^n \times n$ matrix where each row is a vector with entries in $\{-1, 1\}$.
- Y: A $2^{n-1} \times n$ matrix, with each row a vector in $\{-1, 1\}^n$.
- T: A matrix of size $n \times n$ containing n^2 variables.

The system of equations is constructed by considering the bilinear form

$$y^T Tx = 1, \quad \forall\, y \in Y, \forall\, x \in X.$$

The number of potential equations is 2^{2n-1}, while the number of variables is n^2. Our goals are:

- To select n^2 suitable equations forming a full-rank system.
- To find the corresponding unique solution t (vectorized form of T).
- To determine and count distinct solutions.

The construction leads to:

- Matrix B:= $2^{2n-1} \times n^2$
- Matrix b: Column vector with n^2 rows with all entries 1.

the core mathematical problem is to select n^2 independent rows of B to form a square matrix A. The criteria for the system

$$A\, t = b$$

to have a unique solution is $\det(A) \neq 0$. The goal is to find the number of distinct unique solutions as well as efficient methods to identify the independent sets of rows without performing an exhaustive search over all combinations.

III. Methadology

This section explains the approach in detail, highlighting the mathematical modeling, computational construction, and optimization techniques.

A. Matrix Construction

We had sets Xa nd Y as binary vectors of length n. We represent them as matrices with the vectors as rows of matrices X and Y.

- X := $2^n \times n$
- Y := $2^{n-1} \times n$

The structured matrix B was built as follows:

- For each pair (y, x) where $y \in Y$ and $x \in X$.

- The coefficient corresponding to a variable t_{ij} in the resulting vector is the product $y_i x_j$
- Thus, each row of B is a vector of size n^2, representing an equation in the unknowns t_{ij}.

The total number of rows in B is 2^{2n-1}

B. Initial Brute Force Approach

For small n(such as $n = 2$) the number of combinations is manageable:

- $n = 2$: $2^3 = 8$ rows in B.
- Need to select $n^2 = 4$ rows.
- Total combinations: $\binom{8}{4} = 70$.

All combinations were tested:

- Check if selected matrix A has $\det(A) \neq 0$.
- Solve $At = b$ to find unique solutions.

Result for $n = 2$: 16 distinct solutions were found.

C. Challenges with Increasing n

For larger n, combinatorial growth made brute-force infeasible:

- $n = 3$: $2^5 = 32$ rows.
- Select $n^2 = 9$ rows.
- $\binom{32}{9} \approx 2.8 \times 10^7$ combinations.

While feasible, the computation was becoming time-intensive (more than 7 minutes).

- $n = 4$: $2^7 = 128$ rows, select 16 rows, $\binom{128}{16} \approx 9.33 \times 10^{19}$ combinations.

Clearly, a full enumeration was infeasible.

D. Dependency Analysis and Optimization

To optimize, we analyzed the structure of B and found important properties. Key observations:

- **Negation Property**: First half of B is negative of the second half. Thus, a set and its negation cannot be chosen together in forming A.
- **Group Dependencies**: Rows grouped in blocks (e.g., (0,1,2,3), (4,5,6,7)) were linearly dependent. Choosing all rows from one such group leads to rank-deficiency.
- **Masking and Filtering:** We generated masks (binary vectors indicating selected rows) that respected the dependency constraints. Masks leading to dependent row sets were filtered out early.

This dramatically reduced the number of combinations that needed to be checked, making $n = 3$ feasible in around 40–60 seconds using optimized C++ code.

E. Parallel Computation

To speed up computation:

- Multi-threading with OpenMP and C++ standard threads was implemented.
- Different masks were processed in parallel thus dividing the search space among processors.

This allowed simultaneous exploration of different masks, leading to significant speedup.

F. Strategy Shift for $n = 4$

For $n = 4$, even optimized masking was insufficient. So new strategy was adopted:

- Select 8 random vectors from X.
- Select pairs of independent vectors from Y (28 pairs possible for $n = 4$).
- Form rows by multiplying x vectors with y vectors.
- Construct A with these 16 rows.
- Solve $At = b$ if $\det(A) \neq 0$.

Instead of generating all possible masks, random subsets of X and Y were chosen.

G. Relation to Hadamard Matrices

Parallel to computational work, we explored theoretical connections:

- The Structure of B and its orthogonality suggested links to Hadamard matrices. item We studied properties such as orthogonal rows
- Provided intuition for dependency analysis.

Though a complete Hadamard structure was not established, insights from Hadamard theory helped guide the dependency filtering methods.

IV. Results

- For $n = 2$:
  - Full enumeration completed.
  - 16 distinct solutions found.
- For $n = 3$
  - Dependency filtering + parallel computation.
  - 90 distinct solutions found efficiently.
- For $n = 4$:
  - Randomized sampling approach.
  - Several distinct solutions found.

V. Observations

VI. Contributions

- **Development of a Matrix Computation Framework:** Built a matrix computation library in C++ (matrix.hpp, matrix.cpp) to handle operations like multiplication, determinant calculation, and linear system solving which can be generalised to all types of matrices.
- **Design of Dependency-Based Optimization Techniques:** Identified and exploited row-dependency patterns within the structured matrix B, including symmetry (negation relationships) and blockwise dependencies. Designed pruning techniques to significantly reduce the search space.
- **Algorithm for Efficient Solution Enumeration:** Proposed and implemented algorithms that avoid exhaustive checking by smartly selecting independent sets of rows, thus making the solution feasible even for $n = 3$ and scalable approaches possible for $n = 4$.

n	No. of variables in T	Vectors in X	Vectors in Y	Rows in B	No. of combinations
2	4	4	2	8	70
3	9	8	4	32	2.8e7
4	16	16	8	128	9e19

TABLE I
SUMMARY OF OBSERVATIONS FOR DIFFERENT VALUES OF n

- **Application of Parallel Computing:** Leveraged parallel programming techniques to accelerate computation, particularly in mask generation, dependency checking, and system solving phases.
- **Experimental Discovery of Distinct Solutions:** Successfully found all distinct solutions for $n = 2$ and $n = 3$ and partial solutions for $n = 4$ through computational experiments.
- **Exploration of Theoretical Connections:** Investigated potential links between the problem structure and Hadamard matrices, suggesting a deeper combinatorial structure behind the observed behavior.
- **Identification of Future Research Directions:** Documented open questions, especially regarding the structure of dependencies in B, and suggested algorithmic and theoretical paths for future exploration.

VII. Learning Outcomes

During the course of this project, several important technical, mathematical, and practical skills were developed:

- **Advanced C++ Programming:** Built a custom *matrix.hpp* and *matrix.cpp* library for matrix operations (multiplication, determinant calculation, row operations), allowing complete control and optimization of the code.
- **Algorithm Design and Optimization:** Designed efficient algorithms to handle combinatorially large search spaces. Learned how to exploit problem structure (symmetries, negations, row dependencies) to prune unnecessary computations.
- **Parallel Computing Skills:** Implemented parallelism using multi-threading techniques (e.g., OpenMP, std::thread) to speed up the enumeration.
- **Linear Algebra Mastery:** Deepened understanding of concepts like matrix rank, linear independence, determinants, and solution uniqueness in the context of binary matrices.
- **Combinatorial Reasoning:** Applied combinatorial counting, masking strategies, and dependency grouping to manage large numbers of possible selections.
- **Theoretical Connections:** Investigated links to Hadamard matrices and orthogonal designs, enriching the mathematical background behind structured matrices and binary systems.
- **Problem-Specific Insights:** Realized the special patterns in structured matrices (like negation symmetry and block dependencies) and how they impact solution strategies.
- **Scientific Research Mindset:** Learned how to iteratively refine the approach based on experimental bottlenecks, how to pivot strategies when initial ideas became impractical, and how to explore alternative mathematical methods.
- **Handling Large-Scale Computations:** Understood the challenges of scaling algorithms to very large problems like $n = 4$and developed practical strategies like random sampling instead of exhaustive search.

VIII. Conclusion

This project successfully addressed the problem of analyzing unique solutions in binary linear systems with structured matrices.

Starting from a complete mathematical formulation, we built a computational framework from scratch, combining techniques from linear algebra, combinatorics, and algorithmic optimization.Through deep analysis of the structured matrix B, we identified important symmetries and dependencies that allowed us to reduce the computational complexity significantly.
For small instances ($n = 2, 3$) , we were able to enumerate all or nearly all possible systems and find all distinct unique solutions.
For larger instances ($n = 4$), we adapted by moving from exhaustive enumeration to intelligent sampling, demonstrating the flexibility of our methodology.
The connection to Hadamard matrices, although still under exploration, opens a promising direction for theoretical advancement.

Overall, the project demonstrated the effectiveness of combining rigorous mathematical modeling with efficient computational techniques. The experience gained in matrix theory, high-performance computing, and scientific problem-solving provides a strong foundation for tackling even more complex structured linear systems in future work.
Further extensions could involve:

- Developing probabilistic models for estimating the number of distinct solutions.
- Formal classification of dependency structures in B.
- Exploring more sophisticated random search techniques (e.g., genetic algorithms, Monte Carlo methods).
- Deepening the connection to known combinatorial designs such as Hadamard or Conference matrices.

Acknowledgment

I would like to express my sincere gratitude to *Prof. Arpita Mal* for their valuable guidance, insightful feedback, and continuous support throughout the course of this project. Their expertise in linear algebra and Matrix Theory was instrumental in shaping the direction of this work.

A heartfelt thanks to my parents and my friends for their constant encouragement. Their support kept me grounded — and occasionally well-fed — through all-nighters and debugging marathons.

Supporting Documents

All source codes and sample outputs related to this project are available in the Github Repository:
Github Repo

All other supplementary materials related to this project are available in the shared Google Drive folder below:
Shared Google Drive