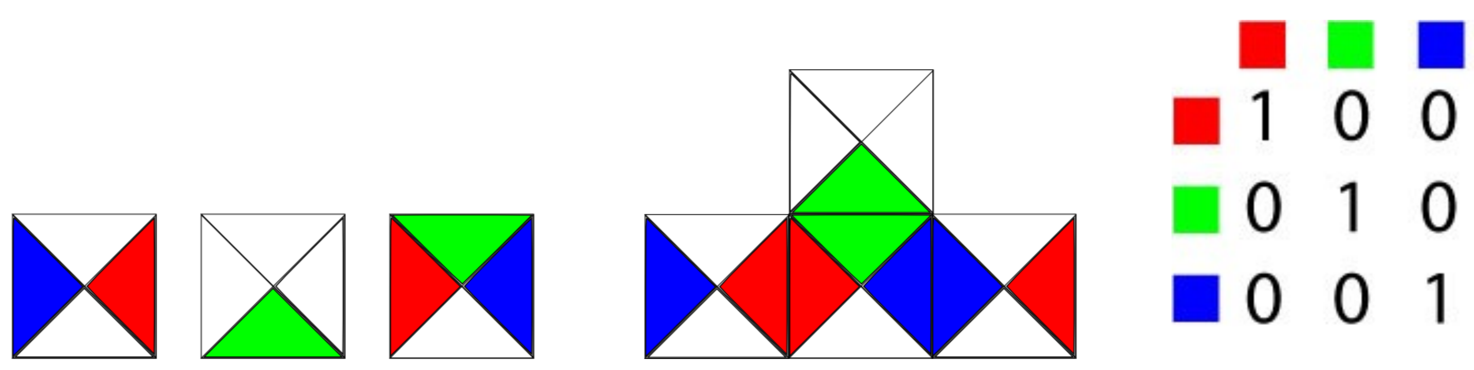
Vector Generation for Self-Assembly Compatibility Matrices

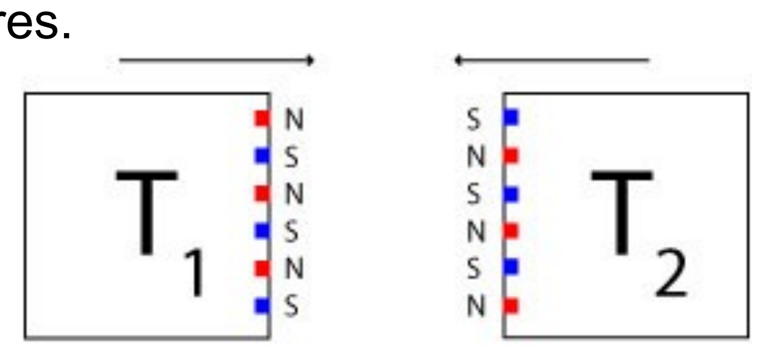
Raul Peña, Dr. Robert Schweller
Department of Computer Science, The University of Texas-Pan American
Poster [1.8]

Introduction

DNA self-assembly uses techniques such as synthesizing DNA strands with specific base sequences to be used as glues to then have tiles attach to one another. Glues on the sides of tiles are labeled and attachment rules between the tile sides are governed by a compatibility matrix giving the strength with which tiles attach to each other.



There is a new method for attachment of tiles using polar north and south magnetic charges in the form of an array on tile edges to either attract or repel. Using a combination of compatible arrays for multiple tiles we can control the pattern of attachments to create larger self assembling structures.



The system for matching tile edges is simple. Each input matrix is a representation of the behavior you want for either North/South or East/West edges of tiles. The positions in the input matrix will designate whether two tile edges are compatible. A 1 represents a positive attraction force between tiles, a 0 represents a balanced magnetic force, and –1 represents a negative repulsion force between tiles.

Using this system for matching edges, we'll need a method to generate the vectors that will represent the possible edges of all tiles. Not only do these magnet vectors have to be unique, but they must constrain to the rules given by an input compatibility matrix.

Example:

$$\begin{array}{c c} & \begin{array}{cccc} e & f & g & h \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix} \end{array}$$

$$a \,.\, e = NxS + 0 + 0 + 0 = 1$$
$$a \,.\, f = NxS + Nx0 + 0 + 0 = 1$$
$$a \,.\, g = NxS + 0 + 0xN + 0 = 1$$
$$a \,.\, h = Nx0 + 0xS + 0 + 0xS = 1$$

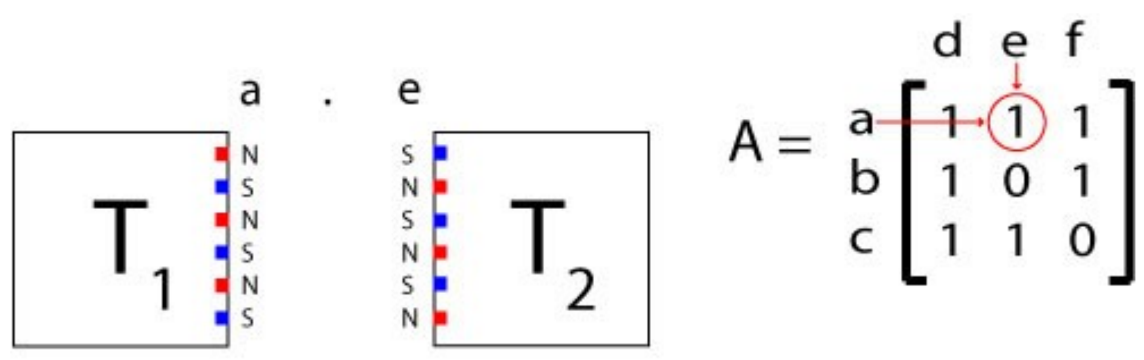
Output vector set

$$\begin{array}{cccccccc} a & b & c & d & e & f & g & h \\ \begin{bmatrix} N \\ - \\ - \\ - \end{bmatrix} & \begin{bmatrix} N \\ N \\ - \\ - \end{bmatrix} & \begin{bmatrix} N \\ - \\ N \\ - \end{bmatrix} & \begin{bmatrix} - \\ S \\ N \\ N \end{bmatrix} & \begin{bmatrix} S \\ - \\ - \\ - \end{bmatrix} & \begin{bmatrix} S \\ N \\ - \\ - \end{bmatrix} & \begin{bmatrix} S \\ - \\ N \\ - \end{bmatrix} & \begin{bmatrix} - \\ S \\ - \\ S \end{bmatrix} \end{array}$$

Results

Not knowing the exact extent of the technology being used to create and use the magnetic charges on self-assembly tiles, we have to compensate with different assumptions of what we can do. This leads to different formulations of the vector generation problem, each with different freedoms or restrictions.

The original formulation involves more freedom and simplicity. You are given an input matrix A with possible values ϵ {-1, 0, 1}, and must output vectors comprised of values ϵ {N, S, null}. The goal being to have the dotproduct of two vectors, each representing an array of magnetic polarities that are on opposite sides of two distinct tiles, equal to a positive, negative, or zero value according to the corresponding position in the input n x n matrix A.



Generating the output vectors for this formulation of the problem is possible, and is achieved by an algorithm that outputs vectors of size Ω(rank(A)) and O(n), where rank(A) is the number of linearly independent rows or columns in A and n is the size of the input matrix.

References

Bin Fu, Matthew J. Patitz, Robert T. Schweller, Bobby Sheline, "Self-Assembly with Geometric Tiles", arxiv.org, arXiv:1104.2809

Open Problem

A different formulation of the problem allows for specification of any integer values in the input matrix, meaning the exact force of attraction or repulsion of two tiles is given by the position in the matrix. This is different from the previous problem in which a 1 in the input matrix represented any positive attraction force, and –1 any repulsing force. This alone would make the problem as simple as the previous one with the same output vector complexity. It is made more interesting with a restriction of the values allowed for the magnet vectors, specifically the omission of the null magnet. The only values that can now be used are either North(N) or South(S). This increases the complexity of the problem, with a lower bound of Ω(n+s), where n is the size of the input matrix, and s is the largest absolute value in the input matrix, but only if the input matrix contains either all odd or all even integers. A solution is impossible if the input matrix contains a mix of both odd and even numbers. An efficient algorithm to generate the vectors for this problem would constitute a good upper bound, but no solution has been proven yet.

Conclusion and future work

The work on these problems has lead to a better understanding of how we can work with a magnetic self-assembly model. The goal of generating efficient vectors to satisfy requirements needed for constructions will be a central part of making a magnetic model useful. Other interesting formulations of the problems mentioned for generating compatible vectors are also being considered, and a solution to the proposed open problem is currently being worked on.