Magnet Matrix Problem

1 Introduction

Having the ability to use arrays of magnets with different polarities to aid in the attachment of self-assembly tiles, we look at the problem of coming up with the sets of arrays/vectors needed for opposite North/South or East/West sides of tiles. Given a compatibility matrix A of size $n \times m$, output a set of n vectors N and a set of m vectors M such that $N_i \cdot M_j = A_{i,j}$, where N_i and $M_j \in P = \{N(\text{north}), S(\text{south}), \text{null}\}$; further, we define the multiplication of such values within the dot product to follow the rules $(null)\times$N$= 0$, $(null)\times$S$= 0$, N$\times$N$= -1$, S$\times$S$= -1$, and N$\times$S$= 1$.

1.1 Solution

For this problem we define the input matrix A to consist of values $\in \{0, 1, -1\}$ with 1 referring to any positive attractive magnetic strength and -1 any negative or repulsive magnetic strength, further k_A is defined as the smallest possible vector size for N and M with input matrix A.

Theorem 1. *Lower bound of $k_A = \Omega(n + m)$ for almost all A*

Proof. ...

Theorem 2. *For matrix A, $k_A = O(n + m)$*

Proof. Algorithm to achieve $O(n + m)$ for input matrix A

Algorithm()
Let $r_{vectors} = array()$ // output vectors for rows
Let $c_{vectors} = array()$ // output vectors for columns
for i from 1 to $numrows(A)$
    $r_{vectors}[i]$ ={};
    for j from 1 to $numcols(A)$
        $c_{vectors}[j]$ = ;
        if(dotproduct($r_{vectors}[i]$, $c_{vectors}[j]$) == $A_(i, j)$)
            // do nothing
        else if(dotproduct($r_{vectors}[i]$, $c_{vectors}[j]$) ¡ $A_(i, j)$)
            addOne($r_{vectors}[i]$, $c_{vectors}[j]$); // add one to result of dotproduct
        else
            substractOne($r_{vectors}[i]$, $c_{vectors}[j]$); // subtract one
            dotproduct
addOne(v_1, v_2)

$v_1[] =$N; // add North value to end of vector v_1
$v_2[] =$S; //
substractOne(v_1, v_2)
$v_1[] =$N; //
$v_2[] =$N; //

1.2 Restriction of not using null

The problem becomes more difficult when considering the omission of the null magnet in the set of vector values P. We then define P' to be the new set consisting of only {N/north and S/south}. Without the power of the null value to make parts of the dot product zero, the method in theorem 2 is no longer usable.

Theorem 3. *Given a matrix A of size $n \times m$, there exists a set of n vectors N and a set of m vectors M, such that $N_i \cdot M_j = A_{i,j}$, where N_i and $M_j \in P'$.*

Proof. Notes:
The input matrix A must consist of either all even or all odd integers(see Theorem 7). The size of vectors in N and M is even if matrix consists of even values and is odd if matrix consists of odd values.

Theorem 4. *Upper bound: $k_A = O(?)$. Possibly $O(n^2)$*

Theorem 5. *Lower bound: $k_A = \Omega(?)$. Possibly $\Omega(n)$*

1.3 Arbitrary matrix values

Up to now we considered the goal of outputting vectors to produce a positive, null, or negative magnetic force given by a position in A. Consider a new problem using an $n \times n$ input matrix A' defined to consist of any values $\in \mathbb{Z}$. The result needed from the dotproduct of two vectors changes to specific integer values, instead of the previously flexible result of allowing any value of positive or negative magnetic forces.

Theorem 6. *There is a lower bound of $\Omega(n + c)$ for an input matrix A', where c is defined as* $\max(A')$

Proof. Assuming lower bounds $\Omega(n)$ and $\Omega(c)$

$$\max(n, c) \geq \frac{(n + c)}{2} = \Omega(n + c)$$

Theorem 7. *Given a matrix A' of size $n \times n$, no solution exists to this problem if A' contains at least one odd integer and at least 1 even integer.*

Proof. If a matrix has both an even and odd integer, then there must be at least one row or column containing both an even and odd integer. We can consider row a from A to contain both odd and even integers, with column b containing the even integer and column d containing the odd integer. We now define a' to be the vector for row a and b' to be the vector for column b. Further, both vectors a' and b' can be split into two subvectors, one of which they share and another which has opposite values to that in the other vector. These subvectors have values $\in P'$. We'll call these sets of subvectors x for the common set, and r for the opposite set. Thus we have an even number $2q = a' \cdot b' = a'_x \cdot b'_x + a'_r \cdot b'_r$ and an odd number $2s + 1 = a' \cdot d' = a'_x \cdot d'_x + a'_r \cdot d'_r$. We can separate the different variables to get $a' \cdot b' - a'_x \cdot b'_x = a'_r \cdot b'_r$ and $a' \cdot d' - a'_x \cdot b'_x = a'_r \cdot d'_r$. From this we still have $a'_r \cdot b'_r$ and $a'_r \cdot d'_r$ being different integers with different parity(one being even and the other odd). Now we are only looking at the r subvectors of a', b', and d', and we know b'_r and d'_r are vectors with opposite values, thus the dot products $a'_r \cdot b'_r$ and $a'_r \cdot d'_r$ produce opposite values which have the same parity. This contradicts our assumption that $a'_r \cdot b'_r$ and $a'_r \cdot d'_r$ have different parity and also that $2r$ and $2s + 1$ have different parity, thus proving that there is no solution for a matrix with both odd and even numbers.

Theorem 8. *$k_A = O(?)$. Can achieve $O(nc)$?*

———'

More problems?