STRATEGY AND COMPLEXITY OF THE GAME OF SQUARES

FLORIAN BREUER AND JOHN MICHAEL ROBSON

Abstract

We introduce a game called Squares where the single player is presented with a pattern of black and white squares and has to reduce the pattern to white by making as few moves as possible. We present a method for solving the game and show that the problem

Problem 1 (Squares Solvability). Given a pattern \( X \) and \( k \in \mathbb{N} \) can \( X \) be solved in \( k \) or less moves?

is NP-complete. We demonstrate a reduction to this problem from Not_All_Equal_3Sat. We also present another NP-complete problem that Squares Solvability can be reduced to.

1. Basic definitions

We start with some formal definitions

Definition 1. A Game Board is a \( m \times n \) board of squares, each either black or white. It has the topology of a flat torus, a property also known as wrap-around.

Definition 2. A cursor is a \( 2 \times 2 \) selection box that can be positioned anywhere on the board.

Definition 3. A move is the act of inverting the colours of the squares inside the cursor. (i.e. black squares become white and white squares become black). A wrap-around move is a move that’s done on the edge of the board, such that part of the cursor is on the one edge of the board and another part on the opposite edge.

2. The Aim of the Game

The game originated as a computer game where the player is presented with a pattern of black and white squares on the board and must solve the board, i.e. reduce the board to white using moves. Note that he may position the cursor anywhere on the board, a move is counted only when he inverts the squares inside the cursor.

The computer also calculates a par, that is a good estimate of the least number of moves required. The idea is, of course, to solve the board on (or possibly even below) par.

We will use the terms “board” and “pattern” interchangeably.

1991 Mathematics Subject Classification 68Q25.
3. Analyzing the Game

Two questions arise here:
(i) How does the computer determine the pattern? Is it random?
(ii) Can all patterns be solved?

Before we answer those questions we note that the probability of a random board being solvable is $2^{-\left(m+n-1\right)}$. That means that if you create one 40×40 board at random each second then you would have to wait an average time of $1.9 \times 10^{16}$ years before you can expect to have found a solvable one. That's about a million times longer than the estimated age of the universe. We shall prove this result later.

3.1 How to Solve a Board

We start by defining parity:

DEFINITION 4. A board is said to have even parity if the number of black squares is even in every row and in every column. Otherwise it has odd parity.

THEOREM 1. A pattern is solvable if and only if it has even parity.

We first need the following lemma:

LEMMA. A move does not alter the parity of the board.

Proof of Lemma. We need only consider the effect on the parity of the two rows and two columns in which the cursor lies. A little thought will convince us that the parity of those rows and columns remains unchanged.

Proof of Theorem 1. Part 1:

The white (empty) board has even parity and moves don’t change the parity, so if a board was reduced to white using moves then it must have started out with even parity. This proves the only if case.

To prove that any board with even parity can be solved we present an algorithm that will always solve a board that has even parity:

Part 2, the Corner Algorithm:

Start by placing the cursor in the top left corner of the board. If the top left square in the cursor is black then we make a move there, otherwise we don’t. We then move the cursor one position to the right and repeat the process: if the top left square in the cursor is black then we make a move there, otherwise we don’t. We continue this process until the cursor has reached the top right corner of the board. By now all the squares in the first row to the left of the cursor are white. But the board has even parity, that means that the number of black squares in the first row must be even. So the last two squares in the first row are either both black (in which case we make the move) or both white (in which case we don’t). So that way we have turned the entire first row white. We repeat the process with the other rows until all but the last two rows have been turned white. Now we note that, because of even parity, the number of black squares in every column must be even. That means that the last two rows are identical and so as we now turn the second last row white (the usual way) the last row will empty along with it.
So we have seen that the Corner Algorithm will solve any board that has even parity. This concludes the second part of our proof for Theorem 1. □

3.2 Further Theorems  We must now note that the colour of each square is only dependent on its starting colour and the number of times it was inverted, so it follows that the chronological order in which the moves are made is not important. We also see that making the same move twice has the same effect as not making it at all, so it would be pointless to make any move more than once.

We also see that the corner algorithm did not make any wrap-around moves. That suggests that we might find better solutions that do use wrap-around. Before we can look into that any further we need two more definitions:

**Definition 5.** Two solutions are only considered different if they actually contain different moves.

**Definition 6.** A solution that does not use any wrap-around moves is called a Limited Solution.

We shall now prove an important theorem about the Corner Algorithm:

**Theorem 2.** The Corner Algorithm finds the unique limited solution to a board, provided the board has even parity.

**Proof of Theorem 2.** To find a limited solution we are not allowed to make any wrap-around moves. Consider the square in the top left corner of the board. The only move that can affect it is the first move of the Corner Algorithm, since all the others would be wrap-around moves. So the colour of this square uniquely determines the move in the top left corner: if that square is black then we must make the move, otherwise we cannot. When we move on to the next square we find that there are only two non-wrap-around moves that can affect it: the one we just made and the next move of the Corner Algorithm, so the move in the next cursor position is again uniquely determined by the board. That way we can see that all the moves in the first row are determined by the board. When we consider the moves in the next row, we again find that there is only one move left that can affect a given square: all other moves are either wrap-around moves (and thus not valid) or have already been determined.

So we find that in every cursor position the move (or lack of one) is forced by the board, and thus the entire solution is uniquely determined. Of course it does not matter, ultimately, in which order these moves are made, but the positions of the moves will be the same for any limited solution of a given board. Theorem 2 is thus proved. □

Now that we know how to find one solution, how do we find the best solution (that is, a solution that uses the least number of moves)?

**Definition 7.** A Solution Board is a $m \times n$ board of black and white squares, where each square corresponds to one cursor position on the game board, such that
the square has the same co-ordinate as the top left corner of the corresponding cursor position.

Each solution board represents a set of moves as follows: a square is black if and only if the corresponding move is made.

It follows that the number of black squares in a solution board is equal to the number of moves made in that solution. We note that a solution board represents a limited solution if and only if the bottom row and rightmost column are empty (white).

**Definition 8.** Two solution boards are said to be equivalent if they represent solutions to the same pattern. We denote a solution board to a pattern $X$ by $[X]$, and the limited solution board by $[X]^*$. 

Every solution board $[X]$ must have the property that if we apply the moves of $[X]$ to the empty board then we must obtain $X$. This follows from the fact that if we apply $[X]$ to $X$ twice then we are left with $X$ again, but if we apply it only once then we get the empty board (by definition), so applying $[X]$ a second time (to the empty board) must leave us with $X$.

This implies that $X$ is uniquely determined by $[X]$. As $[X]^*$ is uniquely determined by $X$ (Theorem 2) we have thus established a one-to-one correspondence of solvable patterns $X$ and their limited solution boards $[X]^*$.

As every $[X]^*$ has $(m-1)(n-1)$ squares which can each either be black or white, and $m+n-1$ squares which must be white it follows that $2^{(m-1)(n-1)}$ boards are solvable and the probability that a random board is solvable is $2^{-(m+n-1)}$.

### 3.3 Finding the Best Solution

We now return to the question of finding the best solution to a given board.

**Definition 9.** A row (column) flip is the act of inverting all the squares in one row (column) of a solution board. Naturally this includes any wrap-around moves in that row (column).

**Theorem 3.** Let $A$ and $B$ be solution boards such that $B$ is obtained from $A$ by a single row (or column) flip. Then $A$ and $B$ are equivalent.

**Proof of Theorem 3.** Let $B$ be identical to $A$ except for row $i$ which has been inverted. All we need to show is that both $A$ and $B$ produce the same pattern when applied to the white board. This is the same as showing that applying both $A$ and $B$ to the white board produces the white board again. If we do this then obviously any move not in row $i$ will be done twice and thus have no effect. In row $i$, however, every possible move will be made exactly once, and it can be seen that every square on the game board that is affected by the moves of row $i$ will be inverted exactly twice, and thus stay white, too. A similar argument follows for the $i$th column.

Thus Theorem 3 is proved. □

**Theorem 4.** Let $[X]$ and $[X]'$ be equivalent solution boards. Then $[X]'$ can be obtained from $[X]$ by a sequence of row and column flips.
Strategy and Complexity of the Game of Squares

Proof of Theorem 4. We shall show that any solution board $[X]$ can be reduced to the equivalent limited solution board $[X]^*$ by a sequence of row and column flips. The theorem then follows because we can reduce $[X]$ to $[X]^*$ and then apply the same combination of flips that would reduce $[X]'$ to $[X]^*$, thus obtaining $[X]'$ from $[X]^*$.

To reduce $[X]$ to $[X]^*$ we must first empty the rightmost column of $[X]$. We do this by taking each row and flipping it if necessary (thus inverting all the black squares in the rightmost column). Now that the rightmost column is empty we repeat the process to empty the bottom row, by flipping the necessary columns. We are left with a solution board that is still equivalent to $[X]$ (by Theorem 3) but has its rightmost column and bottom row empty, which means that it is the limited solution board $[X]^*$. This concludes our proof for Theorem 4.

We have thus shown that we can find the best solution to any solvable board by first finding the limited solution using the Corner Algorithm and then minimizing the number of black squares in the solution board with successive row and column flips. So we see that the problem

Problem 2. Solve a given board using the least number of moves

has been reduced to the problem

Problem 3. Given an $m \times n$ board of black and white squares, reduce the number of black squares to a minimum with a sequence of row and column flips.

We note that the reduction can be done in time polynomial in $\max(m, n)$. Problem 3 corresponds to a decision problem as follows:

Problem 4 (Row Flipping). Given an $m \times n$ board of black and white squares and a natural number $t$, does there exist a sequence of row and column flips that reduce the board to contain $t$ or less black squares?

We shall show that this problem (and thus Problem 1, mentioned in the abstract) is NP-complete.

4. The Complexity of Row Flipping

We will show that the problem of Row Flipping (given an initial board and a number $k$, does some set of flips produce at least $k$ White squares (where $k = mn - t$ in problem 4 above)) is NP-complete.

4.1 “Grey Squares” As a preliminary we show how, given any positive $\epsilon$, we can construct, in time polynomial in $\epsilon^{-1}$, a board such that no set of row and column flips can make the density of White squares greater than $1/2 + \epsilon$.

Choose $s$ as the smallest power of 2 greater than or equal to $\epsilon^{-2}$. The board will have size $s$ by $s$.

In row $j$ of the square, column $i$ will be White if and only if the number of 1-bits
in the bitwise and of the binary representations of \( i \) and \( j \) is even. (The square is simply a listing of the \( s \) words of a Walsh code of length \( s \) with the bits coded as colours [1].)

Suppose that some set of row and column flips produces more than \( s^2(1/2 + \epsilon) \) White squares. We will show that this leads to a contradiction.

We consider the number \( T \) of ordered triples \((c, i, j)\) such that in column \( c \) squares \( i \) and \( j \) have the same colour (note that \( i \) and \( j \) are not necessarily distinct and that \((c, i, j)\) and \((c, j, i)\) are both counted). For any pair of distinct rows \( i \) and \( j \), there are initially exactly \( s/2 \) columns \( c \) such that squares \([c, i]\) and \([c, j]\) differ in colour. This is not altered by row flips in rows other than \( i \) and \( j \) (clearly), by row flips in rows \( i \) or \( j \) (which leave squares \([c, i]\) and \([c, j]\) different if and only if they were previously the same) or by column flips (which do not change the equality/inequality of squares \([c, i]\) and \([c, j]\)). Hence adding over all \( s(s-1) \) pairs \( i \neq j \) and adding in \( s^2 \) for the triples with \( i = j \), we always have

\[
T = s^2(s-1)/2 + s^2 = s^2(s+1)/2 \quad (4.1)
\]

The quantity \( T \) is minimized for a given number of White squares if the White squares are evenly spread over the columns. Hence, by our supposition on the number of White squares, \( T \) is at least what it would be if each of the \( s \) columns contained exactly \( s/2 + s\epsilon \) White squares, namely

\[
T \geq s(s^2/2 + 2s^2\epsilon^2) \quad (4.2)
\]

(4.1) and (4.2) imply that \( s\epsilon^2 \leq 1/4 \) which conflicts with the choice of \( s \) as greater than or equal to \( \epsilon^{-2} \). This contradiction proves that the square described has the stated property. Since the size of the square is \( O(\epsilon^{-4}) \) it can be constructed in time polynomial in \( \epsilon^{-1} \).

We have proved that this square will always have density of White squares less than or equal to \( 1/2 + \epsilon \). Similarly the same bound applies to the Black squares.

### 4.2 Row_Flipping and Black_White_Grey_Row_Flipping

We will define a generalization of Row_Flipping easily shown NP-complete. Then we will show a reduction from this new problem to Row_Flipping, thereby establishing also the NP-completeness of the original Row_Flipping problem.

**Definition 10.** Black_White_Grey_Row_Flipping is a game like Row_Flipping except that there are three colours of squares, Black, White and Grey. Flipping a row or column changes Black squares to White and vice versa as before but leaves Grey squares unchanged. The aim is still to maximize the number of White squares.

### 4.3 Proof of NP-Completeness for Black_White_Grey_Row_Flipping

By a minor abuse of terminology we use the same name for the game and the problem of deciding, given an initial configuration and an integer \( k \), whether the number of White squares can be made at least \( k \).

**Theorem 5.** Black_White_Grey_Row_Flipping is NP-complete.

**Proof of Theorem 5.** It is clear that the problem is in NP. We will establish its NP-hardness by a reduction from Not_All_Equal_3SAT ([2] problem LO3). This
problem consists of a set $U$ of variables and a collection $C$ of clauses over $U$ such that each clause $c \in C$ has $|c| = 3$. The question is: is there a truth assignment for $U$ such that each clause in $C$ has at least one true literal and at least one false literal?

We prefer to think of the condition on the assignment as the existence of $2|C|$ triples $(l_1, l_2, c)$ such that $l_1$ and $l_2$ are two literals both occurring in the clause $c$ and taking different values under the assignment. To see the equivalence of this to the `standard' condition, note that any clause which has at least one true literal and at least one false literal produces two such pairs whereas any other clause produces none.

The Reduction  Given an instance of Not_All_Equal_3SAT with $|U| = m$ and $|C| = n$, we construct an instance of Black_White_Grey_Row_Flipping with $m$ rows and $3n$ columns as follows:

- each row corresponds to a distinct variable
- each column corresponds to a triple $(v_1, v_2, c)$ such that $v_1$ and $v_2$ are two distinct variables both occurring in the clause $c$ (the order of $v_1$ and $v_2$ being immaterial)
- in the column $(v_1, v_2, c)$
  - the squares in rows $v_1$ and $v_2$ are both White if $v_1$ and $v_2$ occur with different signs in $c$ or Black and White otherwise
  - all other squares are Grey
- the target value $k$ is $5n$.

Proof  Any set of flips is equivalent to a set of row flips in which no row is flipped more than once followed by a set of column flips. We consider any such sequence of flips. The set of row flips defines a truth assignment $A$ to the variables $U$, namely the one which assigns True to a variable if and only if its row is flipped.

Initially the two non-Grey squares in a column are the same colour if and only if the assignment of False to all variables makes the two corresponding literals differ. Hence after the set of row flips, they are the same colour if and only if the assignment $A$ makes the two literals differ. Clearly the optimal set of column flips is one which gives two White squares in any column which has two non-Grey squares the same colour; moreover any set of column flips gives exactly one White square in all other columns. Hence the optimal set of column flips gives a total number of White squares of $5$ times the number of clauses with literals of differing values plus $3$ times the number of clauses with three equal literals. Thus it is possible to produce $5n$ White squares if and only if there is an assignment $A$ satisfying the condition.

This completes the proof of the NP-completeness of Black_White_Grey_Row_Flipping.

4.4 Reduction Black_White_Grey_Row_Flipping to Row_Flipping.  Given an instance $I$ of Black_White_Grey_Row_Flipping with $m$ rows and $n$ columns, we construct an instance $I'$ of Row_Flipping as follows:

- Choose $\epsilon = 1/2mn$
- Choose $s$ as described in section 4.1 for this $\epsilon$
Allocate a square of size $s$ by $s$ for each square of $I$. If the square of $I$ is White or Black all the squares of the large square are this colour; otherwise the large square is a copy of the square described in section 4.1 for this $\epsilon$.

The target value is $s^2k + s^2(G-1)/2$, where $G$ is the number of grey squares in $I$.

Clearly a set of flips obtaining the target $k$ for $I$ simply defines a set of flips making $k$ uniform white squares in $I'$. Since the non-uniform squares of $I'$ can not have an excess of Black squares as great as $s^2/2$ (since there are at most $mn$ of them each with an excess less than $s^2/2mn$), this achieves the target for $I'$.

Conversely, consider a set of flips achieving the target for $I'$. Since less than $s^2/2 \times (G+1)$ White squares can have come from the non-uniform large squares, more than $s^2(k-1)$ must have come from the uniform ones. Hence at least one of the $s^2$ copies of the White and Black squares of $I$ which are interleaved to form the uniform large squares of $I'$ must have contributed at least $k$. The set of flips applied to the rows and columns of this copy defines a set of flips achieving the target for $I$.

This concludes the proof of the validity of the reduction. Since the problem is clearly in NP we have proved it NP-complete.

5. Another NP-Complete Problem

Row_Flipping can also be interpreted as an integer optimization problem, as follows

**Problem 5** (Integer Optimization). Given a natural number $h$ and an $m \times n$ matrix $A$ that contains only numbers in $\{-1, 1\}$, do there exist vectors $X = (x_1, x_2, \ldots, x_m)$ and $Y = (y_1, y_2, \ldots, y_n)$ with $x_i, y_j \in \{-1, 1\}$ (for $1 \leq i \leq m, 1 \leq j \leq n$) such that

$$S = \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}x_iy_j \geq h? \quad (5.1)$$

**Theorem 6.** Problem 5 is NP-complete.

**Proof of Theorem 6.** We shall reduce Row_Flipping to problem 5 in time polynomial in $\max(m, n)$.

Given an instance of Row_Flipping we reduce it to problem 5 as follows. The board is translated into the matrix $A$ such that

$$A_{ij} = \begin{cases} 1 & \text{if square } [i, j] \text{ is white} \\ -1 & \text{if square } [i, j] \text{ is black} \end{cases}$$

and $h = mn - 2 \times (\text{target number of black squares})$.

The equivalence between solutions to Row_Flipping and vectors $X$ and $Y$ solving the Integer Optimization problem is obtained as follows:

$$x_i = \begin{cases} -1 & \text{if row } i \text{ is flipped} \\ 1 & \text{if row } i \text{ is not flipped} \end{cases}$$

$$y_j = \begin{cases} -1 & \text{if column } j \text{ is flipped} \\ 1 & \text{if column } j \text{ is not flipped} \end{cases}$$
We see that flipping a row in Row_Flipping is equivalent to multiplying that row in $A$ by $-1$ in this problem. So the number of black squares on the board in Row_Flipping is equal to the number of $-1$'s appearing in the sum $S$ (5.1), and the number of white squares is equal to the number of 1's in the sum. So we see that $S = W - B$ (where $W$ is the number of white squares and $B$ the number of black squares on the board) and thus problem 5 is equivalent to the question of whether we can find row and column flips for an instance in Row_Flipping such that $W - B \geq h$. As $W + B = mn$ it follows that problem 5 is just Row_Flipping in disguise. □

6. Conclusion

We have only treated a specialized problem here and many questions still remain:

(i) What happens in higher dimensions? We suspect that the game itself will still be similar, although the proof of NP-completeness might be different.

(ii) What happens if we use more colours (i.e. have $k$ colours and cycle them instead of inverting them)? If $k$ is odd then the game changes completely. We suspect that the game will still be similar for even $k$.

(iii) What if the topology of the board changes?

(iv) What is the best algorithm to solve Row_Flipping? A slightly modified “brute force” algorithm takes at most $\max(m, n) \times 2^{\min(m, n)}$ flips. Can we do better?

Acknowledgements. One of us (F.B.) would like to thank Dr. L.R. le Riche for his kind help and advice.

References
