Blocking Intercalates In Sudoku Erasure Correcting Codes

Daniel J. Williams, Sian K. Jones, Paul A. Roach, *Member IAENG*, and Stephanie Perkins *

Abstract—**A Sudoku grid is a square grid of size n, further subdivided into n mini-grids and into which the symbols $1, 2, \ldots, n$ are placed such that each symbol appears exactly once in each row, column and mini-grid. It has been suggested that Sudoku puzzles may have applications in Coding Theory, specifically in the recovery of erasures [1, 2]. By encoding a message within a Sudoku grid, a received grid containing erasures can be treated as a puzzle, the solution of which will enable the original message to be recovered. However, the received values must be sufficient to specify a unique solution, or the message may not be recovered correctly. Every Sudoku grid possesses a number of *unavoidable sets*, which are patterns of cell values that, if all are absent from a puzzle, prevent unique completion of that puzzle. The presence of the smallest size of unavoidable set, patterns of cells of size 2×2 also known as intercalates, has been shown to prevent the practical application of Sudoku grids for erasure correction [3]. This paper proposes a scheme that employs *meta-data* derived from a Sudoku grid as an efficient means of fixing, or 'blocking', its intercalates through the use of additional symbols, and evaluates the scheme in the context of erasure correction. Programs developed to locate and enumerate the intercalates of a grid, and to relabel cell values, are described, and results are presented on the numbers of additional symbols required and on the improvement in the successful recovery of erased cell values. As an alternative approach to the 'blocking' of intercalates, consideration is also given to a subset of Suduko grids, known as 2-Quasi-Magic Sudoku, which have specific intercalate properties; the suitability of this subset for erasure correction is evaluated.**

Keywords:* *Sudoku, intercalates, unavoidable sets, Coding Theory, erasure correction

1 Introduction

Erasure correction techinques are employed to enable reliable recovery of digital data [4], and have been used in many applications in telecommunications and data storage and retrieval. Some recent contributions to the field include the use of quadratic residue codes [5], and erasure correction in Voice over Internet Protocol (VoIP) to improve speech quality deterioration [6, 7]. Many combinatorial structures can be represented as recreational mathematical puzzles. These puzzles are solved by using existing information to determine missing values, which has great similarity with erasure correction techniques. The automated solution of recreational puzzles has received much recent attention; as examples, Kakuro has been addressed through the use of search optimization [8], and Sudoku puzzles have been solved through the compatibility matrix method [9], and the intelligent pruning of search spaces [10]. The use of Sudoku grids in Coding Theory for encoding and recovering messages has been proposed [1, 2], and examined in the context of constructing an erasure correction scheme that incorporates the Sudoku structure [3]. In this scheme, a message to be transmitted is encoded in a specific subset of cells within a Sudoku grid, and the remaining cells are filled in such a way as to form a valid grid. Transmission over some channel potentially subjects the grid to erasures. The properties of the grid structure (non-repetition of the values in any row, column and mini-grid) enable the recovery of some, or all, erasures, in a manner analogous to the solution of a Sudoku puzzle. Correction of erasures in the cells containing message data is sufficient to recover the original message. The proposed method of encoding information in Sudoku grids of size 9×9 is to locate the message data in the mini-grids along one diagonal. Any of the 9! different arrangements of the values $1, 2, \ldots, 9$ can be placed in each of those three mini-grids, and the remaining grid can always be completed to a valid grid for transmission [3]. However, this leads to two-thirds of the data sent being redundant, and the 9! ways of arranging each of the three minigrids (working over an alphabet of

*The authors thank the Nuffield Foundation whose funding of Daniel J. Williams, the recipient of an Undergraduate Research Bursary, enabled part of the work of this paper to be completed.
Manuscript received January 14, 2011.

D. J. Williams is in Department of Computing and Mathematical Sciences, University of Glamorgan, Pontypridd, CF37 1DL, United Kingdom.

S. K. Jones is with the Combinatorics & Heuristics Group, Department of Computing and Mathematical Sciences, University of Glamorgan, Pontypridd, CF37 1DL, United Kingdom (email: skjones@glam.ac.uk).

P. A. Roach is with the Combinatorics & Heuristics Group, Department of Computing and Mathematical Sciences, University of Glamorgan, Pontypridd, CF37 1DL, United Kingdom (phone: (0)1443 482258; fax: (0)1443 482169; email: proach@glam.ac.uk).

S. Perkins is with the Combinatorics & Heuristics Group, Department of Computing and Mathematical Sciences, University of Glamorgan, Pontypridd, CF37 1DL, United Kingdom (email: sperkins@glam.ac.uk).

9 symbols) results in a rate of the scheme of

$$\frac{\log_9(9!^3)}{81} = 0.21579. \quad (1)$$

Given this low rate, the scheme must ideally operate at erasure probabilities approaching $1 - 0.21579 = 0.78421$ to be competitive with established codes.

The potential of such an approach for the recovery of erasures is resticted by the presence of *unavoidable sets* in those grids (patterns of cell values which, if all are absent from a puzzle, prevent unique completion of that puzzle), notably *intercalates* [3]. Intercalates, the smallest unavoidable sets, are 2×2 subsquares of cell values. The presence, and number, of intercalates is identified as being critical in the effective use of Sudoku grids for Coding Theory. This difficulty can be addressed in two ways: by employing additional redundant data to improve recovery of erasures; or by selecting a subset of Sudoku grids which possess fewer intercalates. The former approach requires a method to 'fix' or 'block' intercalates, which must involve little additional data cost to encode a message in order to limit the resulting reduction of the rate of the scheme. The latter approach requires the classification of subsets of Sudoku grids according to numbers of intercalates, and possibly the identification of additional constraints that may be placed on the arrangement of values in the grid such that the required beneficial intercalate properties occur. The consequences on the rate of the scheme due to the selection of a subset of all Sudoku grids must be investigated.

This paper proposes and evaluates a scheme for 'fixing' or 'blocking' intercalates in Sudoku grids of size 9×9, by locating and re-labelling some cell values that lie in intercalates (increasing the number of symbols in the grid beyond 9); the number of addional symbols and the locations of the re-labelled cells are recorded in *meta-data* to be sent along with the grid. Additionally, a specific approach to the selection of a subset of Sudoku grids is also presented, and its consequences for coding are considered. The subset selected is the set of 2-Quasi-Magic Sudoku grids, which has previously been proposed as possibly more appropriate for an erasure correction scheme [10], and which is defined in Section 4.

Section 2 describes the notation that will be used throughout this paper, and details known results concerning intercalates in Sudoku puzzles. Section 3 describes the methods developed to locate intercalates in a Sudoku grid (Section 3.1) and to select and re-label some cell values within those intercalates (Section 3.2). A greedy algorithm for selection of cells that will lead to efficient re-labelling is detailed in Section 3.2.1. The results of the approach to re-labelling are presented in Section 3.3, and are evaluated in the context of erasure correction. Specifically, results are presented on the numbers of additional symbols required (as a measure of the efficiency of the scheme) and on the improvement in the successful recovery of erased cell values. The intercalate properties of the set of 2-Quasi-Magic Sudoku grids is examined in Section 4, establishing how these properties make them potentially more appropriate for an erasure correction scheme, and their usefulness is evaluated. Concluding comments are offered in Section 5.

2 Notation and Literature Review

Definition 1. *A Sudoku grid, $S^{x,y}$, is a $n \times n$ array subdivided into n mini-grids of size $x \times y$ (where $n = xy$); the values $1, \ldots, n$ are contained within the array in such a way that each value occurs exactly once in every* row, column *and* mini-grid.

$S^{x,y}$ consists of y *bands*, each composed of x horizontally-consecutive mini-grids, and x *stacks*, each composed of y vertically-consecutive mini-grids. Each $x \times y$ mini-grid possesses x sub-rows, or *tiers* and y sub-columns, or *pillars*. (Note that when the mini-grids are square they also possesses two *diagonals*.)

Definition 2. *${S^{x,y}}_{a,b}$ is the mini-grid of $S^{x,y}$ in row a, $1 \leq a \leq y$, and column b, $1 \leq b \leq x$; $[{S^{x,y}}_{a,b}]_{i,j}$ is the cell in tier i, $1 \leq i \leq x$, and pillar j, $1 \leq j \leq y$.*

A Sudoku puzzle is considered to be formed by the removal of values from a number of cells of a completed puzzle. The cell values that remain are referred to as the puzzle *givens*.

Lemma 3. *The removal of the values from at most three cells from a Sudoku grid results in a puzzle with a unique solution.*

Proof. If the values are removed from three non-adjacent cells then for any row, column and mini-grid containing an empty cell, 8 distinct assigned values remain. Hence the empty cell may only be recovered in one way. Otherwise, at least two cells may be solved uniquely and once these two are solved then the remaining cell can only be solved in one way. □

Lemma 4. *The removal of the values from four cells of a completed Sudoku grid does not necessarily result in a puzzle having a unique solution.*

Proof. If the contents of the four cells correspond to only two distinct values, and the four cells are positioned in only two rows, two columns and two mini-grids then these cells may be solved in two ways (by swapping the positions of the two values). □

Definition 5. *An* unavoidable set *is a collection of cells and values such that if the values are removed from all of these cells then the Sudoku puzzle does not have a unique solution.*

A Sudoku puzzle does not have a unique solution if some of the values removed from the cells form an unavoidable set. An intercalate (defined below) is the smallest form of an unavoidable set.

Theorem 6. *If any Sudoku solution possesses, by some permutation, an unavoidable set, then to ensure a unique solution at least one of these cells must be selected as a given in any corresponding puzzle.*

Definition 7. *An* intercalate *in a Latin square is a subsquare of order 2, containing exactly two distinct values.* [11]

Definition 8. *A* Sudoku intercalate *may be formed within a band:*

$$[S^{x,y}{}_{a,b}]_{i,j} = [S^{x,y}{}_{a,d}]_{m,k} \quad \text{and} \quad [S^{x,y}{}_{a,b}]_{m,j} = [S^{x,y}{}_{a,d}]_{i,k}$$

or within a stack:

$$[S^{x,y}{}_{a,b}]_{i,j} = [S^{x,y}{}_{c,b}]_{m,k} \quad \text{and} \quad [S^{x,y}{}_{a,b}]_{i,k} = [S^{x,y}{}_{c,b}]_{m,j}$$

for $b, d, i, m \in \{1, \ldots, x\}$ *with* $b \neq d$ *and* $i \neq m$ *and for* $a, c, j, k \in \{1, \ldots, y\}$ *with* $a \neq c$ *and* $j \neq k$.

The grid of Figure 1 provides an example of an intercalate in a band, formed within the cells that are shown empty. This puzzle must be completed by arranging the missing values, 3 and 9; two valid arrangements are possible, meaning that the puzzle cannot be completed uniquely. (This example grid also possesses other intercalates, including a stack intercalate shown in italics.)

2	3	4	5	1	9	8	6	7
5	9	7	*6*	*8*	2	4	3	1
1	6	8	3	7	4	5	2	9
	7	4	1		6	2	5	8
	8	1	2		5	6	7	4
6	2	5	7	4	8	9	1	3
7	5	9	*8*	*6*	1	3	4	2
4	3	2	9	5	7	1	8	6
8	1	6	4	2	3	7	9	5

Figure 1: A Sudoku puzzle having two solutions

Corollary 9. *From a completed Sudoku grid the removal of the values from any four cells which do not form a Sudoku intercalate result in a puzzle with 77 givens and a unique solution.*

Proof. Follows directly from Lemma 4. □

The numbers of Sudoku grids containing specified numbers of Sudoku intercalates is represented in the logarithmic graph in Figure 2.

Theorem 10. *Fewer than 1% of Sudoku puzzles with 77 givens are not uniquely solvable.*

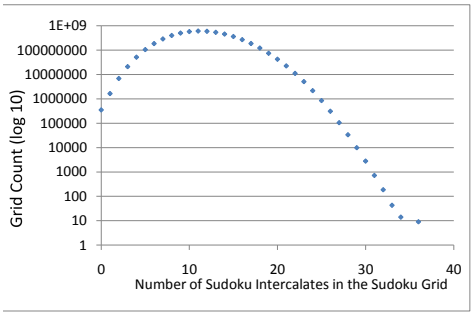

Figure 2: Total numbers of Sudoku grids possessing specified numbers of Sudoku intercalates [12]

Proof. There are $\binom{81}{4}$ ways of removing 4 values from the set of all $6,670,903,752,021,072,936,960$ Sudoku grids [13]. There are $77,241,622,677,889,000,000,000$ ways of forming a Sudoku puzzles with 77 values in which the removed values form a Sudoku intercalate [12] and therefore are not uniquely solvable. (That is the sum of the number of intercalates contained in the grid multiplied by the number of grids with the specific number of Sudoku intercalates.) Thus the percentage number of ways of removing four values and forming a non-uniquely solvable puzzle is

$$\frac{77,241,622,677,889,000,000,000}{\binom{81}{4} \times 6,670,903,752,021,072,936,960} \times 100$$

$$\approx 6.95 \times 10^{-4}\%.$$

□

3 Blocking Intercalates

This section describes the proposed approach to 'fixing', or 'blocking', intercalates. Firstly, a method is required to locate all the intercalates present in a grid (Section 3.1). Knowledge of the locations of all cells within these intercalates will then be used to re-label some of those cell values (Section 3.2). This extends the number of symbols in the 9×9 Sudoku grid beyond 9, and the method of re-labelling seeks to minimise both the total number of additional symbols required and the number of cells that are re-labelled. This minimisation is necessary to ensure that the proposed new scheme for erasure correction employs a minimum of meta-data to be transmitted alongside the useful message data. The combination of re-labelling, and the accompanying meta-data concerning the additional symbols employed and their location, is expected to reduce the effect of intercalates on the unique recovery of grids following erasures.

3.1 Locating Intercalates

Locating all the intercalates of a Sudoku grid involves matching cell values in the grid to patterns such as the one indicated in Figure 3.

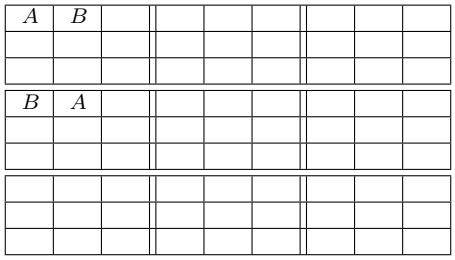

Figure 3: Intercalate cell pattern

Algorithm 1 describes the method employed, which iteratively sets the grid's cell values to A and B, and seeks a matching pair. The algorithm locates all of the stack intercalates, and then reflects the grid in the main diagonal and repeats the process to locate all of the band intercalates.

Algorithm 1 Locating Intercalates

```
for every cell in the Sudoku grid do
  Set First_cell to that cell
  Set A to be the value in First_cell
  for every cell in the tier of First_cell that has a higher
  column number than First_cell do
    Set Second_cell to that cell
    Set B to be the value in Second_cell
    for every cell in the column, but not in the pillar,
    of the First_cell do
      Set Third_cell to that cell
      if Third_cell's value is B then
        if the cell in the same row as Third_cell and
        the same column as Second_cell contains A
        then
          record this intercalate
        end if
      end if
    end for
  end for
end for
REFLECT Sudoku grid along the main diagonal and
repeat algorithm above
```

3.2 Re-labelling Cell Values

Given the locations of all the intercalates of a grid (determined by the method described in Section 3.1), the aim here is to 'block' one cell of each intercalate by re-labelling its value (that is, using a label beyond the values 1 to 9) and to record the value replacement in meta-data to be transmitted with the grid. Each 'blocked' intercalate then consists of four cells and three values. The meta-data will enable the unique recovery of the grid, even following the removal of all the cell values in all of its intercalates. The method is constructed in such a way that a minimum number of additional labels (symbols) and a minimum number of re-labellings are employed.

4	1	3	5	6	2	9	8	7
9	**7**	8	1	4	3	6	5	2
6	5	**2**	8	9	7	**4**	3	**1**
3	4	1	9	2	5	7	6	8
7	**9**	**5**	6	3	8	**1**	2	**4**
2	8	**6**	7	1	4	5	9	3
1	3	9	2	7	6	8	4	5
5	2	**7**	4	**8**	9	3	1	6
8	6	4	3	**5**	1	2	7	9

(a) Original grid, containing five intercalates (bold)

4	1	3	5	6	2	9	8	7	10	11
9	**7**	8	1	4	3	6	5	2	10	11
6	5	**10**	8	9	7	**11**	3	**1**	**2**	**4**
3	4	1	9	2	5	7	6	8	10	11
10	**9**	**5**	6	3	8	**1**	2	**4**	**7**	11
2	8	**6**	7	1	4	5	9	3	10	11
1	3	9	2	7	6	8	4	5	10	11
5	2	**7**	4	**10**	9	3	1	6	**8**	11
8	6	4	3	**5**	1	2	7	9	10	11
7	10	**2**	10	**8**	10	10	10	10		
11	11	11	11	11	11	**4**	11	11		

(b) Augmented grid with four cells 'blocked' (indicated in bold and underlined, with the corresponding meta-data indicated in bold)

4	1	3	5	6	2	9	8	7	10	11
		8	1	4	3	6	5	2	10	11
	5		8	9	7		3		**2**	**4**
3	4	1	9	2	5	7	6	8	10	11
			6	3	8		2		**7**	11
	8		7	1	4	5	9	3	10	11
1	3	9	2	7	6	8	4	5	10	11
	2		4		9	3	1	6	**8**	11
	6	4	3		1	2	7	9	10	11
7	10	**2**	10	**8**	10	10	10	10		
11	11	11	11	11	11	**4**	11	11		

(c) Augmented grid with all 18 intercalate cells removed

4	1	3	5	6	2	9	8	7	10	11
		8	1	4	3	6	5	2	10	11
	5	**10**	8	9	7	**11**	3		**2**	**4**
3	4	1	9	2	5	7	6	8	10	11
10			6	3	8		2		**7**	11
	8		7	1	4	5	9	3	10	11
1	3	9	2	7	6	8	4	5	10	11
	2		4	**10**	9	3	1	6	**8**	11
	6	4	3		1	2	7	9	10	11
7	10	**2**	10	**8**	10	10	10	10		
11	11	11	11	11	11	**4**	11	11		

(d) The recovered 'blocked' cells, from which unique solution is possible

Figure 4: Cell re-labelling: (a) the original grid; (b) five 'blocked' intercalates; (c) intercalated cell data removed; (d) recovered 'blocked' cells.

The method of achieving this 'blocking' is illustrated in Figure 4, and outlined in Algorithm 2. The original Sudoku grid is shown in Figure 4(a) with all five of its intercalates shown in bold. If all 18 of those cells were removed and treated as a puzzle, the puzzle would have multiple possible solutions. In Figure 4(b), the grid has been augmented with two additional rows and columns extending outwards from the grid - the meta-data. The 'blocking of the intercalates has been achieved by replacing each of the underlined values with a value from the meta-data (10 or 11 in this example). For the example of the stack intercalate involving the values 2 and 6, the upper-rightmost cell (row 3 column 3) has been re-labelled using the symbol 10; as a consequence the value 10 in the corresponding row and column of the meta-data has been replaced by 2.(Hence an occurrence of the value 10 in the meta-data indicates that the value 10 has not been employed to relabel any cell values in the corresponding row or column.) There is a further stack intercalate, involving the values 1 and 4, the upper-leftmost cell of which is at row 3 column 7. This second intercalate is not disjoint from the first intercalate; the intercalates share a row. Hence the relabelling of a cell in that common row could not employ the value 10 without creating multiple solutions; the next augmented value, 11, is chosen in this example to re-label the value 4 in the upper-lefttmost cell of the band intercalate (at row 3 column 7); again the meta-data indicate this re-labelling. (Note that if the cell to be re-labelled in the second intercalate had been chosen in the lower row, which is not shared with the first intercalate, the value 10 could have been chosen again). The 'blocking' shown in Figure 4(b) is efficient, in that the number of 'blocked' cells is smaller than the number of intercalates (which was made possible by one cell being located within two intercalates - at row 7 column 1).

If the grid in Figure 4(b) is sent as a message, and all 18 intercalate cells deleted in transmission (indicated in Figure 4(c)), then the meta-data can be used to recover the 'blocked' cells (Figure 4(d)) - and the meta-data also indicate which original value $(1, \ldots, 9)$ corresponds to each extended value (10 and 11). From a grid such as the one in Figure 4(d), the original grid can be recovered. (Note that in practice, cells other than the 18 intercalate cells might be deleted in transmission, and not all intercalate cells might be deleted; the recovery of non-intercalate cells is generally more straightforward. Also, there may be erasures to the meta-data, but that will be addressed in Section 3.3).

A cell in a Sudoku grid may reside in none, one or many intercalates. Algorithm 2 attempts to minimise the number of intercalates 'blocked', thereby reducing the number of extended values required, by selecting to 'block' cells which reside in the most intercalates. The algorithm refers to an extended Sudoku grid (Ext_Grid[Row,Col]) which augments the original grid with meta-data (recording information about the extended values used to 'block' intercalate cells), and an array Interc_Count[Row,Col] which records in how many intercalates each original grid cell resides. It also refers to a Greedy Algorithm (marked *), which further minimises the number of extended values required and which is explained in Section 3.2.1.

Algorithm 2 Blocking Intercalates

```
Set Ext_Grid[Row,Col] to be extended Sudoku grid
Set Interc_Count[Row,Col] array to record for each cell at position [Row,Col] the number of intercalates in which that cell resides
while There exists Row and Col such that Intercalate_Counter[Row,Col]>0 do
  Set Max_Interc_Cell to the maximum value in Intercalate_Counter[Row,Col]
  Set Max_Number_Repeats to the number of cells in Intercalate_Counter[Row,Col] having value Max_Interc_Cell
  Set Extended_Val = 10
  if Max_Number_Repeats = 1 then
    Find [Row, Col] for which Interc_Count[Row,Col] = Max_Interc_Cell
    Set Block_Cell_Row = Row, Block_Cell_Col = Col
  else
    use Greedy Algorithm * to select [Row, Col] for one cell for which Interc_Count[Row,Col] = Max_Interc_Cell
    Set Block_Cell_Row = Row, Block_Cell_Col = Col
  end if
  Set Replaced_Val to be cell value in Ext_Grid[Block_Cell_Row,Block_Cell_Col]
  while Extended_Val is already used in Block_Cell_Row or in Block_Cell_Col, or Extended_Val has already been used to replace an occurrence of Replaced_Val do
    Set Extended_Val = Extended_Val + 1
  end while
  Set Ext_Grid[Block_Cell_Row,Block_Cell_Col] = Extended_Val
  Update row and column meta-data, replacing Extended_Val with Replaced_Val
  for every cell [Row,Col] in every intercalate involving the cell at [Block_Cell_Row,Block_Cell_Col] do
    Set Interc_Count[Row,Col]=Interc_Count[Row,Col]-1
  end for
end while
```

3.2.1 Greedy Algorithm for Choosing Fixed Cell Values

Algorithm 2 chooses the next intercalate cell to 'block' to be the one that resides in the maximum number of (currently 'unblocked') intercalates. Where multiple cells

share that maximum number, a greedy algorithm (Algorithm 3) is used to choose between them. The goal is to select the cell which requires the lowest extended value, thus minimising the number of extended values required in the extended grid.

Algorithm 3 Greedy Algorithm

```
Set Current_Cell to be the first cell for which Interc_Count[Row,Col] = Max_Interc_Cell
Set Max_Extended_Val = Max_Size_Extended_Grid
for every cell [Row,Col] for which Interc_Count[Row,Col] = Max_Interc_Cell do
  Set Extended_Val = 10
  Set Replaced_Val to be cell value at [Row,Col]
  while Extended_Val is already used in Block_Cell_Row or in Block_Cell_Col, or Extended_Val has already been used to replace an occurrence of Replaced_Val do
    Set Extended_Val = Extended_Val + 1
  end while
  if Extended_Val < Max_Extended_Val then
    Set Max_Extended_Val = Extended_Val
    Set Current_Cell to be the cell at [Row,Col]
  end if
end for
Return [Row, Col] of Current_Cell
```

3.3 Results and evaluation

A test set of 200 grids (developed from puzzles provided by [14]) was used to test the effectiveness of re-labelling. For each grid in the set, the approach of Section 3.1 was used to locate its intercalates, and then these intercalates were 'blocked' using the re-labelling approach of Section 3.2. The values in all cells located within any intercalate in the grid were then removed, and the resulting puzzle was solved (using a standard Sudoku solver, modified to employ the meta-data to insert the re-labelled cell values). Puzzles were then categorized according to whether they possessed a unique solution (referred to henceforth as *solved*, while those that possessed more than one solution are considered here as *not solved*).

Of the two-hundred test grids, 26 of the set were not solved. Those puzzles that did not have a unique solution must by Theorem 6 correspond to a grid which possesses some unavoidable set such that none of its constituent cells are given in the puzzle. That is, the set of all empty cells (located within intercalates) must include an unavoidable set of size larger than 4. This is illustrated in Figure 5 for one such puzzle that is not solved. The original grid is shown in Figure 5(a). Following the "blocking" of intercalates to create an extended grid, and the removal of all intercalate cell values, the solver failed to determine unique solutions for 10 cells (left blank in Figure 5(b)); these empty cells are thus shown to form an unavoidable set of size 10 (involving the values 2, 3, 4, 5 and 9).

8	4	7	3	9	5	2	1	6
2	6	3	4	7	1	5	9	8
1	9	5	6	8	2	4	3	7
7	8	4	1	2	3	9	6	5
5	3	9	8	6	4	7	2	1
6	1	2	7	5	9	8	4	3
4	7	8	2	1	6	3	5	9
9	2	1	5	3	8	6	7	4
3	5	6	9	4	7	1	8	2

(a) Original grid

8	4	7	3	9	5	2	1	6	10	11	12
2	6	3	4	11	10	5	9	8	1	7	12
1	9	5	6	8	2	4	3	7	10	11	12
7	8	4	12	2	11	9	6	5	10	3	1
11	3	9	8	12	4	7	2	1	10	5	6
6	1	10	7	5	9	8	4	3	2	11	12
4	7	8	2	1	6	3	5	10	9	11	12
		1			8	10	7		6	11	12
		6			7	1	10		8	11	12
10	10	2	10	10	1	6	8	9			
5	11	11	11	7	3	11	11	11			
12	12	12	1	6	12	12	12	12			

(b) The augmented grid, after erasure of intercalate cells and a failed attempt to solve it

Figure 5: A puzzle that cannot be solved, despite intercalate 'blocking'

Table 1: Numbers of intercalates

		Numbers of intercalates			
		min	max	median	mean
Puzzles	solved	2	19	10	9.7880
	not solved	9	18	15	14.5000
	total	2	19	10	10.1650

Table 1 shows the numbers of intercalates in the puzzles of the test set. The table indicates the minimum, maximum, median and mean (to 4 decimal places) numbers of extended values for those puzzles that solved (174), those that did not solve (26) and the total test set (200). Those puzzles that were not solved possess significantly more intercalates, on average, than those that were solved. Table 2 shows the numbers of extended values (beyond the standard values $1, 2, \ldots, 9$) employed in the extended grids for the test set puzzles, and Table 3 shows the numbers of cells that were re-labelled in the grids in order to 'block' all intercalates in each puzzle, again categorised into those puzzles that solved, those that did not solve and the total test set. In keeping with the larger average numbers of intercalates, those puzzles that were not solved required, on average, more extended values and more re-labelled cells.

The puzzles that were not solved possess unavoidable sets of sizes larger than 4 within the cells of the intercalates, and possess greater numbers of intercalate cells sharing rows and columns (increasing the number of extended val-

Table 2: Numbers of extended values used in 'blocking' intercalates

		Numbers of extended values			
		min	max	median	mean
Puzzles	solved	1	6	2	2.3103
	not solved	2	6	3	3.1538
	total	1	6	2	2.4300

Table 3: Numbers of cells re-labelled to 'block' all intercalates

		Numbers of re-labelled cells			
		min	max	median	mean
Puzzles	solved	2	10	6	6.1149
	not solved	5	12	8.5	8.2692
	total	2	12	6	6.4000

ues required). This increases the overhead of meta-data (while the extended grid is still not adequate to ensure that erased cell values can be corrected).

The introduction of meta-data inevitably reduces the rate of the Coding Scheme. This increase may be justified if the erasure-correcting capability of the scheme is sufficiently improved. Taking the ceiling of the average number of extended values required (Table 2) to be 3, the extended grid to be transmitted within the proposed erasure correction scheme contains an additional 54 cells of meta-data, working over an alphabet of 12 symbols, and the useful data transmitted remains as in Equation 1. This leads to a rate of

$$\frac{\log_{12}(9!^3)}{135} = 0.11449. \tag{2}$$

Correspondingly, the required operation would be at erasure probabilities approaching $1 - 0.11449 = 0.88551$. While the erasure correction properties of the scheme have been increased greatly by the meta-data, two problems remain. Firstly, the failure of the robustness of a significant proportion of grids to erasure of intercalate cells (13%) is significant; the difficulties of the presence of intercalates reported in [3] have been repeated, albeit on a lower scale, by the presence of larger unavoidable sets. Secondly, in a practical implementation, the meta-data will be subject to the same rates of erasure as the original grid data, and the erasure of these 'blocking' values will greatly reduce the erasure correcting properties of the extended grid. In combination, these difficulties make the requirement to perform at such a high erasure probability unlikely.

4 Intercalate Properties of 2-Quasi-Magic Sudoku Grids

An alternative approach to overcoming the difficulty of the low rate of an erasure correction scheme that incorporates the Sudoku structure is to employ a subset of all Sudoku grids that possess beneficial unavoidable set properties (explained in Section 1). Specifically, the grids should form a class that possess few or no intercalates, such as 2-Quasi-Magic Sudoku (which has previously been proposed for erasure correction in [10]). This Section begins with some properties of 2-Quasi-Magic Sudoku reported in [15].

Definition 11. *A Δ-Quasi-Magic Sudoku grid (Q) is a $S^{3,3}$ with the additional constraints that the values contained in the tiers, pillars and diagonals of the mini-grids sum to an integer in the interval $[15 - \Delta, 15 + \Delta]$ for $\Delta \in \{0, \ldots, 9\}$.*

A 2-Quasi-Magic Sudoku grid is a Sudoku grid in which the values in all the tiers, pillars and diagonals of every mini-grid sum to a number between 13 and 17. There are $248,832$ 2-Quasi-Magic Sudoku grids [15] (a small subset of all Sudoku grids that meet the additional constraints). Each of these grids is such that its rows, columns, bands, stacks and values can be permuted to form other grids within that total number. If the grids are categorised into sets, such that each grid within a set can be permuted, in the above ways, into any other grid in that set, there will be 40 such sets. These 40 sets are referred to as *isomorphic classes* of 2-Quasi-Magic Sudoku [16]. Now, consideration is given to the necessary arrangement of given values within a 2-Quasi-Magic Sudoku puzzle in order for that puzzle to have a unique solution.

Lemma 12. *The removal of the values from at most three cells from a 2-Quasi-Magic Sudoku grid results in a puzzle with a unique solution.*

Proof. Since 2-Quasi-Magic Sudoku grids are a subset of Sudoku grids then this follows directly from Lemma 3. □

A Sudoku puzzle cannot be guaranteed to have a unique solution if the values from four cells are removed from a complete grid (Lemma 4). The four non-given cells cannot be solved uniquely if they form a Sudoku intercalate (Definition 8). Similarly a 2-Quasi-Magic Sudoku intercalate is defined in Definition 13.

Definition 13. *A 2-Quasi-Magic Sudoku intercalate is defined as:*

$$[Q_{a,b}]_{i,j} = [Q_{a,d}]_{m,k} \quad \text{and} \quad [Q_{a,b}]_{m,j} = [Q_{a,d}]_{i,k}$$

or within a stack:

$$[Q_{a,b}]_{i,j} = [Q_{c,b}]_{m,k} \quad \text{and} \quad [Q_{a,b}]_{i,k} = [Q_{c,b}]_{m,j}$$

for $b, d, i, k \in \{1, \ldots, x\}$ with $a \neq c$ and $i \neq m$, and $a, c, i, k \in \{1, \ldots, y\}$ with $b \neq d$ and $j \neq k$. The tiers and pillars of $Q_{b,a}$ and $Q_{b,c}$ must still sum to an integer in the interval $[13, 17]$ in both arrangements of the two values within the designated cells.

A Sudoku intercalate and a 2-Quasi-Magic Sudoku intercalate differ only in the fact that neither arrangement of the values in the 2-Quasi-Magic Sudoku intercalate can contradict the sum constraint. The latter is illustrated in Figure 6 (in which a 2-Quasi-Magic Sudoku band intercalate is shown in bold, and a 2-Quasi-Magic Sudoku stack intercalate is shown in italics).

2	9	5	8	3	4	6	7	1
6	*3*	*4*	1	**7**	9	2	5	8
7	1	8	5	**6**	2	9	3	4
3	8	6	9	1	5	4	2	7
9	5	1	2	4	7	8	6	3
4	2	7	6	8	3	1	9	5
8	*4*	*3*	7	2	6	5	1	9
1	6	9	3	5	8	7	4	2
5	7	2	4	9	1	3	8	6

Figure 6: A 2-Quasi-Magic Sudoku grid with two 2-Quasi-Magic Sudoku intercalates

Lemma 14. *From a completed 2-Quasi-Magic Sudoku grid the removal of four values from any four cells which do not form a 2-Quasi-Magic Sudoku intercalate result in a puzzle with 77 givens and a unique solution.*

Proof. Follows directly from Corollary 9. □

In a similar way to Theorem 10 (for Sudoku) it is shown in [16] that fewer than 1% of 2-Quasi-Magic Sudoku puzzles with 77 values are not uniquely solvable.

A 2-Quasi-Magic Sudoku puzzle does not have a unique solution if some of the values removed from the cells form an unavoidable set, in which more than one arrangement of the values satisfies the sum constraint.

Theorem 15. *If any 2-Quasi-Magic Sudoku grid possesses (by some permutation) an unavoidable set applicable to 2-Quasi-Magic Sudoku, then to ensure a unique solution at least one of these cells must contain a given in any corresponding puzzle.*

Theorem 15 gives a necessary and sufficient condition for a 2-Quasi-Magic Sudoku puzzle to have a unique solution.

4.1 Results and evaluation

Only the grids in four of the 40 isomorphic classes contain 2-Quasi-Magic Sudoku intercalates [16], and they all possess exactly two (illustrated by one representative grid from one such class in Figure 6). (As a point of interest, the intercalate 'blocking' algorithm of Section 3 was employed on these and was found to be sufficient to recover the grids correctly following the removal of those cells located within the intercalates.)

This low number of intercalates might initially support the use of the 2-Quasi-Magic Sudoku structure for an erasure-correction scheme, given that the presence of intercalates in Sudoku grids has been shown to be a significant consideration against the use of general Sudoku grids [3]. However, the rate of such a scheme must be considered.

In Equation 1, the rate of the scheme employing a Sudoku structure was calculated on the basis that a message can be encoded in the three mini-grids along one diagonal. Each mini-grid allowed for any one of the 9! ways of arranging the values $1, 2, \ldots, 9$. The additional constraints of 2-Quasi-Magic Sudoku result in only 736 possible ways of arranging the values $1, 2, \ldots, 9$ in each mini-grid [16]. If it is assumed that a valid 2-Quasi-Magic Sudoku grid can always be constructed regardless of the mini-grids chosen to be placed along one diagonal, the resulting rate would be

$$\frac{\log_9(736^3)}{81} = 0.11127. \tag{3}$$

This new rate (which is lower than those given in Equations 1 and 2) would require that the scheme operate efficiently at erasure probabilities approaching $1 - 0.11127 = 0.88873$ to be competitive with established codes.

In fact, the choice of mini-grids along the diagonal would prevent completion to a valid grid in some cases (due largely to the restrictions on the centre values of mini-grids in a 2-Quasi-Magic Sudoku grid [15]). The rate given in Equation 3 must therefore be regarded as 'generously high'. Even this value seems to be unrealistic for an erasure-correction scheme. In practice, the number of distinct 2-Quasi-Magic Sudoku grids into which information must be encoded is too small to form the basis of an erasure-correction scheme.

5 Conclusions and Future Work

This paper examined the use of Sudoku grid meta-data to 'block' intercalates, to assist in the unique completion of grids received in transmission with all cell values erased from one or more intercalate pattern. The method described, which introduces additional symbols and relabels a minimum of cell values, successfully 'blocks' intercalates in most cases tested. However, when the cells comprising the intercalates include an unavoidable set of larger size, and no cells in that larger set have been re-labelled, the method cannot guarantee recovery of all erased cells. This difficulty, combined with the reduction in the rate of the scheme due to the meta-data and the corresponding increase in the erasure probabilities at which the scheme would have to perform correction, make this approach impractical as it is.

Further investigation is required of the numbers and locations of unavoidable sets of larger sizes, to establish whether the 'blocking' process described here can be

modified to ensure that the cells selected for re-labelling include cells present in those larger sets. If the amount of meta-data required for blocking is kept relatively low, a robust and effective scheme may still be produced, although analysis would also be required of the effects of erasures in the meta-data.

The use of a subset of all Sudoku grids, 2-Quasi-Magic Sudoku, reduced the effects on erasure recovery due to the presence of intercalates, as an alternative approach to the use of meta-data. The rate was reduced even more dramatically, due to the small number of grids available into which information to be transmitted can be encoded.

While that particular subset of Sudoku is too small to form a practical scheme, it may be possible to identify a much larger class possessing beneficial unavoidable set properties. In the absence of knowledge concerning such a class, the notion of a practical scheme remains, and further work is required to identify subsets of Sudoku grids through the examination of unavoidable set properties.

References

[1] E. Soedarmandji and R. J. McEliece, "Iterative decoding for Sudoku and Latin square codes.," in *Forty-Fifth Annual Allerton Conference*, pp. 488–494, 2007.

[2] L. A. Phillips, S. Perkins, P. A. Roach, and D. H. Smith, "Sudoku and error-correcting codes," in *Proceedings of the 4^{th} Research Student Workshop* (P. Roach, ed.), pp. 65–69, University of Glamorgan, 12th March 2009.

[3] L. A. Phillips, "Erasure-correcting Codes Derived from Sudoku and Related Combinatorial Structures,," Technical Report UG-M-10-1, Division of Mathematics and Statistics, University of Glamorgan, Pontypridd, U.K., April 2010.

[4] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. North-Holland, 1977.

[5] C.-D. Lee and Y. Chang, "Decoding the (41, 21, 9) quadratic residue code," in *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS)* (S. I. Ao, O. Castillo, C. Douglas, D. D. Feng, and J.-A. Lee, eds.), vol. 2, pp. 780–783, IAENG, March 2010.

[6] F. Merazka, "Improved packet loss recovery using interleaving for CELP-type speech coders in packet networks," *International Journal of Computer Science*, vol. 36, no. 1, 2009.

[7] F. Merazka, "Forward error correction concealment method for celp-based coders in packet networks," in *Proceedings of the World Congress on Engineering 2010 (WCE10)*, vol. 1, pp. 178–181, IAENG, June/July 2010.

[8] R. P. Davies, P. A. Roach, and S. Perkins, "The Use of Problem Domain Information in the Automated Solution of Kakuro Puzzles," *International Journal of Computer Science (IJCS)*, vol. 37, no. 2, pp. 118–127, 2010.

[9] S. Gubin, "A sudoku solver," in *Proceedings of the World Congress on Engineering and Computer Science (WCECS2009)*, vol. 1, pp. 223–228, IAENG, October 2009.

[10] P. A. Roach, I. J. Grimstead, S. K. Jones, and S. Perkins, "A knowledge-rich approach to the rapid enumeration of Quasi-Magic Sudoku search spaces," in *Proceedings of ICAART 2009, the 1st International Conference on Agents and Artificial Intelligence* (J. Filipe, A. Fred, and B. Sharp, eds.), (Porto, Portugal), pp. 246–254, January 2009.

[11] H. W. Norton, "The 7×7 squares," *Ann. Eugenics*, vol. 9, pp. 269–307, 1939.

[12] Dobrichev, "Solution grids in u-space (unavoidable sets)." Available at: `http://www.setbb.com/phpbb/viewtopic.php?t=1711&view=next&sid=2b623e2deca881952efb6711b3abe1ed&mforum=sudoku`.

[13] B. Felgenhauer and F. Jarvis, "Mathematics of Sudoku I," *Mathematical Spectrum*, vol. 39, pp. 15–22, September 2006.

[14] J. Bumgardner, "Free printable sudoku puzzles." Available at: `http://www.krazydad.com/sudoku`.

[15] S. K. Jones, S. Perkins, and P. A. Roach, "Properties, isomorphisms and enumeration of 2-Quasi-Magic Sudoku," *Discrete Mathematics, doi:10.1016/j.disc.2010.09.026*, 2010.

[16] S. K. Jones, *On Properties of Sudoku and Similar Combinatorial Structures*. PhD thesis, University of Glamorgan, 2010.