Discrete Mathematics 312 (2012) 3306–3315

Contents lists available at SciVerse ScienceDirect

Discrete Mathematics

journal homepage: www.elsevier.com/locate/disc

# The Sudoku completion problem with rectangular hole pattern is NP-complete

Ramón Béjar [a,*], Cèsar Fernández [a], Carles Mateu [a], Magda Valls [b]

[a] *Department of Computer Science, University of Lleida, Jaume II 69, Lleida - 25001, Spain*

[b] *Department of Mathematics, University of Lleida, Jaume II 69, Lleida - 25001, Spain*

ARTICLE INFO

*Article history:*
Received 20 May 2011
Received in revised form 13 July 2012
Accepted 22 July 2012
Available online 14 August 2012

*Keywords:*
Latin square
Quasigroup
Sudoku
NP-complete

ABSTRACT

The sudoku completion problem is a special case of the latin square completion problem and both problems are known to be NP-complete. However, in the case of a rectangular hole pattern – i.e. each column (or row) is either full or empty of symbols – it is known that the latin square completion problem can be solved in polynomial time. Conversely, we prove in this paper that the same rectangular hole pattern still leaves the sudoku completion problem NP-complete.



## 1. Introduction

A *latin square* of order $s$ is an $s \times s$ matrix filled with $s$ different symbols, such that no symbol is repeated in a row or column.

A *partial latin square* is a partially filled matrix, with some empty cells, which also satisfies that no symbol is repeated in a row or column. A problem that has received considerable attention is the latin square completion problem: given a partial latin square, determine if the empty cells of the matrix can be filled, such that the final matrix is a latin square. This problem has been shown to be NP-complete [8], and its average computational complexity has also been studied [11,10,3]. Another line of research focuses on how to construct partial latin squares with guaranteed solution but high average computational complexity [1] from complete latin squares generated with Markov chain algorithms [12].

A *sudoku* of order $s$, with $s = nm$, is an $s \times s$ matrix filled with $s$ different symbols, partitioned into $s$ rectangular $n \times m$ block regions, such that no symbol is repeated in a row, column or block region. We denote by a *region row* a set of $n$ block regions which are horizontally aligned. Analogously, we denote by a *region column* a set of $m$ block regions which are vertically aligned. Hence, there are $m$ region rows and $n$ regions columns. The sudoku problem we consider here is a generalization of the popular *number place* puzzle, that is a sudoku of order 9 with $3 \times 3$ block regions. At the same time, a sudoku is a special case of a gerechte design, in which the matrix of order $s$ is partitioned into $s$ regions, where a region can be any set of $s$ cells from the matrix. The connections between gerechte designs and design of comparative experiments and coding theory have been explored in [4]. See also [5] for a more detailed study about the use of gerechte designs in the design of agricultural experiments.

* Corresponding author.
*E-mail addresses:* ramon@diei.udl.cat (R. Béjar), cesar@diei.udl.cat (C. Fernández), carles@diei.udl.cat (C. Mateu), magda@matematica.udl.cat (M. Valls).

0012-365X/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.disc.2012.07.022

Analogously to the case of partial latin squares, one can also study the completion problem for *partial sudokus*: determining if a partial sudoku can be completed. Nowadays, this problem has started to receive attention in the artificial intelligence community [15,14,2].

In this paper, we study the computational complexity of the sudoku completion problem in the particular case where the filled and empty cells follow a *rectangular hole pattern*, i.e. where either each row is completely full or completely empty, or each column is completely full or completely empty. It follows from Hall's theorem that the latin square completion problem is polynomially solvable for the case of rectangular hole pattern. By contrast, we prove that, for the sudoku completion problem, this case still leaves the problem NP-complete. As a proof, we construct a polynomial time reduction from the latin square completion problem to the *sudoku completion problem with rectangular hole pattern*. In fact, without lack of generality, the reduction is defined for the particular case of *column hole pattern*, i.e. each column is either full or empty.

The paper is structured as follows. In Section 2 we define the sudoku completion problem, and we recall that this problem, for arbitrary rectangular block regions and arbitrary hole patterns, is NP-complete. This result is used as a starting point for our main result, given in Section 3. There we present the NP-completeness proof for the case of the column hole pattern, which extends the approach followed in Section 2. In constructing that reduction we introduce a special class of latin square, called canonical zero-diagonal latin square.

## 2. The general sudoku completion problem

The latin square completion problem was shown to be NP-complete in [8]. Since then, several special cases of the latin square completion problem have been studied, see for example [6,9] or [13]. Those special cases are defined on how missing cells are distributed on the latin square. This distribution is known as the *hole pattern*, for instance random hole pattern, balanced hole pattern or rectangular hole pattern. Conversely in sudoku problems, such patterns also have an important impact on the hardness of problems, as well as the block region shape [2].

It has been shown in [16], that the sudoku completion problem is NP-complete for the particular case of square block regions ($n$ rows and $n$ columns in each region block). As shown in [2], even when block regions are not square the completion problem is also NP-complete. That proof is presented below for a better understanding and contextualization of the new results presented in Section 3. The proof for this case is a generalization of the proof of [16], and provides a reduction from the latin square completion problem to the sudoku completion problem with rectangular block regions.

**Theorem 1.** *The sudoku completion problem of order $s$, $s = nm$, with rectangular block regions ($n \neq m$), is NP-complete.*

**Proof.** Given an instance of the latin square completion problem of order $n$, the reduction follows by constructing an instance of the sudoku completion problem of order $s = nm$ and $n < m$ (rectangular block regions) such that the sudoku completion problem instance has a solution iff the latin square completion problem instance has a solution. The entries of the latin square completion problem instance are embedded into the first columns of the regions of the first region row.

More formally, let $L$ be the latin square completion problem instance and $S$ the sudoku completion problem instance, and let their entries be denoted by

$$L = (L_{i,j}), \quad 0 \leq i, j \leq n-1;$$
$$S = (S_{i,j}^{k,l}), \quad 0 \leq i, l \leq m-1,\ 0 \leq j, k \leq n-1,$$

where $L_{i,j}$ corresponds to the entry of $L$ located at the $i$-th row and $j$-th column, and $S_{i,j}^{k,l}$ is the entry of $S$ located at the $k$-th row and $l$-th column inside the $i$-th region row, $j$-th region column. Notice that the first row and first column are numbered as zero.

Then, the reduction maps $L$ into $S$ with entries

$$S_{i,j}^{k,l} = \begin{cases} (L_{k,j}, 0), & \text{if } i = l = 0 \text{ and } L_{k,j} \neq \perp \\ \perp, & \text{if } i = l = 0 \text{ and } L_{k,j} = \perp \\ (k + j \pmod n, i + l \pmod m), & \text{otherwise.} \end{cases}$$

where $\perp$ denotes that the cell is empty. Notice that when cells are not empty their entries are ordered pairs.

Under this construction, the original entries of the latin square completion problem instance are placed at positions with $i = l = 0$. Notice that the contents of the cells in $S$ are not repeated, neither in the same row, column nor rectangular region.

It is straightforward to observe that the sudoku completion problem instance has a solution if and only if the latin square completion problem instance has a solution. □

## 3. The sudoku completion problem with column hole pattern

In this section we describe a reduction from the latin square completion problem to the sudoku completion problem with column hole pattern.

Firstly, in the next subsection, we define and construct canonical zero-diagonal latin squares, which are used in the reduction as auxiliary designs. Afterwards, Section 3.2 presents in detail the reduction.

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 0 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 2 \\ 2 & 3 & 0 & 5 & 6 & 7 & 8 & 9 & 1 & 4 \\ 3 & 4 & 5 & 0 & 7 & 8 & 9 & 1 & 2 & 6 \\ 4 & 5 & 6 & 7 & 0 & 9 & 1 & 2 & 3 & 8 \\ 5 & 6 & 7 & 8 & 9 & 0 & 2 & 3 & 4 & 1 \\ 6 & 7 & 8 & 9 & 1 & 2 & 0 & 4 & 5 & 3 \\ 7 & 8 & 9 & 1 & 2 & 3 & 4 & 0 & 6 & 5 \\ 8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 & 0 & 7 \\ 9 & 2 & 4 & 6 & 8 & 1 & 3 & 5 & 7 & 0 \end{pmatrix}$$

**Fig. 1.** Example of CZD-LS of order $n = 10$.

### *3.1. Construction of canonical zero-diagonal latin squares*

A *canonical zero-diagonal latin square* (CZD-LS) of order $n$ is a latin square $L = (L_{i,j}),\ 0 \le i, j \le n - 1$, with entries in $\mathbb{Z}_n$ such that $L_{i,i} = 0$, $L_{i,0} = i$ and $L_{0,j} = j$. As an example, Fig. 1 shows a CZD-LS of order $n = 10$.

Notice that CZD-LS of orders $n = 1, 2$ are trivially constructed and for $n = 3$ such a construction does not exist. In general, the existence of CZD-LS for $n \ge 4$ follows from Lemma 2.3 of [7].

However, since the construction of a CZD-LS of order $n \ge 4$ is one of the building steps of our polynomial time reduction for the NP-completeness proof, we need to guarantee, not only the existence, but also how to obtain a CZD-LS in polynomial time.

**Proposition 2.** *A CZD-LS of order $n \ge 4$ can be constructed in polynomial-time.*

**Proof.** For the construction, we will distinguish between the case $n$ even and $n$ odd.

In the case $n$ even, the entries $L_{i,j}$ of a CZD-LS $L$ can be constructed as follows

$$L_{i,j} = \begin{cases} 0, & \text{if } i = j, \\ i + j \pmod{n}, & \text{if } j \le n - i - 1 \text{ and } i \ne j, \\ i + j + 1 \pmod{n}, & \text{if } n - i - 1 < j < n - 1,\ i < n - 1 \\ & \text{and } i \ne j \\ 2i \pmod{n}, & \text{if } 0 < i < \dfrac{n}{2} \text{ and } j = n - 1, \\ 2j \pmod{n}, & \text{if } i = n - 1 \text{ and } 0 < j < \dfrac{n}{2}, \\ 2\left(i - \dfrac{n}{2}\right) + 1 \pmod{n}, & \text{if } \dfrac{n}{2} \le i < n - 1 \text{ and } j = n - 1, \\ 2\left(j - \dfrac{n}{2}\right) + 1 \pmod{n}, & \text{if } i = n - 1 \text{ and } \dfrac{n}{2} \le j < n - 1. \end{cases}$$

Notice that this matrix is symmetric. Hence, to show that it is a latin square, it is enough to show that the values in each row are different. We should distinguish between the following three situations:

- Case $0 \le i < n/2$.
  In row $i$, the values are
  $$i, i + 1, \ldots, 2i - 1, 0, 2i + 1, \ldots, n - 1, 1, 2, \ldots, i - 1, 2i,$$
  hence, they cover all possible values from 0 to $n - 1$.
- Case $n/2 \le i < n - 1$.
  In this case, the values for row $i$ are:
  $$i, i + 1, \ldots, n - 1, 1, 2, \ldots, 2i, 0, 2i + 2, \ldots, i - 1, 2i + 1.$$
  Again, they cover all possible values.
- Case $i = n - 1$.
  In the last row, the values are
  $$n - 1, 2, 4, \ldots, n - 2, 1, 3, 5, \ldots, n - 3, 0,$$
  so they also are pairwise different.

$$L = \left(\begin{array}{c|ccccccccc|c}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \hline
1 & 0 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & \mathbf{2} & \\
2 & 3 & 0 & 5 & 6 & 7 & 8 & \mathbf{9} & 1 & 4 & \\
3 & \mathbf{4} & 5 & 0 & 7 & 8 & 9 & 1 & 2 & 6 & \\
4 & 5 & \mathbf{6} & 7 & 0 & 9 & 1 & 2 & 3 & 8 & \\
5 & 6 & 7 & \mathbf{8} & 9 & 0 & 2 & 3 & 4 & 1 & \\
6 & 7 & 8 & 9 & \mathbf{1} & 2 & 0 & 4 & 5 & 3 & \\
7 & 8 & 9 & 1 & 2 & \mathbf{3} & 4 & 0 & 6 & 5 & \\
8 & 9 & 1 & 2 & 3 & 4 & \mathbf{5} & 6 & 0 & 7 & \\
9 & 2 & 4 & 6 & 8 & 1 & 3 & 5 & \mathbf{7} & 0 & \\ \hline
10 & & & & & & & & & & 0
\end{array}\right)$$

**Fig. 2.** First steps in the construction of a CZD-LS of order $n = 11$.

$$L = \left(\begin{array}{c|ccccccccc|c}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \hline
1 & 0 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & \mathbf{10} & 2 \\
2 & 3 & 0 & 5 & 6 & 7 & 8 & \mathbf{10} & 1 & 4 & 9 \\
3 & \mathbf{10} & 5 & 0 & 7 & 8 & 9 & 1 & 2 & 6 & 4 \\
4 & 5 & \mathbf{10} & 7 & 0 & 9 & 1 & 2 & 3 & 8 & 6 \\
5 & 6 & 7 & \mathbf{10} & 9 & 0 & 2 & 3 & 4 & 1 & 8 \\
6 & 7 & 8 & 9 & \mathbf{10} & 2 & 0 & 4 & 5 & 3 & 1 \\
7 & 8 & 9 & 1 & 2 & \mathbf{10} & 4 & 0 & 6 & 5 & 3 \\
8 & 9 & 1 & 2 & 3 & 4 & \mathbf{10} & 6 & 0 & 7 & 5 \\
9 & 2 & 4 & 6 & 8 & 1 & 3 & 5 & \mathbf{10} & 0 & 7 \\ \hline
10 & 4 & 6 & 8 & 1 & 3 & 5 & 9 & 7 & 2 & 0
\end{array}\right)$$

**Fig. 3.** Construction of a CZD-LS of order $n = 11$.

In the case $n$ odd, consider a CZD-LS $L'$ of even order $n - 1$ constructed as shown above. Then, a CZD-LS $L$ of order $n$ can be obtained from $L'$ following the steps described below. Figs. 2 and 3 show an example of such a construction using the CZD-LS of order $n - 1 = 10$ in Fig. 1 to construct a CZD-LS of order $n = 11$.

- To obtain $L$, add to $L'$ a last column and a last row.
- Then $L_{0,n-1} = n - 1$, $L_{n-1,0} = n - 1$ and $L_{n-1,n-1} = 0$.
- Consider the submatrix of $L$ consisting of cells $L_{i,j}$ with $1 \le i, j \le n-2$. In Fig. 2 the submatrix is delimited with a square.
- In such submatrix, consider a *latin transversal*.[1] Such a transversal always exists in this submatrix. One possible choice would be taking the following entries: $L_{1,n-2}$, $L_{2,n-4}$, $L_{i,i-2}$ for $i \in \{3, \ldots, n-3\}$, and $L_{n-2,n-3}$. Following with the example, such entries are the ones typed in bold in Fig. 2.
- Then, the value of each entry in the transversal is placed at the last column and last row, and replaced by value $n - 1$. More formally, let $k$ be the value of the latin transversal at row $i$ and column $j$. Then, $L_{i,n-1} = k$, $L_{n-1,j} = k$ and $L_{i,j} = n-1$. Fig. 3 shows the resulting CZD-LS in our example.

Following this construction, it can be easily proven that entries in the same row or in the same column are always different. □

### *3.2. NP-completeness proof*

In the NP-completeness proof we provide a polynomial time reduction from the latin square completion problem to the sudoku completion problem with column hole pattern.

[1] A *transversal array* is a set of $n$ cells in an $n \times n$ square such that no two come from the same row and no two come from the same column. Then, a *latin transversal* is a transversal such that no two cells contain the same element.

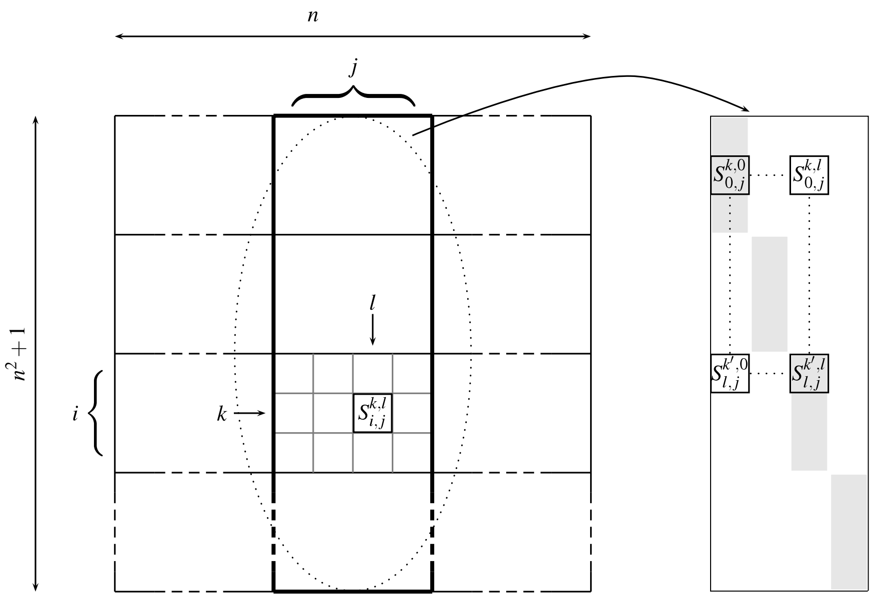


**Fig. 4.** Detail of the structure of the region columns. By construction we have $S^{k,0}_{0,j} = S^{k',l}_{l,j}$ and $S^{k,l}_{0,j} = S^{k',0}_{l,j}$.

More precisely, given a latin square completion problem instance $L$ of order $n$, the reduction follows by constructing a sudoku completion problem instance $S$ of order $n \times (n^2 + 1)$ with column hole pattern. The overall idea of this reduction is similar to the one used in the proof of Theorem 1. That is, columns in the latin square completion problem instance are mapped to the first column of each rectangular block region in the first region row. The remaining cells are filled using a CZD-LS $C$ of order $n^2 + 1$. Finally, the first column of each region column is emptied, after appropriately swapping the contents of cells mapped from the original latin square completion problem instance with the contents of appropriate cells from other columns, to preserve their effect.

We will show that a solution of such sudoku completion problem instance exists if, and only if, a solution of the original latin square completion problem instance also exists.

We present next in detail the different steps involved in the reduction.

We first describe the construction of $S$ from $L$ and $C$. Let $L$ be the original partial latin square of size $n$ and $C$ the CZD-LS of size $n^2 + 1$. Their entries are denoted by:

$$L = (L_{i,j}), \quad 0 \le i, j \le n - 1;$$
$$C = (C_{i,j}), \quad 0 \le i, j \le n^2.$$

Empty cells in $L$ are assigned the value $\perp$. Notice that the first row and column are numbered as zero.

Then the partial sudoku of size $n(n^2 + 1)$ is denoted as

$$S = (S^{k,l}_{i,j}), \quad 0 \le i, l \le n^2,\ 0 \le j, k \le n - 1,$$

where $S^{k,l}_{i,j}$ corresponds to the entry located at the $i$-th region row, $j$-th region column, and inside such a region it is placed on the $k$-th row and $l$-th column (see Fig. 4). These entries are ordered pairs, defined as follows

$$S^{k,l}_{i,j} = \begin{cases} (L_{k,j}, C_{0,0}) = (L_{k,j}, 0), & \text{if } i = l = 0, \\ \big(j + k \ (\mathrm{mod}\ n),\ C_{i,i}\big) = (j + k \ (\mathrm{mod}\ n), 0), & \text{if } i = l \neq 0, \\ (i + j + k \ (\mathrm{mod}\ n),\ C_{i,l}), & \text{otherwise.} \end{cases}$$

In fact, the first condition is responsible for embedding the information from the partial latin square into the partial sudoku. When in the partial latin square the cell is empty, it remains also empty in the partial sudoku. Fig. 5 shows an example of this construction for a partial latin square of size $n = 3$.

It can easily be seen that this construction is a partial sudoku. This is due to the fact that the second component corresponds to the cells of the CZD-LS, which is combined with a first component whose values are $0, 1, \ldots, n - 1$, which ensures that the contents are different in each row, each column and each region.

Partial latin square $L$:

| | | |
|---|---|---|
| 2 | | 0 |
| 1 | 0 | |
| 0 | 1 | 2 |

Partial Sudoku $S$:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2,0 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 0,0 | 2,1 | 2,2 | 2,3 | 2,4 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 |
| 1,0 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 0,0 | 2,1 | 2,2 | 2,3 | 2,4 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 |
| 0,0 | 2,1 | 2,2 | 2,3 | 2,4 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | 1,0 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 2,0 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 |
| 1,1 | 0,0 | 1,3 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 1,2 | 2,1 | 1,0 | 2,3 | 2,4 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | 2,2 | 0,1 | 2,0 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 0,2 |
| 2,1 | 1,0 | 2,3 | 2,4 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | 2,2 | 0,1 | 2,0 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 0,2 | 1,1 | 0,0 | 1,3 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 1,2 |
| 0,1 | 2,0 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 0,2 | 1,1 | 0,0 | 1,3 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 1,2 | 2,1 | 1,0 | 2,3 | 2,4 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | 2,2 |
| 2,2 | 2,3 | 0,0 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | 2,1 | 2,4 | 0,2 | 0,3 | 1,0 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 0,1 | 0,4 | 1,2 | 1,3 | 2,0 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 1,1 | 1,4 |
| 0,2 | 0,3 | 1,0 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 0,1 | 0,4 | 1,2 | 1,3 | 2,0 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 1,1 | 1,4 | 2,2 | 2,3 | 0,0 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | 2,1 | 2,4 |
| 1,2 | 1,3 | 2,0 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 1,1 | 1,4 | 2,2 | 2,3 | 0,0 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | 2,1 | 2,4 | 0,2 | 0,3 | 1,0 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 0,1 | 0,4 |
| 0,3 | 0,4 | 0,5 | 0,0 | 0,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,6 | 1,3 | 1,4 | 1,5 | 1,0 | 1,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,6 | 2,3 | 2,4 | 2,5 | 2,0 | 2,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,6 |
| 1,3 | 1,4 | 1,5 | 1,0 | 1,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,6 | 2,3 | 2,4 | 2,5 | 2,0 | 2,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,6 | 0,3 | 0,4 | 0,5 | 0,0 | 0,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,6 |
| 2,3 | 2,4 | 2,5 | 2,0 | 2,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,6 | 0,3 | 0,4 | 0,5 | 0,0 | 0,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,6 | 1,3 | 1,4 | 1,5 | 1,0 | 1,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,6 |
| 1,4 | 1,5 | 1,6 | 1,7 | 0,0 | 1,9 | 1,1 | 1,2 | 1,3 | 1,8 | 2,4 | 2,5 | 2,6 | 2,7 | 1,0 | 2,9 | 2,1 | 2,2 | 2,3 | 2,8 | 0,4 | 0,5 | 0,6 | 0,7 | 2,0 | 0,9 | 0,1 | 0,2 | 0,3 | 0,8 |
| 2,4 | 2,5 | 2,6 | 2,7 | 1,0 | 2,9 | 2,1 | 2,2 | 2,3 | 2,8 | 0,4 | 0,5 | 0,6 | 0,7 | 2,0 | 0,9 | 0,1 | 0,2 | 0,3 | 0,8 | 1,4 | 1,5 | 1,6 | 1,7 | 0,0 | 1,9 | 1,1 | 1,2 | 1,3 | 1,8 |
| 0,4 | 0,5 | 0,6 | 0,7 | 2,0 | 0,9 | 0,1 | 0,2 | 0,3 | 0,8 | 1,4 | 1,5 | 1,6 | 1,7 | 0,0 | 1,9 | 1,1 | 1,2 | 1,3 | 1,8 | 2,4 | 2,5 | 2,6 | 2,7 | 1,0 | 2,9 | 2,1 | 2,2 | 2,3 | 2,8 |
| 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | 0,0 | 2,2 | 2,3 | 2,4 | 2,1 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 1,0 | 0,2 | 0,3 | 0,4 | 0,1 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 2,0 | 1,2 | 1,3 | 1,4 | 1,1 |
| 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 1,0 | 0,2 | 0,3 | 0,4 | 0,1 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 2,0 | 1,2 | 1,3 | 1,4 | 1,1 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | 0,0 | 2,2 | 2,3 | 2,4 | 2,1 |
| 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 2,0 | 1,2 | 1,3 | 1,4 | 1,1 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | 0,0 | 2,2 | 2,3 | 2,4 | 2,1 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 1,0 | 0,2 | 0,3 | 0,4 | 0,1 |
| 0,6 | 0,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,0 | 0,4 | 0,5 | 0,3 | 1,6 | 1,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,0 | 1,4 | 1,5 | 1,3 | 2,6 | 2,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,0 | 2,4 | 2,5 | 2,3 |
| 1,6 | 1,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,0 | 1,4 | 1,5 | 1,3 | 2,6 | 2,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,0 | 2,4 | 2,5 | 2,3 | 0,6 | 0,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,0 | 0,4 | 0,5 | 0,3 |
| 2,6 | 2,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,0 | 2,4 | 2,5 | 2,3 | 0,6 | 0,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,0 | 0,4 | 0,5 | 0,3 | 1,6 | 1,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,0 | 1,4 | 1,5 | 1,3 |
| 1,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,3 | 1,4 | 0,0 | 1,6 | 1,5 | 2,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,3 | 2,4 | 1,0 | 2,6 | 2,5 | 0,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,3 | 0,4 | 2,0 | 0,6 | 0,5 |
| 2,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,3 | 2,4 | 1,0 | 2,6 | 2,5 | 0,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,3 | 0,4 | 2,0 | 0,6 | 0,5 | 1,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,3 | 1,4 | 0,0 | 1,6 | 1,5 |
| 0,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,3 | 0,4 | 2,0 | 0,6 | 0,5 | 1,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,3 | 1,4 | 0,0 | 1,6 | 1,5 | 2,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,3 | 2,4 | 1,0 | 2,6 | 2,5 |
| 2,8 | 2,9 | 2,1 | 2,2 | 2,3 | 2,4 | 2,5 | 2,6 | 0,0 | 2,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 1,0 | 0,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 | 2,0 | 1,7 |
| 0,8 | 0,9 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 1,0 | 0,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 | 2,0 | 1,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,3 | 2,4 | 2,5 | 2,6 | 0,0 | 2,7 |
| 1,8 | 1,9 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 | 2,0 | 1,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,3 | 2,4 | 2,5 | 2,6 | 0,0 | 2,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 1,0 | 0,7 |
| 0,9 | 0,2 | 0,4 | 0,6 | 0,8 | 0,1 | 0,3 | 0,5 | 0,7 | 0,0 | 1,9 | 1,2 | 1,4 | 1,6 | 1,8 | 1,1 | 1,3 | 1,5 | 1,7 | 1,0 | 2,9 | 2,2 | 2,4 | 2,6 | 2,8 | 2,1 | 2,3 | 2,5 | 2,7 | 2,0 |
| 1,9 | 1,2 | 1,4 | 1,6 | 1,8 | 1,1 | 1,3 | 1,5 | 1,7 | 1,0 | 2,9 | 2,2 | 2,4 | 2,6 | 2,8 | 2,1 | 2,3 | 2,5 | 2,7 | 2,0 | 0,9 | 0,2 | 0,4 | 0,6 | 0,8 | 0,1 | 0,3 | 0,5 | 0,7 | 0,0 |
| 2,9 | 2,2 | 2,4 | 2,6 | 2,8 | 2,1 | 2,3 | 2,5 | 2,7 | 2,0 | 0,9 | 0,2 | 0,4 | 0,6 | 0,8 | 0,1 | 0,3 | 0,5 | 0,7 | 0,0 | 1,9 | 1,2 | 1,4 | 1,6 | 1,8 | 1,1 | 1,3 | 1,5 | 1,7 | 1,0 |

**Fig. 5.** Example of the reduction for $n = 3$. Initial embedding of $L$ into the partial sudoku $S$.

In the end, the aim of the construction is obtaining a partial sudoku with column hole pattern, that is, where columns with parameter $l = 0$ are empty. To this purpose, the construction goes on appropriately swapping the contents of some cells while still preserving the property of being a partial sudoku. We proceed as follows:

- Consider every non-empty cell $S_{0,j}^{k,0}$ (that is, the ones placed on the first columns of each region in the first region row). Notice that these cells are the ones that have inherited the information in the partial latin square, $S_{0,j}^{k,0} = (L_{k,j}, 0)$.
- For each $S_{0,j}^{k,0}$ that is different to $\perp$, we will take one of its *mates*, that is, another cell in the same row, $S_{0,j}^{k,l}$, $l \neq 0$, such that there exists $k'$ for which it holds that $S_{0,j}^{k,0} = S_{l,j}^{k',l}$ and $S_{0,j}^{k,l} = S_{l,j}^{k',0}$ (the opposite corners of the rectangle defined by these positions coincide, see Fig. 4). In fact, Proposition 3 shows that there always exist exactly $n$ mates for each non-empty cell.
- Then, for each $S_{0,j}^{k,0}$, we take one of its mates, and swap their contents, and we also swap the contents of the cells in the other two corners of the rectangle (that is $S_{l,j}^{k',0}$ and $S_{l,j}^{k',l}$). In the case that there exist two cells $S_{0,j}^{k,0}$ and $S_{0,j'}^{k',0}$ with the same value, one should take care in choosing mates with different contents. Propositions 3 and 4 show that it is always possible.
- Then, in the last step of the construction, the first column of each column region is emptied, namely those positions with $l = 0$.

**Proposition 3.** *For every j and k satisfying* $0 \leq j, k \leq n-1$, *every non-empty cell* $S_{0,j}^{k,0}$ *has exactly n mates.*

**Proof.** Let us fix the values $j$ and $k$. From the construction of $S$ it follows that $S_{0,j}^{k,0} = (L_{k,j}, 0)$. Hence, the cells that can be located at its opposite corner have to be assigned the value 0 in its second component, which are, by construction, $S_{l,j}^{k',l}$, for $0 \leq k' \leq n-1$ and $1 \leq l \leq n^2$.

Notice that the content of the $n$ cells $S_{l,j}^{k',l}$ is $(j + k' \pmod n, 0)$, for $0 \leq k' \leq n-1$. Hence, $S_{0,j}^{k,0}$ will be equal to $S_{l,j}^{k',l}$ for $k' \equiv L_{k,j} - j \pmod n$.

We are looking for the existence of values for $l$ such that verify that the two other corners coincide, that is $S_{l,j}^{k',0} = S_{0,j}^{k,l}$. The values of these cells are

$$S_{l,j}^{k',0} = (l + j + k' \pmod n, C_{l,0}) = (l + j + k' \pmod n, l)$$

$$S_{0,j}^{k,l} = (j + k \pmod n, C_{0,l}) = (j + k \pmod n, l).$$

So, equality will hold if, and only if,

$$l + j + k' \equiv j + k \pmod n \iff l \equiv k - k' \pmod n \equiv j + k - L_{k,j} \pmod n.$$

It straightforwardly follows that there exist $n^2/n = n$ values for $l$ that satisfy this condition, which determine the positions. □

**Proposition 4.** *Let* $S_{0,j_1}^{k_1,0}, S_{0,j_2}^{k_2,0}, \ldots, S_{0,j_t}^{k_t,0}$ *be the entries in S corresponding to non-empty cells in L. Then it is possible to choose mates with pairwise different contents for these entries.*

**Proof.** Firstly, notice that the contents of the possible mates for two different values $S_{0,j}^{k,0} \neq S_{0,j'}^{k',0}$ are disjoint. The first component of the mates content will be $j+k \pmod n$ and $j'+k' \pmod n$, respectively. If these values are different, obviously, the sets of possible mates contents will also be. In the case where $j+k \equiv j'+k' \pmod n$, the sets of mates will be in columns $l \equiv j + k - L_{j,k} \pmod n$ and $l' \equiv j' + k' - L_{j',k'} \pmod n$, respectively, so they will also have disjoint contents.

Hence, the problem could appear for cells with the same value, namely $S_{0,j}^{k,0} = S_{0,j'}^{k',0}$. Again, when $j + k \not\equiv j' + k' \pmod n$, the contents of their possible mates will be disjoint. In the case that $j+k \equiv j'+k' \pmod n$, their set of mates will be located at columns $l \equiv j+k+L_{k,j} \equiv j'+k'+L_{k',j'} \pmod n$. In this last case, the repeated value can appear at most $n$ times, but from Proposition 3 it follows that there are $n$ mates for each value with different contents, so we can select mates with pairwise different contents for all the occurrences of the same value. □

Finally, the following proposition shows that the latin square completion problem can be reduced to the sudoku completion problem with column hole pattern.

**Theorem 5.** *Let L be a partial latin square, and let T be the partial sudoku obtained by the previously detailed construction. The partial latin square L can be completed if, and only if, the partial sudoku T can be completed.*

**Proof.** On the one hand, notice that if a completion of $L$ exists, a completion of $T$ can also be obtained: originally non-empty cells located at $l = 0$ can be assigned the values obtained in the construction of $T$, before emptying the columns, and those originally empty cells (located at $i = l = 0$) can be assigned the values of the solution of $L$, along with second component 0.

Conversely, if a completion of $T$ exists, each empty cell $L_{k,j}$ can be assigned the first component of $T^{k,0}_{0,j}$. Firstly, notice that in the completion for any originally non-empty cell $T^{k,0}_{0,j}$ the only possible content is the one from its chosen mate (since for all such cells their mates are taken with pairwise different contents). So, a completion of $T$ cannot encode a completion for a different partial latin square $L'$. Analogously, for an originally empty cell $T^{k,0}_{0,j}$ the only possible contents are the values $(v, 0)$ for $v$ not appearing in row $k$ of $L$ (since these values appear embedded in some cell $T^{k,l'}_{0,j'}$) and not appearing in column $j$ of $L$ (since these values appear embedded in some cell $T^{k',l'}_{0,j}$). So, in a completion of $T$ the contents of the originally empty cells $T^{k,0}_{0,j}$ can only encode a completion of the input partial latin square $L$, if such a completion exists. □

**Theorem 6.** *The sudoku completion problem with column hole pattern is NP-complete.*

**Proof.** From the previous result, it follows that we can reduce the latin square completion problem to the sudoku completion problem with column hole pattern. Since the reduction works in polynomial time and given that the latin square completion problem is NP-complete, the theorem straightforwardly follows. □

### *3.3. Example of the construction*

Finally, to illustrate the construction introduced, consider its application to the particular partial latin square $L$ shown at the top of Fig. 5, that is:

| | | |
|---|---|---|
| 2 | | 0 |
| 1 | 0 | |
| 0 | 1 | 2 |

Following the construction of the partial sudoku $S$, it can be seen that, for instance, the mates of $S^{1,0}_{0,1} = (0, 0)$ will be located at positions

$$l \equiv j + k - L_{k,j} \equiv 1 + 1 - 0 \equiv 2 \pmod 3 \Longrightarrow l = 2, 5, 8.$$

So, the contents of the possible mates are $S^{1,2}_{0,1} = (2, 2)$, $S^{1,5}_{0,1} = (2, 5)$ and $S^{1,8}_{0,1} = (2, 8)$. We take one of its three mates, for instance the one with $l = 5$ (depicted in dark gray in Fig. 5). Following the same procedure we can select a mate for each $S^{k,0}_{0,j} \neq \perp$.

Proposition 4 shows that the selection can be done such that the contents of the mates are pairwise different. For instance, in the example, the contents of the possible mates of $S^{2,0}_{0,0} = (0, 0)$, $S^{1,0}_{0,1} = (0, 0)$ and $S^{0,0}_{0,2} = (0, 0)$ coincide: $(2, 2)$, $(2, 5)$ and $(2, 8)$. So, since there are three possibilities, a different one can be taken for each. For instance, the selected mates in Fig. 5 correspond to $S^{2,2}_{0,0} = (2, 2)$, $S^{1,5}_{0,1} = (2, 5)$ and $S^{0,8}_{0,2} = (2, 8)$ (marked in dark gray in the figure).

Then, the content of each $S^{k,0}_{0,j} \neq \perp$ can be swapped with the content of its selected mate, while maintaining the property of being a partial sudoku. Finally, the first column of each column region is emptied as shown in Fig. 6. Since the partial latin square of this example has no completion, the resulting partial sudoku has also no completion.

Notice that this would not be necessarily true if the contents of the selected mates were not pairwise different. Consider what happens if for the occurrences of the symbol $(0, 0)$ we have in $S^{1,0}_{0,1}$ and $S^{0,0}_{0,2}$ we take mates with the same content, say for example mates $S^{1,5}_{0,1} = (2, 5)$ and $S^{0,5}_{0,2} = (2, 5)$. So, after we swap the symbols and we empty the first column of each region column the symbol $(2, 5)$ disappears from $S^{1,0}_{0,1}$ and $S^{0,0}_{0,2}$. Then, it turns out that now in a completion we can change the positions assigned to these two occurrences of $(2, 5)$ to positions which are different from their original ones before the columns were emptied. In particular, $(2, 5)$ can be assigned to $S^{0,0}_{0,1}$ and $S^{1,0}_{0,2}$. So, we can finally get a completion that corresponds to this different partial latin square $L'$, that has a solution:

| | | |
|---|---|---|
| 2 | 0 | |
| 1 | | 0 |
| 0 | 1 | 2 |

That is, if the selected mates for the non-empty cells are not taken with pairwise different contents, the resulting partial sudoku could have solutions that encode solutions for a different partial latin square $L'$.

## Acknowledgments

The research was partially supported by the Spanish CICYT, MICINN and MINECO Projects MTM2010-21580-C02-01, ARINF (TIN2009-14704-C03-01), and TASSAT (TIN2010-20967-C04-03). The authors would like to thank the referees for their helpful comments that helped to improve the paper significantly.

Partial latin square $L$:

| | | |
|---|---|---|
| 2 | | 0 |
| 1 | 0 | |
| 0 | 1 | 2 |

Partial Sudoku $S$:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0,1 | 0,2 | 0,3 | 2,0 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | | 2,1 | 2,2 | 2,3 | 2,4 | 2,5 | 2,6 | 2,7 | 0,0 | 2,9 |
| | 1,1 | 1,2 | 1,0 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | | 2,1 | 2,2 | 2,3 | 2,4 | 0,0 | 2,6 | 2,7 | 2,8 | 2,9 | | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 |
| | 2,1 | 0,0 | 2,3 | 2,4 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | | 0,1 | 1,0 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | | 1,1 | 2,0 | 1,3 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 |
| | 0,0 | 1,3 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 1,2 | | 1,0 | 2,3 | 2,4 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | 2,2 | | 2,0 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 0,2 |
| | 1,0 | 2,3 | 2,4 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | 2,2 | | 2,0 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 0,2 | | 0,0 | 1,3 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 1,2 |
| | 2,0 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 0,2 | | 0,0 | 1,3 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 1,2 | | 1,0 | 2,3 | 2,4 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | 2,2 |
| | 2,3 | 2,2 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | 2,1 | 2,4 | | 0,3 | 0,2 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 0,1 | 0,4 | | 1,3 | 1,2 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 1,1 | 1,4 |
| | 0,3 | 1,0 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 0,1 | 0,4 | | 1,3 | 2,0 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 1,1 | 1,4 | | 2,3 | 0,0 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | 2,1 | 2,4 |
| | 1,3 | 2,0 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 1,1 | 1,4 | | 2,3 | 0,0 | 2,5 | 2,6 | 2,7 | 2,8 | 2,9 | 2,1 | 2,4 | | 0,3 | 1,0 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 0,1 | 0,4 |
| | 0,4 | 0,5 | 0,0 | 0,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,6 | | 1,4 | 1,5 | 1,0 | 1,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,6 | | 2,4 | 2,5 | 2,0 | 2,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,6 |
| | 1,4 | 1,5 | 1,3 | 1,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,6 | | 2,4 | 2,5 | 2,0 | 2,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,6 | | 0,4 | 0,5 | 0,0 | 0,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,6 |
| | 2,4 | 2,5 | 2,0 | 2,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,6 | | 0,4 | 0,5 | 0,0 | 0,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,6 | | 1,4 | 1,5 | 1,0 | 1,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,6 |
| | 1,5 | 1,6 | 1,7 | 0,0 | 1,9 | 1,1 | 1,2 | 1,3 | 1,8 | | 2,5 | 2,6 | 2,7 | 1,0 | 2,9 | 2,1 | 2,2 | 2,3 | 2,8 | | 0,5 | 0,6 | 0,7 | 2,0 | 0,9 | 0,1 | 0,2 | 0,3 | 0,8 |
| | 2,5 | 2,6 | 2,7 | 1,0 | 2,9 | 2,1 | 2,2 | 2,3 | 2,8 | | 0,5 | 0,6 | 0,7 | 2,0 | 0,9 | 0,1 | 0,2 | 0,3 | 0,8 | | 1,5 | 1,6 | 1,7 | 0,0 | 1,9 | 1,1 | 1,2 | 1,3 | 1,8 |
| | 0,5 | 0,6 | 0,7 | 0,4 | 0,9 | 0,1 | 0,2 | 0,3 | 0,8 | | 1,5 | 1,6 | 1,7 | 0,0 | 1,9 | 1,1 | 1,2 | 1,3 | 1,8 | | 2,5 | 2,6 | 2,7 | 1,0 | 2,9 | 2,1 | 2,2 | 2,3 | 2,8 |
| | 2,6 | 2,7 | 2,8 | 2,9 | 0,0 | 2,2 | 2,3 | 2,4 | 2,1 | | 0,6 | 0,7 | 0,8 | 0,9 | 1,0 | 0,2 | 0,3 | 0,4 | 0,1 | | 1,6 | 1,7 | 1,8 | 1,9 | 2,0 | 1,2 | 1,3 | 1,4 | 1,1 |
| | 0,6 | 0,7 | 0,8 | 0,9 | 1,0 | 0,2 | 0,3 | 0,4 | 0,1 | | 1,6 | 1,7 | 1,8 | 1,9 | 2,0 | 1,2 | 1,3 | 1,4 | 1,1 | | 2,6 | 2,7 | 2,8 | 2,9 | 0,0 | 2,2 | 2,3 | 2,4 | 2,1 |
| | 1,6 | 1,7 | 1,8 | 1,9 | 2,0 | 1,2 | 1,3 | 1,4 | 1,1 | | 2,6 | 2,7 | 2,8 | 2,9 | 2,5 | 2,2 | 2,3 | 2,4 | 2,1 | | 0,6 | 0,7 | 0,8 | 0,9 | 1,0 | 0,2 | 0,3 | 0,4 | 0,1 |
| | 0,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,0 | 0,4 | 0,5 | 0,3 | | 1,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,0 | 1,4 | 1,5 | 1,3 | | 2,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,0 | 2,4 | 2,5 | 2,3 |
| | 1,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,0 | 1,4 | 1,5 | 1,3 | | 2,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,0 | 2,4 | 2,5 | 2,3 | | 0,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,0 | 0,4 | 0,5 | 0,3 |
| | 2,7 | 2,8 | 2,9 | 2,1 | 2,2 | 2,0 | 2,4 | 2,5 | 2,3 | | 0,7 | 0,8 | 0,9 | 0,1 | 0,2 | 0,0 | 0,4 | 0,5 | 0,3 | | 1,7 | 1,8 | 1,9 | 1,1 | 1,2 | 1,0 | 1,4 | 1,5 | 1,3 |
| | 1,8 | 1,9 | 1,1 | 1,2 | 1,3 | 1,4 | 0,0 | 1,6 | 1,5 | | 2,8 | 2,9 | 2,1 | 2,2 | 2,3 | 2,4 | 1,0 | 2,6 | 2,5 | | 0,8 | 0,9 | 0,1 | 0,2 | 0,3 | 0,4 | 2,0 | 0,6 | 0,5 |
| | 2,8 | 2,9 | 2,1 | 2,2 | 2,3 | 2,4 | 1,0 | 2,6 | 2,5 | | 0,8 | 0,9 | 0,1 | 0,2 | 0,3 | 0,4 | 2,0 | 0,6 | 0,5 | | 1,8 | 1,9 | 1,1 | 1,2 | 1,3 | 1,4 | 0,0 | 1,6 | 1,5 |
| | 0,8 | 0,9 | 0,1 | 0,2 | 0,3 | 0,4 | 2,0 | 0,6 | 0,5 | | 1,8 | 1,9 | 1,1 | 1,2 | 1,3 | 1,4 | 0,0 | 1,6 | 1,5 | | 2,8 | 2,9 | 2,1 | 2,2 | 2,3 | 2,4 | 1,0 | 2,6 | 2,5 |
| | 2,9 | 2,1 | 2,2 | 2,3 | 2,4 | 2,5 | 2,6 | 0,0 | 2,7 | | 0,9 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 1,0 | 0,7 | | 1,9 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 | 2,0 | 1,7 |
| | 0,9 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 1,0 | 0,7 | | 1,9 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 | 2,0 | 1,7 | | 2,9 | 2,1 | 2,2 | 2,3 | 2,4 | 2,5 | 2,6 | 2,8 | 2,7 |
| | 1,9 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 | 2,0 | 1,7 | | 2,9 | 2,1 | 2,2 | 2,3 | 2,4 | 2,5 | 2,6 | 0,0 | 2,7 | | 0,9 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 1,0 | 0,7 |
| | 0,2 | 0,4 | 0,6 | 0,8 | 0,1 | 0,3 | 0,5 | 0,7 | 0,0 | | 1,2 | 1,4 | 1,6 | 1,8 | 1,1 | 1,3 | 1,5 | 1,7 | 1,0 | | 2,2 | 2,4 | 2,6 | 2,8 | 2,1 | 2,3 | 2,5 | 2,7 | 2,0 |
| | 1,2 | 1,4 | 1,6 | 1,8 | 1,1 | 1,3 | 1,5 | 1,7 | 1,0 | | 2,2 | 2,4 | 2,6 | 2,8 | 2,1 | 2,3 | 2,5 | 2,7 | 2,0 | | 0,2 | 0,4 | 0,6 | 0,8 | 0,1 | 0,3 | 0,5 | 0,7 | 0,0 |
| | 2,2 | 2,4 | 2,6 | 2,8 | 2,1 | 2,3 | 2,5 | 2,7 | 2,0 | | 0,2 | 0,4 | 0,6 | 0,8 | 0,1 | 0,3 | 0,5 | 0,7 | 0,0 | | 1,2 | 1,4 | 1,6 | 1,8 | 1,1 | 1,3 | 1,5 | 1,7 | 1,0 |

**Fig. 6.** Example of the reduction for $n = 3$. Resulting construction after swapping the corresponding cells and finally emptying the first column of each region column.

## References

[1] D. Achlioptas, C. Gomes, H. Kautz, B. Selman, Generating satisfiable problem instances, in: Proc. of National Conference on Artificial Intelligence, AAAI'00, 2000, pp. 193–200.
[2] C. Ansótegui, R. Béjar, C. Fernández, C.P. Gomes, C. Mateu, The impact of balancing on problem hardness in a highly structured domain, in: Twenty-First National Conference on Artificial Intelligence, AAAI'06, 2006, pp. 438–443.
[3] C. Ansótegui, A. del Val, I. Dotú, C. Fernández, F. Manyà, Modelling choices in quasigroup completion: SAT vs. CSP, in: Proc. of the National Conference on Artificial Intelligence, AAAI-04, AAAI Press, 2004, pp. 137–142.
[4] R.A. Bailey, P.J. Cameron, R. Connelly, Sudoku, gerechte designs, resolutions, affine space, spreads, reguli, and Hamming codes, American Mathematical Monthly 115 (2008) 383–404.
[5] R.A. Bailey, J. Kunert, R.J. Martin, Some comments on gerechte designs, Journal of Agronomy and Crop Science 165 (1990) 121–130.
[6] D. Bryant, Completing partial commutative quasigroups constructed from partial Steiner triple systems is NP-complete, Discrete Mathematics 309 (14) (2009) 4700–4704.
[7] A.G. Chetwynd, S.J. Rhodes, Chessboard squares, Discrete Mathematics 141 (1–3) (1995) 47–59.
[8] C. Colbourn, The complexity of completing partial Latin squares, Discrete Applied Mathematics 8 (1984) 151–158.
[9] T. Easton, R.G. Parker, On completing Latin squares, Discrete Applied Mathematics 113 (2–3) (2001) 167–181.
[10] C. Gomes, B. Selman, Problem structure in the presence of perturbations, in: Proceedings of the Fourteenth National Conference on Artificial Intelligence, AAAI'97, AAAI Press, New Providence, RI, 1997, pp. 221–227.
[11] C.P. Gomes, B. Selman, N. Crato, Heavy-tailed distributions in combinatorial search, in: Proceedings of the Third International Conference of Constraint Programming, CP'97, Springer-Verlag, Linz, Austria, 1997, pp. 121–135.
[12] M.T. Jacobson, P. Matthews, Generating uniformly distributed random Latin squares, Journal of Combinatorial Designs 4 (1996) 405–437.
[13] H. Kautz, Y. Ruan, D. Achlioptas, C. Gomes, B. Selman, M. Stickel, Balance and filtering in structured satisfiable problems, in: Proc. of International Conference on Artificial Intelligence, JCAI'01, 2001, pp. 193–200.
[14] I. Lynce, J. Ouaknine, Sudoku as a SAT problem, in: Proc. of Ninth International Symposium on Artificial Intelligence and Mathematics, ISAIM-06, 2006.
[15] H. Simonis, Sudoku as a constraint problem, in: Proc. of Fourth International Workshop on Modelling and Reformulating Constraint Satisfaction Problems, in CP-2005, 2005, pp. 13–27.
[16] T. Yato, T. Seta, Complexity and completeness of finding another solution and its application to puzzles, in: Proc. of National Meeting of the Information Processing Society of Japan, IPSJ, 2002, pp. 1052–1060.