Extremely Complex 4-Colored Rectangle-Free Grids: Solution of Open Multiple-Valued Problems

Bernd Steinbach
Institute of Computer Science
Freiberg University of Mining and Technology
Freiberg, Germany
Email: steinb@informatik.tu-freiberg.de

Christian Posthoff
Department of Computing and Information Technology
The University of the West Indies
Trinidad & Tobago
Email: christian@posthoff.de

***Abstract*—This paper aims at the rectangle-free coloring of grids using four colors. It has been proven in a well developed theory that there is an upper bound of rectangle-free 4-colorable grids as well as a lower bound of grids for which no rectangle-free color pattern of four colors exist. Between these tight bounds the grids of the size 17×17, 17×18, 18×17, and 18×18 are located for which it is not known until now whether a rectangle-free coloring by four colors exists. We present in this paper an approach that solves all these open problems.**

From another point of view this paper aims at the solution of a multiple-valued problem having an extremely high complexity. There are $1.16798 * 10^{195}$ different grids of four colors. It must be detected whether at least one of this hardly imaginable large number of patterns satisfies strong additional conditions. In order to solve this highly complex problem, several approaches were taken into account to find out properties of the problem which finally allowed us to calculate the solution.

***Keywords*-four-valued coloring, rectangle-free grid, Boolean equation, SAT-solver, XBOOLE.**

I. Introduction

There are many practical tasks which can be modeled and solved by graph coloring [8]. Such tasks are, for instance, the frequency assignment of radio stations, the aircraft scheduling to flights or the state assignment for optimizing finite state machines. Graph coloring can be done by assigning colors either to the vertices or to the edges of a given graph. We focus in this paper to the edge coloring.

There are different data structures that represent a graph. We select the adjacency matrix, where the rows represent source vertices and columns represent destination vertices. Three further properties specify the studied graph:

1) each source vertex is connected with each destination vertex by an edge,
2) each edge is colored by exactly one of four colors,
3) it is not allowed, that all edges of any quadruple of edges $\{e(s_1, d_1), e(s_1, d_2), e(s_2, d_1), e(s_2, d_2)\}$ are colored by the same color.

These three conditions means in terms of the used data structure that all positions of the adjacency matrix must be colored by one of four colors, and it is not allowed that in the four cross points of any pair of rows and any pair of columns the same color appears. It should be noted that due to the third property mentioned above a tight relationship to bipartite Ramsey numbers [6] exists.

A comprehensive theory for such a grid coloring with regard to several fixed numbers of colors is published in [3]. In this paper the problem is shortly defined as follows. "A two-dimensional *grid* is a set $G_{n,m} = [n] \times [m]$. A grid $G_{n,m}$ is *c-colorable* if there is a function $\chi_{n,m} : G_{n,m} \rightarrow [c]$ such that there are no rectangles with all four corners the same color."

Due to some Theorems it is known that 4-colorable grids up to certain sizes of the grid exist. Some other Theorems give lower bounds of grids, which are not 4-colorable. From this theory follows that there must be a 4-colorable grid $G_{16,20}$, however, an example for such a grid was not shown. According to the best of our knowledge only one grid $G_{16,17}$ is known that is almost 4-colorable with the exception of a single position. This grid was constructed by the student Rohan Puttagunta. This grid can be restricted to a 4-colorable grid $G_{16,16}$ that looks like a confused mixture of 4 colors. We will give in this paper well structured 4-colorable grids $G_{16,16}$ and $G_{16,20}$.

Until now it is unknown whether there is a 4-colorable grid of the size 17×17, 17×18, 18×17, or 18×18. Many scientists failed to solve this problem using a wide range of approaches [4]. We will answer this question in this paper. From the first point of view this task seems to be easy. It is only necessary to fix 18×18 colors on the grid $G_{18,18}$ and check whether all possible rectangles do not have the same color on their corners. However, when our first check fails we have to take into account all other color patterns. The grid $G_{18,18}$ has $18 * 18 = 324$ positions and one of four colors must be selected for each of them. Hence, there are $4^{324} = 1.16798 * 10^{195}$ different patterns in which one of four colors is assigned to each of the 18×18 positions of the grid. Assume we are able to evaluate one pattern in one nano-second (10^{-9} seconds) and we spend 100 years ($3 * 10^9$ seconds) then we must repeat the job $3.8 * 10^{176}$ times in order to know whether there is an allowed color pattern and

if YES which valid color pattern exists. So we see that we are going to solve an extremely complex problem.

The rest of the paper is organized as follows. In section 2 we introduce both a 4-valued model and a deduced Boolean model of the 4-color problem of grids. The reflection of some basic approaches in section 3 reveals important properties and strong limits of the problem. In Section 4 we present both steps to the final solution and intermediate results before we conclude the paper.

II. Model of 4-colorable Grids

A. 4-valued Model

The four colors can be represented by the four values $1, 2, 3, 4$. The value of the grid in the row r and the column c can be modeled by the 4-valued variable $x_{r,c}$. One rectangle of the grid is selected by the rows r_i and r_j and by the columns c_k and c_l. The color condition for the rectangles can be described using the following three operations:

1) *equal (multiple-valued):*

$$x \equiv y = \begin{cases} 1 & \text{if } x \text{ is equal to } y \\ 0 & \text{otherwise} \end{cases} , \quad (1)$$

2) *and (conjunction, binary):*

$$x \wedge y = \begin{cases} 1 & \text{if both } x \text{ and } y \text{ are equal to 1} \\ 0 & \text{otherwise} \end{cases} , \quad (2)$$

3) *or (disjunction, binary):*

$$x \vee y = \begin{cases} 0 & \text{if both } x \text{ and } y \text{ are equal to 0} \\ 1 & \text{otherwise} \end{cases} . \quad (3)$$

The function $f_{ec}(x_{r_i,c_k}, x_{r_i,c_l}, x_{r_j,c_k}, x_{r_j,c_l})$ (4) depends on four 4-valued variables and has a Boolean result that is true in the case that the colors in the corners of the rectangle selected by the rows r_i and r_j and by the columns c_k and c_l are equal to each other.

$$\begin{aligned} &f_{ec}(x_{r_i,c_k}, x_{r_i,c_l}, x_{r_j,c_k}, x_{r_j,c_l}) = \\ &((x_{r_i,c_k} \equiv 1) \wedge (x_{r_i,c_l} \equiv 1) \wedge (x_{r_j,c_k} \equiv 1) \wedge (x_{r_j,c_l} \equiv 1)) \vee \\ &((x_{r_i,c_k} \equiv 2) \wedge (x_{r_i,c_l} \equiv 2) \wedge (x_{r_j,c_k} \equiv 2) \wedge (x_{r_j,c_l} \equiv 2)) \vee \\ &((x_{r_i,c_k} \equiv 3) \wedge (x_{r_i,c_l} \equiv 3) \wedge (x_{r_j,c_k} \equiv 3) \wedge (x_{r_j,c_l} \equiv 3)) \vee \\ &((x_{r_i,c_k} \equiv 4) \wedge (x_{r_i,c_l} \equiv 4) \wedge (x_{r_j,c_k} \equiv 4) \wedge (x_{r_j,c_l} \equiv 4)) \end{aligned} \quad (4)$$

The condition that in the four corners of the rectangle selected by the rows r_i and r_j and by the columns c_k and c_l not only one of the four colors $1, 2, 3, 4$ will appear is

$$f_{ec}(x_{r_i,c_k}, x_{r_i,c_l}, x_{r_j,c_k}, x_{r_j,c_l}) = 0 \ . \quad (5)$$

For the whole grid $G_{n,m}$ we have the condition:

$$\bigvee_{i=1}^{n-1} \bigvee_{j=i+1}^{n} \bigvee_{k=1}^{m-1} \bigvee_{l=k+1}^{m} f_{ec}(x_{r_i,c_k}, x_{r_i,c_l}, x_{r_j,c_k}, x_{r_j,c_l}) = 0 \ . \quad (6)$$

Table I
Mapping of 4-valued color x to 2 Boolean variables a and b

x	a	b
1	0	0
2	1	0
3	0	1
4	1	1

B. Binary Model

Internally the computer works with binary values. Hence, the next step of preparations is the mapping of the model into the Boolean space. The four values of a single color can be expressed by two Boolean values. Table I shows the used mapping.

The mapping of the model into the Boolean domain requires a doubling of the number of variables but allows to skip the special comparison operation. The function (7) depends on eight Boolean variables and has a Boolean result that is true in the case that the colors in the corners of the rectangle selected by the rows r_i and r_j and by the columns c_k and c_l are equal to each other.

$$\begin{aligned} &f_{ecb}(a_{r_i,c_k}, b_{r_i,c_k}, a_{r_i,c_l}, b_{r_i,c_l}, a_{r_j,c_k}, b_{r_j,c_k}, a_{r_j,c_l}, b_{r_j,c_l}) = \\ &\quad (\overline{a}_{r_i,c_k} \wedge \overline{b}_{r_i,c_k} \wedge \overline{a}_{r_i,c_l} \wedge \overline{b}_{r_i,c_l} \wedge \\ &\qquad \overline{a}_{r_j,c_k} \wedge \overline{b}_{r_j,c_k} \wedge \overline{a}_{r_j,c_l} \wedge \overline{b}_{r_j,c_l}) \vee \\ &\quad (a_{r_i,c_k} \wedge \overline{b}_{r_i,c_k} \wedge a_{r_i,c_l} \wedge \overline{b}_{r_i,c_l} \wedge \\ &\qquad a_{r_j,c_k} \wedge \overline{b}_{r_j,c_k} \wedge a_{r_j,c_l} \wedge \overline{b}_{r_j,c_l}) \vee \\ &\quad (\overline{a}_{r_i,c_k} \wedge b_{r_i,c_k} \wedge \overline{a}_{r_i,c_l} \wedge b_{r_i,c_l} \wedge \\ &\qquad \overline{a}_{r_j,c_k} \wedge b_{r_j,c_k} \wedge \overline{a}_{r_j,c_l} \wedge b_{r_j,c_l}) \vee \\ &\quad (a_{r_i,c_k} \wedge b_{r_i,c_k} \wedge a_{r_i,c_l} \wedge b_{r_i,c_l} \wedge \\ &\qquad a_{r_j,c_k} \wedge b_{r_j,c_k} \wedge a_{r_j,c_l} \wedge b_{r_j,c_l}) \end{aligned} \quad (7)$$

The conditions of the 4-color problem on a grid $G_{n,m}$ are achieved when the function f_{ecb} (7) is equal to 0 for all rectangles which can be expressed by

$$\begin{aligned} \bigvee_{i=1}^{n-1} \bigvee_{j=i+1}^{n} \bigvee_{k=1}^{m-1} \bigvee_{l=k+1}^{m} & f_{ecb}(a_{r_i,c_k}, b_{r_i,c_k}, \\ & a_{r_i,c_l}, b_{r_i,c_l}, a_{r_j,c_k}, b_{r_j,c_k}, a_{r_j,c_l}, b_{r_j,c_l}) = 0 \ . \end{aligned} \quad (8)$$

III. Basic Approaches and Results

A. Solving Boolean Equations

In order to solve the strong complex coloring problem we need deep knowledge of its properties. The details of our basic exploration are published in [10]. Here we summarize the main results.

The Boolean equation (8) can be solved for small grid sizes completely. The representation by ternary vectors of XBOOLE [9], [11] and [13] helps to restrict the needed memory.

Table II
SOLUTIONS OF THE BOOLEAN EQUATION (8)

r	c	v	TV	solutions	forbidden	ratio
2	2	8	24	252	4	1.56
3	2	12	304	3,912	184	4.49
4	2	16	3,416	59,928	5,608	8.56
5	2	20	36,736	906,912	141,664	13.51
6	2	24	387,840	13,571,712	3,205,504	19.11
7	2	28	4,061,824	201,014,784	67,420,672	25.12

Table II shows the detailed results for grids of two to seven rows (labeled by r) and of two columns (labeled by c). The column labeled by v gives the number of Boolean variables of the equation (8). The column labeled by TV enumerates the number of ternary vectors needed to express the solutions given in the next column. The benefit of the ternary representation is obvious. Because XBOOLE uses orthogonal ternary vectors it is easy to calculate all solutions.

The number of all possible color patterns is defined by 4 to the power of the product of rows and columns 4^{r*c} which is equal to the power of 2 to the number of variables 2^v. The difference between 2^v and the number of solutions is equal to the number of forbidden color patterns. The column ratio in Table II gives the percentage of the number of forbidden patterns divided by all 2^v possible color patterns.

For the simplest grid $G_{2,2}$ almost all of the $2^8 = 256$ color patterns are allowed solutions. Only the four patterns specified by the function (7) are forbidden. The forbidden fraction of 4-color patterns of $G_{2,2}$ is only 1.56%. This ratio grows to more than 25% for the grid $G_{7,2}$. That means that the number of forbidden patterns grows stronger than the number all possible 4-color patterns by enlarging the size of the grid. From this observation we learn that there must be a 4-colorable grid $G_{n,m}$ with the property that at least one of the grids $G_{n+1,m}$ or $G_{n,m+1}$ is not 4-colorable. This boarder is reached for a 4-colorable grid $G_{n,m}$ with a ratio of forbidden patterns of 0.75. The practical results of Table II confirm the theory of [3] that forbidden color patterns grow even stronger than the exponential growth of the number of possible color patterns for growing numbers of rows and columns of the grid.

B. Exploit Permutations of Colors, Rows, and Columns

The limit in the previous approach was the required memory of about 800 Megabytes to represent the solution of the grid $G_{7,2}$ which could be calculated in less than 5 seconds. To break the limitation of memory requirements we exploited some heuristic properties of the problem:

1) Knowing one single solution of the 4-color problem, 4! = 24 permutations of this solution with regard to the four colors are also solutions.
2) The permutation of rows and columns of a given solution pattern creates another pattern that is a valid solution, too.

Table III
SELECTED SOLUTIONS OF THE BOOLEAN EQUATION (8) WITH FIXED UNIFORM DISTRIBUTION OF THE COLORS IN THE TOP ROW AND IN THE LEFT COLUMN

r	c	v	TV	solutions
2	2	8	1	4
2	12	48	6,912	2,361,960
3	2	12	1	16
3	8	48	4,616,388	4,616,388
4	2	16	1	64
4	5	40	674,264	12,870,096
5	2	20	1	256
5	4	40	573,508	12,870,096
6	2	24	4	960
6	3	36	15,928	797,544
7	2	28	16	3,600
7	3	42	183,152	104,93,136
8	2	32	64	3,600
8	3	48	2,152,819	136,603,152
9	2	36	256	50,625
10	2	40	768	182,250
15	2	60	147,456	104,162,436
19	2	76	6,553,600	14,999,390,784

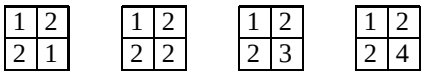

Figure 1. The selected 4-colored grids $G_{2,2}$ of the first row in Table III

3) A nearly uniformly distribution of the colors in both the rows and the columns is given for the largest number of allowed 4-colored grids.

Hence, in [10] we applied a heuristic which restricts to the calculation of such solutions having a single fixed uniform distribution of the colors in the top row and in the left column. We restrict in this experiment the calculation to 12 columns and 2 Gigabytes of available memory.

Table III shows selected results of the explained restricted calculation repeated from [10]. Figure 1 depicts the four selected 4-colored grids of the first row in Table III. Using the same memory size, the number of Boolean variables could be enlarged from 28 for $G_{7,2}$ to 76 for $G_{19,2}$. That means, by utilizing properties of the 4-color problem mentioned above, we have solved problems that are $2^{48} = 281,474,976,710,656$ times larger than before, but again the available memory size restricts the solution of larger 4-colorable grids.

C. Exchange of Space and Time

The function (7) describes the forbidden patterns of a single rectangle. All rectangles are specified by all possible pairs of rows and all possible pairs of columns. Hence, there are

$$|rectangle| = \binom{n}{2} * \binom{m}{2} \tag{9}$$

rectangles labeled by `all_rect` in the following program fragments. The conditions of these rectangles must be excluded from all pattern of the Boolean space B^{n*m} which

```
for(i = 0; i < all_rect; i++)
    aps = DIF(aps, f_ecb[i]);
```

Figure 2. Iterative approach with unrestricted space requirements

can be calculated using the DIF-operation of XBOOLE [9], [11], and [13].

The main data structure of XBOOLE is an orthogonal list of ternary vectors (TVL). Each ternary vector consists of n elements '0', '1', and '–' for a Boolean space B^n. Two different binary vectors which are equal to each other in $n-1$ positions can be merged into a single ternary vector that contains a dash element in the position of given different binary values. A ternary vector of d dash elements expresses 2^d binary vectors. Hence, a TVL can be used as a compact representation of a set of binary vectors. The strong benefits in terms of needed memory of TVLs can be seen in Table II. The XBOOLE-operation C=DIF(A,B) calculates the *set difference* $C = A \setminus B = A \cap \neg B$ for the given TVLs A and B such that as much as possible dash elements of A remain in the result C.

We use the DIF-operation of XBOOLE to solve the grid coloring problem. Assume aps is the *actual partial solution* which is initialized by the whole Boolean space B^{n*m}; then the core algorithm of Figure 2 solves the 4-coloring problem when unrestricted memory space can be used.

It is an important drawback of the approach of Figure 2 that the size of aps typically increases extremely up to a certain index i and decreases later on. Therefore we implemented a recursive algorithm that allows to exchange space and time.

If the extremely large size of aps avoids the next iteration step we split aps into aps0 and aps1, solve both subproblems sequentially, and combine both solutions at the end. This approach can be applied recursively as shown in Figure 3. In this algorithm we use additionally the NTV-operation of XBOOLE. The NTV-operation returns the *number of ternary vectors* of the given TVL. The SPLIT-operation splits the given TVL aps approximately in the middle into the TVLs aps0 and aps1. Two stacks with the access operations PUSH and POP are used as storage for the recursive calculation.

We applied the algorithm of Figure 3 to grids of 12 rows, used a fixed value SplitLimit = 400, and canceled the calculation when the first solutions were found.

The approach of exchanging space and time allows again an extreme improvement: solutions for 4-colored grids which are modeled with up to 384 Boolean variables were found instead of 76 variables in the second (already improved) approach. This means that the approach of exchanging space and time for the 4-color grid problems allows solving problems which are $2^{308} = 5.214812 * 10^{92}$ times

```
for(i = 0; i < all_rect; i++) {
    aps = DIF(aps, f_ecb[i]);
    if(NTV(aps) > SplitLimit) {
        SPLIT(aps, aps0, aps1);
        aps_stack.PUSH(aps1);
        position_stack.PUSH(i);
        aps = aps0;
    }
    if(NTV(aps) == 0) {
        aps = aps_stack.POP();
        i = position_stack.POP();
    }
}
```

Figure 3. Recursive approach with restricted space requirements

Table IV
RECURSIVE SEQUENTIAL SOLUTION USING THE ALGORITHM OF FIGURE 3 CANCELED AFTER THE CALCULATION OF THE FIRST SOLUTION SET

r	c	v	TV	solutions	maximal stack size
12	2	48	337	6,620	3
12	3	72	147	2,423	22
12	4	96	319	7,386	30
12	5	120	236	1,188	47
12	6	144	181	1,040	61
12	7	168	231	627	69
12	8	192	109	413	81
12	9	216	72	227	79
12	10	240	34	103	87
12	11	264	40	109	88
12	12	288	112	293	82
12	13	312	51	81	81
12	14	336	82	1,415	97
12	15	360	1	1	80
12	16	384	2	3	81

larger than before.

The time to solve the 4-color grids of Table IV were less than 1 second until 12 columns, less than 3 seconds until 15 columns, and grows to 427 seconds for 16 columns. An approach to reduce the required time is given by parallel computing [14].

IV. STEPS TO SOLVE ALL THE 4-COLORED GRIDS UP TO 18×18

A. Applying SAT-Solver

It is the aim of the satisfiability problem (short SAT) to find at least one assignment of Boolean variables such that a Boolean expression in conjunctive form becomes true. The power of SAT-solvers [2] has improved over the last decades forced by several SAT-competitions. We tried to solve the 4-color grid problem using the best SAT-solvers from the SAT-competitions of the last years. The equation (8) can be easily transformed into a SAT-equation by negation of both sides and the application of de Morgan's law to the Boolean

Table V
TIME TO SOLVE SQUARE 4-COLORED GRIDS USING DIFFERENT SAT-SOLVER

rows columns	v	time in minutes:seconds clasp	lingeling	plingeling	precosat
12	288	0:00.196	0:00.900	0:00.990	0:00.368
13	338	0:00.326	0:01.335	0:04.642	0:00.578
14	392	0:00.559	0:03.940	0:02.073	0:00.578
15	450	46:30.716	54:02.304	73:05.210	120:51.739

2	4	1	3	1	2	4	3	4	4	1	3	2	1	3
1	2	3	1	1	2	3	2	3	4	4	4	4	2	3
4	2	1	2	2	1	1	3	2	3	3	4	3	4	4
3	4	3	4	3	4	1	4	1	3	1	4	1	2	2
1	3	4	4	2	3	1	2	4	2	4	2	3	1	2
2	2	3	3	4	1	2	4	4	1	1	2	3	3	4
4	3	2	2	1	4	2	2	4	1	3	1	4	3	1
3	1	2	3	4	4	1	1	3	4	3	2	2	4	1
1	4	4	3	2	3	2	1	1	3	2	1	4	2	4
4	3	4	3	3	2	3	4	2	4	2	2	1	1	1
4	4	3	1	4	3	2	1	2	1	3	3	2	1	2
3	2	1	1	4	3	3	3	1	2	2	4	2	3	1
2	1	2	1	3	3	4	4	2	2	1	1	4	2	3
1	1	1	2	3	4	4	3	3	1	4	2	1	2	4
2	3	4	2	1	1	4	1	3	3	2	3	1	4	2

Figure 4. 4-colored grid $G_{15,15}$ found by the SAT-solver *clasp* in about 46.5 minutes

expression on the left-hand side. In this way we get the required conjunctive form for the SAT-solver (10).

$$\bigwedge_{i=1}^{n-1}\bigwedge_{j=i+1}^{n}\bigwedge_{k=1}^{m-1}\bigwedge_{l=k+1}^{m}\overline{f_{ecb}(a_{r_i,c_k},b_{r_i,c_k},}$$
$$\overline{a_{r_i,c_l},b_{r_i,c_l},a_{r_j,c_k},b_{r_j,c_k},a_{r_j,c_l},b_{r_j,c_l})}=1\ . \quad (10)$$

Table V shows the required time to find the first solution for square 4-colored grids $G_{12,12}$, $G_{13,13}$, $G_{14,14}$, and $G_{15,15}$ using the SAT-solver *clasp* [5], *lingeling* [1], *plingeling* [1], and *precosat* [1]. Figure 4 shows the 4-colored grid $G_{15,15}$ found by clasp within 46 and a half minutes.

Following the hint of one of the reviewers we tried to use the multiple-valued SAT-solver CAMA [7]. Unfortunately, we get the answer from the authors of [7] that their own created software CAMA does not exist anymore.

From the utilization of the SAT-solvers we learned that:

1) SAT-solver are powerful tools that are able to solve 4-colored grid up to $G_{15,15}$,
2) it is not possible to calculate a 4-colored grid larger than $G_{15,15}$ directly.

The reasons for the second statement are first that the search space for the 4-colored grid $G_{16,16}$ is $2^{62} = 4.61*10^{18}$ times larger than the search space for the 4-colored grid $G_{15,15}$, and second that the fraction of 4-colorable grids is reduced for the larger grid even stronger. We take as base of the time

1	1	1	2	2	2	2	1	3	3	3	4	4	4	4	3
1	3	4	1	2	4	3	2	3	2	1	3	4	1	2	4
4	1	3	1	3	2	4	2	1	3	2	3	2	4	1	4
3	4	1	1	4	3	2	2	2	1	3	3	1	2	4	4
2	2	2	1	1	1	1	2	4	4	4	3	3	3	3	4
2	4	3	2	1	3	4	1	4	1	2	4	3	2	1	3
3	2	4	2	4	1	3	1	2	4	1	4	1	3	2	3
4	3	2	2	3	4	1	1	1	2	4	4	2	1	3	3
3	3	3	4	4	4	4	3	1	1	1	2	2	2	2	1
3	2	1	3	4	1	2	4	1	3	4	1	2	4	3	2
1	3	2	3	2	4	1	4	4	1	3	1	3	2	4	2
2	1	3	3	1	2	4	4	3	4	1	1	4	3	2	2
4	4	4	3	3	3	3	4	2	2	2	1	1	1	1	2
4	1	2	4	3	2	1	3	2	4	3	2	1	3	4	1
2	4	1	4	1	3	2	3	3	2	4	2	4	1	3	1
1	2	4	4	2	1	3	3	4	3	2	2	3	4	1	1

Figure 5. 4-colored grid $G_{16,16}$ constructed using maximal 1-colored sub-grids $G_{4,4}$

measurement the age of our planet Earth which is about 4 billion years ($4*10^9$ years). Based on the measured time to find the first 4-colored grid $G_{15,15}$ and knowing the larger search space it can be estimated that it takes approximately 6,000,000 times the age of the Earth to find a 4-colored grid $G_{16,16}$.

B. Construction of well-structured 4-colored grids $G_{16,16}$ and $G_{16,20}$

The found solution for $G_{15,15}$ of Figure 4 gave us a hint how larger 4-colored grids may be constructed. The evaluation of the rows and columns over the whole grid shows that the colors are nearly uniformly distributed. However, single colors dominate in sub-grids. Due to the 16 positions in rows and columns of $G_{16,16}$ and the four allowed colors sub-grids $G_{4,4}$ are taken into account. From our complete solutions for small grids we know that maximal 9 positions can be occupied by a single color without violation of any condition.

Such maximal 1-colored $G_{4,4}$ grids cannot be repeated in rows or columns, but in diagonal order. In this way, a $G_{8,8}$ grid can be dominated by two colors. In such a structure both 1-dominated $G_{4,4}$ sub-grids can be extended by a single position of color 2. Vice versa both 2-dominated $G_{4,4}$ sub-grids can be extended by a single position of color 1. It remain $8*8-4*9-4*1=24$ positions which can be filled up with colors 3 and 4, respectively. An analog $G_{8,8}$ grid dominated by the colors 3 and 4 can be built. Similarly, the whole 4-colored grid $G_{16,16}$ can be constructed using reversed 2-color dominated $G_{8,8}$ grids. Figure 5 shows the constructed 4-colored grid $G_{16,16}$.

In sub-grids $G_{4,4}$ of Figure 5 each pair of rows and each pair of columns is covered by the dominating color.

1	1	1	2	2	2	2	1	3	3	3	4	4	4	4	3	1	4	3	2
1	3	4	1	2	4	3	2	3	2	1	3	4	1	2	4	2	1	4	3
4	1	3	1	3	2	4	2	1	3	2	3	2	4	1	4	3	2	1	4
3	4	1	1	4	3	2	2	2	1	3	3	1	2	4	4	4	3	2	1
2	2	2	1	1	1	1	2	4	4	4	3	3	3	3	4	1	4	3	2
2	4	3	2	1	3	4	1	4	1	2	4	3	2	1	3	2	1	4	3
3	2	4	2	4	1	3	1	2	4	1	4	1	3	2	3	3	2	1	4
4	3	2	2	3	4	1	1	1	2	4	4	2	1	3	3	4	3	2	1
3	3	3	4	4	4	4	3	1	1	1	2	2	2	2	1	1	4	3	2
3	2	1	3	4	1	2	4	1	3	4	1	2	4	3	2	2	1	4	3
1	3	2	3	2	4	1	4	4	1	3	1	3	2	4	2	3	2	1	4
2	1	3	3	1	2	4	4	3	4	1	1	4	3	2	2	4	3	2	1
4	4	4	3	3	3	3	4	2	2	2	1	1	1	1	2	1	4	3	2
4	1	2	4	3	2	1	3	2	4	3	2	1	3	4	1	2	1	4	3
2	4	1	4	1	3	2	3	3	2	4	2	4	1	3	1	3	2	1	4
1	2	4	4	2	1	3	3	4	3	2	2	3	4	1	1	4	3	2	1

Figure 6. 4-colored grid $G_{16,20}$ constructed using maximal 1-colored sub-grids $G_{4,4}$ and cyclic extension of all four colors

1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0		
1	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0		
1	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0		
0	1	0	1	0	0	0	1	0	0	0	0	0	1	0	1	0	0		
0	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0		
0	0	1	0	1	0	0	0	1	0	0	0	0	0	1	1	0	0		
0	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	1	0		
0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0	1		
0	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	1		
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1		
0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1		
0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1		
0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	1	0		
0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0		
0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0		
0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0		
0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	0	0		
0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0		

Figure 7. Rectangle-free grid $G_{18,18}$ colored by one fourth of all positions with the first color (1)

Hence, in these ranges each color is only allowed in a single position. Cyclically shifted combinations of all four colors allow the extension of the 4-colored grid $G_{16,16}$ of Figure 5 to 4-colored grid $G_{16,20}$ as shown in Figure 6.

Similarly to the 4-colored grid $G_{16,20}$ of Figure 6 a 4-colored grid $G_{20,16}$ can be constructed based on the 4-colored grid $G_{16,16}$ of Figure 5. Unfortunately, it is not possible to build a 4-colored grid $G_{17,17}$ that includes the 4-colored grid $G_{16,16}$ of Figure 5.

C. Restriction to a Single Color of 4-colored Grids

Due to the strong complexity, a divide and conquer approach may facilitate the solution of the 4-colored grid $G_{17,17}$ or even the grid $G_{18,18}$. The divide step restricts first to single color. At least one fourth of the grid positions must be covered by the first color without contradiction to the color restrictions. When such a partial solution is known, the same fill-up step must be executed taking into account the already fixed positions of the grid. This procedure must be repeated for all four colors.

The advantage of this approach is that a single Boolean variable describes whether the color is assigned to a grid position or not. Such a restriction to one half of the needed Boolean variables reduces the search space form $2^{2*18*18} = 1.16 * 10^{195}$ to $2^{18*18} = 3.41 * 10^{97}$ for the grid $G_{18,18}$ drastically.

The function f_{ecb} (7) which describes equal colors in the corners of a rectangle can be simplified to f_{ecb1} (11) for a single color in the divide and conquer approach.

$$f_{ecb1}(a_{r_i,c_k}, a_{r_i,c_l}, a_{r_j,c_k}, a_{r_j,c_l}) = (a_{r_i,c_k} \wedge a_{r_i,c_l} \wedge a_{r_j,c_k} \wedge a_{r_j,c_l}) \quad (11)$$

Transformed into a SAT problem we get:

$$\bigwedge_{i=1}^{n-1} \bigwedge_{j=i+1}^{n} \bigwedge_{k=1}^{m-1} \bigwedge_{l=k+1}^{m} \overline{f_{ecb1}(a_{r_i,c_k}, a_{r_i,c_l}, a_{r_j,c_k}, a_{r_j,c_l})} = 1 \ . \quad (12)$$

A disadvantage of this approach is that the implicit assignment of exactly one color to each grid position is lost. The values of the pair of variables $(a_{r_i,c_k}, b_{r_i,c_k})$ in the solution of (10) determine one of the four colors for the position of the row r_i and the column c_k. The value of the single variable a_{r_i,c_k} in the solution of (12) determines only whether the chosen color is assigned, $a_{r_i,c_k} = 1$, or one of the remaining colors must be used $a_{r_i,c_k} = 0$.

The equation $f_{ecb1} = 0$ has 15 solutions; only the assignment of values 1 to all a-variables is excluded. One solution of $\overline{f}_{ecb1} = 1$ calculated by a SAT solver will be the assignment of values 0 to all a-variables. This is a correct solution; the chosen color does not conflict with the color restriction when it is not assigned to any grid position. However, we are not interested in this trivial solution; we are looking for a solution where the chosen color covers one fourth of the grid positions. Therefore a SAT solver cannot solve this problem directly.

We developed a quite complicated algorithm that allows to find solutions of (12) with maximal assignments of values 1. For the sake of space we must exclude the details of this approach from this paper. These details are the topic of the paper [12]. However the results are important for the following final step to the solution. Using XBOOLE, we developed a program that implements the mentioned

r_1	s_1	t_1	r_2
t_4	u_1	u_2	s_2
s_4	u_4	u_3	t_2
r_4	t_3	s_3	r_3

Figure 8. Cyclic quadruple in a grid $G_{4,4}$

approach and calculated an allowed assignment of 81 values 1 to the 324 grid position of $G_{18,18}$. Figure 7 shows this single color solution.

Our effort to fill up the 1-colored grid $G_{18,18}$ of Figure 7 with the second color on again 81 grid positions failed. This results from the fact that the freedom for the choice of the positions is restricted by the assignments of the first color. We learned from this approach that it is not enough to know an allowed coloring for one color, these assignments must not constrain the assignment of the other colors.

D. Cyclic Color Assignments of 4-colored Grids

The smallest restrictions for the coloring of a grid by four colors are given when the number of assignments to the grid positions is equal for all four colors. For quadratic grids $G_{n,m}, n = m$, with an even number of rows n and columns m, quadruples of all grid positions can be chosen which contain all four colors. There are several possibilities of such selections of quadruples. One of them is cyclic rotation of a chosen grid position by 90 degrees around the center of the grid. Figure 8 illustrates this possibility for a simple grid $G_{4,4}$. The quadruples are labeled by the letters r, s, t, and u. The attached index specifies the element of the quadruple.

In addition to the color restriction (12) for the chosen single color we can require that this color occurs exactly once in each quadruple. This property can be expressed by two additional rules. For the corners of the grid of Figure 8, for instance, we model as first rule the requirement:

$$r_1 \vee r_2 \vee r_3 \vee r_4 = 1 \ , \tag{13}$$

so that at least one variable r_i must be equal to 1. As second rule, the additional restriction

$$\begin{aligned}&(r_1 \wedge r_2) \vee (r_1 \wedge r_3) \vee (r_1 \wedge r_4) \vee \\ &(r_2 \wedge r_3) \vee (r_2 \wedge r_4) \vee (r_3 \wedge r_4) = 0\end{aligned} \tag{14}$$

prohibits that more than one variable r_i is equal to 1.

A SAT formula can be constructed using (12) and for all cyclic quadruples as illustrated in Figure 8 both the fitted requirements (13) and the fitted restrictions (14) negated using de Morgans laws. The solution of such a SAT-formula for a square grid of even numbers of rows and columns must assign exactly one fourth of the variables to 1. Such a solution can be used rotated by 90 degrees for the second color, rotated by 180 degrees for the third color, and rotated by 270 degrees for the forth color without any contradiction.

1	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0
0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	1	0
1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1
0	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1
0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0
0	1	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	1	1	0	1	0	0	0	1	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	0	0	0
0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0
1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0
1	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1
0	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0

Figure 9. Cyclic reusable coloring of grid $G_{18,18}$

1	2	2	1	4	4	4	1	3	1	1	3	4	3	2	3	4	2
3	1	4	4	4	3	3	2	4	1	2	2	1	1	2	3	2	3
2	2	3	3	1	1	4	2	4	2	1	4	1	3	4	4	1	3
1	1	3	2	4	3	1	2	1	4	4	3	2	4	3	4	1	2
2	4	2	3	3	4	3	3	4	1	2	3	2	4	1	2	1	1
3	4	4	1	1	2	2	2	1	4	3	3	3	1	4	2	4	1
2	1	3	2	2	2	3	4	3	3	1	4	3	4	2	1	4	1
4	1	4	3	1	2	4	1	2	2	2	1	3	4	3	3	3	2
4	4	1	3	4	3	2	1	2	3	3	4	2	1	2	1	1	4
2	3	3	4	3	4	2	1	1	4	3	4	1	2	1	3	2	2
4	1	1	1	2	1	3	4	4	4	3	2	4	3	1	2	3	2
3	2	3	4	2	1	2	3	1	1	2	1	4	4	4	1	3	4
3	2	4	2	3	1	1	1	2	3	4	4	4	3	3	2	2	1
3	3	4	3	2	4	1	4	3	2	1	1	2	1	1	4	2	4
4	3	2	1	2	4	1	2	2	3	4	3	1	2	4	1	3	3
1	3	2	2	1	3	2	3	4	2	4	2	3	3	1	1	4	4
1	4	1	4	3	3	4	4	3	2	4	1	1	2	2	2	3	1
4	2	1	4	1	2	1	3	3	1	3	2	2	2	3	4	4	3

Figure 10. 4-colored grid $G_{18,18}$

We generated the cnf-file of this SAT-formula which depends on 324 variables and contains 23,976 clauses for the grid $G_{18,18}$ and tried to find a solution using the SAT-solver *clasp*. The SAT-solver *clasp* found the first cyclic reusable solution for the grid $G_{18,18}$ after 7 hours 6 minutes 34.959 seconds. Figure 9 shows this solution for the first color of the grid $G_{18,18}$.

Using the core solution of Figure 9 we have constructed the 4-colored grid $G_{18,18}$ of Figure 10 by three times rotating around the grid center by 90 degrees each and assigning the next color.

r_1	s_1	t_1	u_1	r_2
u_4	v_1	w_1	v_2	s_2
t_4	w_4	x_1	w_2	t_2
s_4	v_4	w_3	v_3	u_2
r_4	u_3	t_3	s_3	r_3

Figure 11. Cyclic quadruple in a grid $G_{5,5}$

Many other 4-colored grids can be created from the solution in Figure 10 by permutations of rows, columns and colors. Several valid 4-colored grids $G_{17,18}$ originate from the 4-colored grid $G_{18,18}$; by removing any single row and by removing any single column we get 4-colored grids $G_{18,17}$. Obviously, several so far unknown 4-colored $G_{17,17}$ can be selected from the 4-colored grid of Figure 10 removing both any single row and any single column.

It should be mentioned that the approach of cyclic reusable single assignments can be applied to 4-colored square grids of an odd number of rows and columns, too. The central position must be colored with the first chosen color. Figure 11 shows the principle of the quadruple assignment in this case.

The SAT-solver *clasp* found the first cyclic 4-colorable solution for odd grids up to $G_{15,15}$ in less than 0.6 seconds but could not solve this task for grid $G_{17,17}$ within one week.

V. Conclusion

We explored in this paper the so far unsolved problem whether the grids $G_{17,17}$, $G_{17,18}$, $G_{18,17}$, and $G_{18,18}$ are 4-colorable. Our study has shown that the fraction of 4-colorable grids of the size 18×18 is extremely small. Hence, finding a 4-colored grid $G_{18,18}$ out of the unimaginable large number of $1.16798 * 10^{195}$ of all possible assignments of 4 colors is significantly more difficult than detecting a single atom within the whole galaxy.

We evaluated several approaches, developed and utilized different computer programs, and combined all collected knowledge. From all our approaches we learned, neither a fitting default computer program nor a human being will be able to solve such a highly complex problem, but commonly we were able to find 4-colored solutions for the grid of the size 18×18 and consequently for all sub-grids.

Our successful final approach to find 4-colored grids $G_{18,18}$ consists in the detection of a cyclic sub-task as conclusion of an experiment using XBOOLE, an extension of the model, the utilization of the SAT-solver *clasp* and the construction of the 4-colored grid of the size 18×18.

For Mathematitians in the area of bi-partite Ramsey numbers we can state: instead of $17 \leq BR(2,4) \leq 19$ we have now $BR(2,4) = 19$.

References

[1] Biere, A. *Lingeling, Plingeling, PicoSAT and PrecoSAT at SAT Race 2010*. Technical Report 10/1, August 2010, FMV Reports Series, Institute for Formal Models and Verification, Johannes Kepler University, Altenbergerstr. 69, 4040 Linz, Austria, 2010, pp 1 - 4.

[2] Biere, A., Heule, M., Van Maaren, H. and Walsh, T. *Handbook of Satisfiability*. Volume 185: Frontiers in Artificial Intelligence and Applications, IOS Press Amsterdam, 2009.

[3] Fenner, St., Gasarch, W., Glover, Ch. and Purewal, S. *Rectangle Free Coloring of Grids*. http://www.cs.umd.edu/ gasarch/papers/grid.pdf.

[4] Fortnow, L. and Gasarch, B. *Computational Complexity and other fun stuff in math and computer science*. blog on webpage: http://blog.computationalcomplexity.org/2009/11/17x17-challenge-worth-28900-this-is-not.html

[5] Gebser, M., Kaufmann, B., Neumann, A. and Schaub, T. *clasp: A Conflict-Driven Answer Set Solver*. in: Baral, C., Brewka, G. and Schlipf J. LPNMR 2007 (LNAI 4483), Springer, 2007, pp. 260 - 265.

[6] Graham, R., Rothschild, B. and Spencer, J. *Ramsey Theory - Second Edition*. JOHN WILEY & SONS, New York, 1990.

[7] Liu, C. Kuehlmann, A. and Moskewicz, M. *CAMA: A Multi-Valued Satisfiability Solver*. ICCAD 2003, San Jose, California, USA, November 11-13, 2003, pp. 326 - 333.

[8] Marx, D. *Graph colouring problems and their applications in scheduling*. Periodica Polytechnica, Electrical Engineering, Volume 48, Number 1, 2004, pp. 11 - 16.

[9] Posthoff, Ch. and Steinbach, B. *Logic Functions and Equations - Binary Models for Computer Science*. Springer, Dordrecht, The Netherlands, 2004.

[10] Steinbach, B., Posthoff, Ch. and Wessely, W. *Approaches to Shift the Complexity Limitations of Boolean Problems*. in: Computer - Aided Design of Discrete Devices - CAD DD 2010, Proceedings of the Seventh International Conference, 16 - 17 November 2010, Minsk, Belarus, pp. 84 - 91.

[11] Steinbach, B. and Posthoff, Ch. *Logic Functions and Equations - Examples and Exercises*. Springer Science + Business Media B.V., 2009.

[12] Steinbach, B. and Posthoff, Ch. *Utilization of Permutation Classes for Solving Extreme Complex 4-Colorable Rectangle-free Grids*. submitted to The 2012 International Conference on Systems and Informatics (ICSAI 2012) Yantai, China.

[13] Steinbach, B. *XBOOLE - A Toolbox for Modelling, Simulation, and Analysis of Large Digital Systems*. System Analysis and Modeling Simulation, Gordon & Breach Science Publishers, Volume 9, Number 4, 1992, pp. 297–312.

[14] Steinbach, B., Wessely, W. and Posthoff, Ch.: *Several Approaches to Parallel Computing in the Boolean Domain*. in: Chaudhuri, P.; Ghosh, S.; Buyya, R. K.; Cao, J.; Dahiya, D.: 2010 1st International Conference on Parallel, Distributed and Grid Computing (PDGC 2010), October 28 - 30, 2010, Jaypee University of Information Technology Waknaghat, Solan, H.P., India, 2010, pp. 6 - 11.