Searching for Monochromatic-Square-Free Ramsey Grid Colorings via SAT Solvers

Paul Walton
Computer Science and Computer Engineering
University of Arkansas
Fayetteville, AR 72701, USA
pwalton@uark.edu

Wing Ning Li
Computer Science and Computer Engineering
University of Arkansas
Fayetteville, AR 72701, USA
wingning@uark.edu

***Abstract*—Monochromatic-square-free Ramsey grid coloring is a challenging problem to solve computationally. It relates to Ramsey-theoretic combinatorics and multiple party communication protocol complexity. Recently, using a parallel search algorithm on cluster based super computers, a 2-color 14x14 solution was found in 2.5 hours. In this paper, we report on another approach to solve the monochromatic-square-free Ramsey grid coloring problem. The approach first reduces monochromatic-square-free Ramsey grid coloring to satisfiability, and then applies an existing SAT solver to solve it. Using a SAT solver on a sequential machine, we found another 2-color 14x14 solution in about two seconds, which is more than four thousand times faster than the parallel algorithm using 288 cores. We also showed that no 2-color 15x15 solution exists, which answered the open question whether a 15x15 grid graph was monochromatic-square-free 2-colorable or not.**

Keywords*—*Ramsey Theory, Grid Coloring, SAT, Reduction, experimentation.

I. Introduction

The monochromatic-square-free Ramsey grid coloring problem is a special case of the shape-free grid coloring problem from Ramsey Theory [1], [8], and it asks the following:

> Given a grid graph $G_{n,m}$ of size n-by-m, and a set of c distinct colors, does there exist a coloring of the grid vertices such that there is no square of size a-by-a within $G_{n,m}$, where $a \neq 0$, whose corner vertices are all colored the same? If the answer is "Yes" then there exists a monochromatic-square-free coloring of the grid, otherwise no such coloring exists.

Figure 1 shows two colorings of the grid graph $G_{4,4}$. One is monochromatic-square-free, and the other one is not (indicated by the X's).

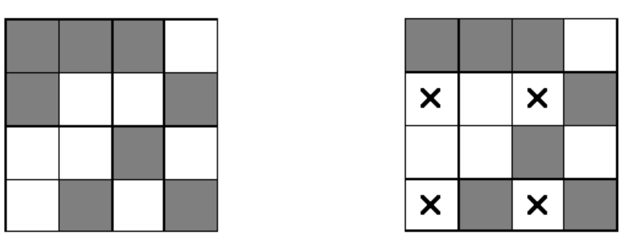

Fig. 1. A legally 2-colored grid, and an illegally 2-colored grid [2].[0]

Recently, Apon and Li [2] have developed backtracking based sequential and parallel algorithms to solve the monochromatic-square-free Ramsey grid coloring problem. They reported the first-known 2-color solution for the 13x13 grid, which is also a lexicographical first solution, and the first known 2-color solution for the 14x14 grid. The solution for the 14x14 grid was obtained by a parallel search on a cluster based super computer. Using 288 cores, it took the program about 2.5 hours to find the solution.

In our paper we will be using a reduction to the satisfiability problem as a method to solve the monochromatic-square-free Ramsey grid coloring problem. The satisfiability problem (SAT) was the first known NP-complete problem [9], and no polynomial time algorithm exists to solve the problem unless P=NP. Nonetheless, over the years researchers have been developing, and improving algorithms to solve the SAT problem [3], [11], [5], [4], [6]. Compared to the monochromatic-square-free Ramsey grid coloring problem, algorithms to solve the SAT problem have reached a high level of maturity, and are fairly well known [3]. While the algorithms are still exponential in the size of the SAT instance [3], many of the SAT solvers today are good enough that they have made using this technique a viable option. For example, SAT solvers have been successfully used to solve design automation problems that involve logic circuits [11]. In all of our experiments we used the Minisat SAT solver [7], which is a standard open-source SAT solver. There are many other SAT solvers to choose from, but Minisat is a standard benchmark that most researchers use.

Few researchers have reduced computational problems to satisfiability in order to solve it unless the problems themselves are related to boolean formula, or functions such as in VLSI circuit design [11]. Using a natural encoding of the input instances of the monochromatic-square-free Ramsey grid coloring problem, where the graph is specified by a set of vertices and a set of edges, the monochromatic-square-free Ramsey grid coloring problem cannot be NP-complete unless P=NP [2], [10]. So, it may seem counterintuitive for us to propose reducing the monochromatic-square-free Ramsey grid coloring problem to the satisfiability problem, using a SAT solver to solve the mapped SAT instance, and then map the truth assignment back to an answer of the monochromatic-square-free Ramsey grid coloring instance. Nevertheless, our experi-

[0]Image included with permission from Li.

978-1-4799-0604-8/13/$31.00 ©2013 IEEE

mentation has demonstrated that the proposed approach is currently the most efficient method to solve the monochromatic-square-free Ramsey grid coloring problem. It took about 2.5 hours using 288 cores of a super computer for the parallel algorithm in [2] to find a 2-color 14x14 solution. Using the proposed approach, and the Minisat SAT solver running on a typical PC, we were able to obtain a 2-color 14x14 solution in about 2.14 seconds. Using the sequential algorithm reported in [2], it took the sequential algorithm about two weeks to obtain a 13x13 solution (the lexicographically first solution). Again, using the proposed method, we were able to obtain a 13x13 solution (which is not the lexicographically first solution though) in about 0.25 seconds. Both sequential programs ran on comparable computers, and our method is about 4.5 million times faster than what is reported in [2]. In addition to finding a solution in less time, the efficiency of the approach allows us to show that for grid graphs of size 15x15 or larger no monochromatic-square-free Ramsey grid coloring exists for two colors. Hence, we have settled the open question whether a 15x15 grid graph was monochromatic-square-free 2-colorable or not [2].

The remainder of the paper is organized as follows: Section 2 provides a precise definition of the grid coloring problem, and gives a reduction to the satisfiability problem for any grid graph $G_{n,m}$ and number of colors c. Section 3 describes the implementation and experiment results. Section 4 covers the conclusion, future work, and open questions.

II. Reduction to SAT

The SAT problem and the monochromatic-square-free Ramsey grid coloring problem are precisely defined as follows.

Definition 1: SAT [9]:

INPUT: A set of boolean variables $x_1, x_2, \ldots, x_n$ and a set of clauses $C_1, C_2, \ldots, C_m$, where each clause is the disjunction (logical or) of some literals. A literal is a variable (x_i) or the complement of a variable ($\overline{x_i}$).

OUTPUT: Determine if it is possible to have a truth assignment of the boolean variables that makes all clauses true (satisfied). If the answer is yes, give such a truth assignment.

Let $G_{n,m} = (V, E)$ be an $n \times m$ grid graph, which has $n \cdot m$ vertices and $2 \cdot n \cdot m - n - m$ edges, where $V = \{(i,j) \mid 0 \leq i < n, 0 \leq j < m\}$ and $E = \{< (i,j),(i+1,j) >\mid 0 \leq i < n-1, 0 \leq j < m\} \cup \{< (i,j),(i,j+1) >\mid 0 \leq i < n, 0 \leq j < m-1\}$. A square in $G_{n,m}$ is any set of four distinct vertices that form a square. Formally, $\{(i,j),(i+a,j),(i,j+a),(i+a,j+a)\}$ is a square, where $0 \leq i < n-1, 0 \leq j < m-1$ and $1 \leq a < min\{n-i, m-j\}$. A square is called monochromatic if all four vertices have the same color under some color assignment of the vertices.

Definition 2: Monochromatic-Square-Free Ramsey Grid Coloring [2]:

INPUT: $G_{n,m} = (V, E)$ and an integer c, which is the number of available colors.

OUTPUT: Determine if it is possible to assign a color to each vertex (i,j) in V, from the set of colors $\{0, 1, \ldots, c-1\}$, such that no monochromatic square exists in $G_{n,m}$. If the answer is yes, give such a color assignment or coloring.

We call those instances that have a yes answer c-colorable.

A. Transformation

To map the monochromatic-square-free Ramsey grid coloring problem to SAT, we need to create variables and clauses from $G_{n,m}$ and c. Informally, what we would like those variables and clauses to say is that there is a true assignment that makes all clauses true if and only if each vertex gets a color from $\{0, 1, \ldots, c-1\}$ in such a way that no monochromatic square exists. To this end, we will associate each vertex (i,j) and each color $k \in \{0, 1, \ldots, c-1\}$ with a boolean variable $X_{(i,j,k)}$. The intended meaning for $X_{(i,j,k)}$ is that the variable is true if vertex (i,j) is assigned color k, and is false if k is not assigned to (i,j). Notice that in general a truth assignment may not assign $X_{(i,j,k)}$ to true for any k, which implies vertex (i,j) is not assigned a color. Or a truth assignment may assign $X_{(i,j,k)}$ to true for several values of k, which implies multiple colors are assigned to vertex (i,j). We must avoid the former case since each vertex must get a color. However, the later case will not create a problem. If a vertex has multiple colors, none of which results in a monochromatic square, then we have multiple solutions to the grid coloring problem, and can simply pick any one of these colors as the assigned color for vertex (i,j). But, we have to set up clauses to prevent the former case, and make sure that the condition that no monochromatic square exists is true. So we have the following constraints, which will be handled by clauses in the resulting SAT instance.

1) Each vertex (i,j), where $0 \leq i < n$ and $0 \leq j < m$, must be colored with at least one of the colors chosen from the set of allowable colors $\{0, 1, ..., c-1\}$.
2) The color assignment (that is the truth assignment to $X_{(i,j,k)}$) should not create any monochromatic squares.

Constraint 1 is a constraint for each vertex. Since there are $n \cdot m$ vertices, we will have that many constraints as clauses, defined as follows:

$$C_{(i,j)} = (X_{(i,j,0)} \vee X_{(i,j,1)} \vee \cdots \vee X_{(i,j,c-1)})$$

where $0 \leq i < n$, and $0 \leq j < m$. For $C_{(i,j)}$ to be true, at least one of the $X_{(i,j,k)}$, where $0 \leq k < c$, must also be true. Hence, vertex (i,j) is assigned a color.

Constraint 2 is a constraint for each of the squares in $G_{n,m}$, and in plain English it states that "it is not the case that the four vertices (i,j), $(i+a,j)$, $(i,j+a)$, and $(i+a,j+a)$ all have the same color k". This translates to the following clauses formally:

$$C_{(i,j,k,a)} = (\overline{X_{(i,j,k)} \wedge X_{(i+a,j,k)} \wedge X_{(i,j+a,k)} \wedge X_{(i+a,j+a,k)}})$$

where $0 \leq i < n-1$, $0 \leq j < m-1$, $0 \leq k < c-1$, and $1 \leq a < min\{n-i, m-j\}$.

By applying De Morgan's Law this becomes:

$$C_{(i,j,k,a)} = (\overline{X}_{(i,j,k)} \vee \overline{X}_{(i+a,j,k)} \vee \overline{X}_{(i,j+a,k)} \vee \overline{X}_{(i+a,j+a,k)}).$$

Note that the above clauses cover all unique square and color combinations. In particular, if we fix i, j, and a then we focus on one square. As the k varies from 0 to $c-1$, these clauses ensure that those four vertices do not have the same color for each of the colors from 0 to $c-1$. If a truth assignment exists that satisfies all of these clauses then the grid graph is monochromatic-square-free.

Thus, the number of unique squares that exist inside an $n \times m$ grid depends on how n and m relate to each other. If $n = m$ we have the following summation:

$$\sum_{i=0}^{n-1} i^2 = \frac{n(n-1)(2n-1)}{6},$$

when $n \leq m$ we have:

$$\sum_{i=0}^{n-1} i^2 + \sum_{j=n}^{m-1} \sum_{k=1}^{n-1} (n-k) = \frac{n(n-1)(3m-n-1)}{6},$$

and likewise, when $n \geq m$ we have:

$$\sum_{i=0}^{m-1} i^2 + \sum_{j=m}^{n-1} \sum_{k=1}^{m-1} (m-k) = \frac{m(m-1)(3n-m-1)}{6}.$$

The total number of clauses for constraint 2 is the number of colors, c, multiplied by the total number of unique squares. For instance, if $n = m$ then the total number of clauses created will be

$$\frac{cn(n-1)(2n-1)}{6}.$$

B. An illustrative example

While the following example may seem trivial, it does illustrate each step of the transformation in a concrete fashion. Applying this to a larger problem would have required more space in this paper, and would not yield any more clarity or insight to the reader.

1) A 2x2 Grid, with 2 colors: A monochromatic-square-free Ramsey grid coloring instance: Given $G_{2,2}$, where $n = 2$ and $m = 2$, and the set of valid colors $C = \{0, 1\}$, where $c = 2$, is there a coloring of the grid that is monochromatic-square-free, yes or no?

2) Boolean variables: Create the following set of Boolean variables:

$$\{X_{(0,0,0)}, X_{(0,0,1)}, X_{(0,1,0)}, X_{(0,1,1)},\\ X_{(1,0,0)}, X_{(1,0,1)}, X_{(1,1,0)}, X_{(1,1,1)}\}.$$

3) Clauses due to constraint 1: Create the following clauses to make sure all the vertices are colored:

$$C_{(0,0)} \equiv (X_{(0,0,0)} \vee X_{(0,0,1)})$$
$$C_{(0,1)} \equiv (X_{(0,1,0)} \vee X_{(0,1,1)})$$
$$C_{(1,0)} \equiv (X_{(1,0,0)} \vee X_{(1,0,1)})$$
$$C_{(1,1)} \equiv (X_{(1,1,0)} \vee X_{(1,1,1)})$$

4) Clauses due to constraint 2: Due to the small grid graph and the range on a defined above, there is only one vertex, (0,0), serving as the upper left corner of any square. Create the following clauses to check all unique color and square combinations for each vertex:

$$C_{(0,0,0,1)} \equiv (\overline{X}_{(0,0,0)} \vee \overline{X}_{(1,0,0)} \vee \overline{X}_{(0,1,0)} \vee \overline{X}_{(1,1,0)})$$
$$C_{(0,0,1,1)} \equiv (\overline{X}_{(0,0,1)} \vee \overline{X}_{(1,0,1)} \vee \overline{X}_{(0,1,1)} \vee \overline{X}_{(1,1,1)})$$

5) The corresponding SAT instance: The SAT instance obtained from $G_{2,2}$ and $c = 2$ is the following.

The set of Boolean variables is

$$\{X_{(0,0,0)}, X_{(0,0,1)}, X_{(0,1,0)}, X_{(0,1,1)},\\ X_{(1,0,0)}, X_{(1,0,1)}, X_{(1,1,0)}, X_{(1,1,1)}\}.$$

and the clauses are:

$$\{C_{(0,0)}, C_{(0,1)}, C_{(1,0)}, C_{(1,1)}, C_{(0,0,0,1)}, C_{(0,0,1,1)}\}.$$

In this case, the following truth assignment (literals in the set are set to true) will satisfy the above SAT instance, and therefore also define a monochromatic-square-free coloring of the grid graph:

$$\{X_{(0,0,0)}, \overline{X}_{(0,0,1)}, X_{(0,1,0)}, \overline{X}_{(0,1,1)},\\ X_{(1,0,0)}, \overline{X}_{(1,0,1)}, \overline{X}_{(1,1,0)}, X_{(1,1,1)}\}.$$

0	0
0	1

And, the following truth assignment will not satisfy the SAT instance because $C_{(0,0,0,1)}$ is not satisfied, and therefore will not define a monochromatic-square-free coloring of the grid:

$$\{X_{(0,0,0)}, \overline{X}_{(0,0,1)}, X_{(0,1,0)}, \overline{X}_{(0,1,1)},\\ X_{(1,0,0)}, \overline{X}_{(1,0,1)}, X_{(1,1,0)}, \overline{X}_{(1,1,1)}\}.$$

0	0
0	0

C. Proof of Correctness

We show that an instance of the Monochromatic-Square-Free Grid Coloring problem is c-colorable if and only if the corresponding SAT instance obtained from the above transformation is satisfiable. Further, a monochromatic-square-free Ramsey grid coloring can be obtained from the truth assignment by assigning color k to vertex (i, j), where k is the smallest value of the third index such that $X_{(i,j,k)}$ is true.

Lemma 1: If the instance of the Monochromatic-Square-Free Ramsey Grid Coloring problem is c-colorable then the corresponding SAT instance obtained from the above transformation is satisfiable.

Proof: Under the Monochromatic-Square-Free c-color assignment, each vertex (i, j) gets some color k, $0 \leq k < c-1$. Set $X_{(i,j,k)}$ to true in the truth assignment and set $X_{(i,j,p)}$ to false, where $0 \leq p < c-1$ and $p \neq k$. Now all the variables are assigned a truth value. Clearly each clause $C_{(i,j)}$ is satisfied under this truth assignment as for some k, $X_{(i,j,k)}$ is true, and $C_{(i,j)}$ is the disjunction of each $X_{(i,j,p)}$, where $0 \leq p < c-1$. As for each clause $C_{(i,j,k,a)} = (\overline{X}_{(i,j,k)} \vee \overline{X}_{(i+a,j,k)} \vee \overline{X}_{(i,j+a,k)} \vee \overline{X}_{(i+a,j+a,k)})$, it is not satisfied provided the four boolean variables $X_{(i,j,k)}$, $X_{(i+a,j,k)}$, $X_{(i,j+a,k)}$, and $X_{(i+a,j+a,k)}$ are all true under the truth assignment. If that's the case then the four vertices, (i, j), $(i+a, j)$, $(i, j+a)$, $(i+a, j+a)$, which form a square, are all colored k, which is a contradiction as the c-color assignment is Monochromatic-Square-Free. Hence all clauses are satisfied, and the SAT instance is satisfiable. ■

Lemma 2: If the SAT instance obtained from the above transformation is satisfiable then the corresponding Monochromatic-Square-Free Ramsey Grid Coloring instance is c-colorable.

Proof: Since $C_{(i,j)}$ is true under the satisfiable truth assignment, then at least one of the variables $X_{(i,j,p)}$, where $0 \leq p < c-1$, must be true. Let k be the smallest value such that $X_{(i,j,k)}$ is true. Vertex (i,j) gets color k in the coloring. Now every vertex is colored. If for some four vertices (i,j), $(i+a,j)$, $(i,j+a)$, $(i+a,j+a)$, which form a square, are all colored k, then the four boolean variables $X_{(i,j,k)}$, $X_{(i+a,j,k)}$, $X_{(i,j+a,k)}$, and $X_{(i+a,j+a,k)}$ are all true under the satisfiable truth assignment. Then this means $C_{(i,j,k,a)} = (\overline{X}_{(i,j,k)} \vee \overline{X}_{(i+a,j,k)} \vee \overline{X}_{(i,j+a,k)} \vee \overline{X}_{(i+a,j+a,k)})$ is not satisfied, which is a contradiction. Hence, the coloring is Monochromatic-Square-Free. ■

From Lemmas 1 and 2 we obtain the following Theorem.

Theorem 1: An instance of the Monochromatic-Square-Free Ramsey Grid Coloring problem is c-colorable if and only if the corresponding SAT instance obtained from the above transformation is satisfiable.

III. RESULTS

The transformation method described in section 2 has been implemented as a program that takes n, m and c as input, and produces the mapped SAT instance in a DIMACS CNF file [3], which most standard SAT solvers use as input. The SAT instance is then solved by the Minisat SAT solver [7]. The truth assignment produced by the SAT solver is then translated into a monochromatic-square-free Ramsey grid coloring by another program. In all cases, the PC used was an Intel Core2 Duo 3 GHz CPU with 4 GB of RAM, and the command line flags given to the Minisat program were "-no-luby -rinc=1.5 -phase-saving=0 -rnd-freq=0.02", which seemed to yield better results than the standard default values. The conversion time between instances and solutions is too small and is ignored.

Using this approach, we have generated a monochromatic-square-free 2-coloring for $G_{13,13}$ in 0.25 seconds using the Minisat SAT solver. The result is shown below:

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

In [2], Apon and Li report that it took their sequential algorithm about two weeks to obtain a 13x13 solution (the lexicographically first solution). Ours was run on a comparable PC to the one they reported, and while ours is not the lexicographically first solution, our method was about 4.5 million times faster.

Again, using this approach, we generated a monochromatic-square-free 2-coloring for $G_{14,14}$ in 2.14 seconds using the Minisat SAT solver. The result is shown below:

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

This coloring is different from what is reported in [2], which was obtained by a parallel algorithm using 288 cores of a super computer and took 2.5 hours to get. Even though the hardware we used is a typical PC and the hardware the parallel algorithm used is 288 cores of a super computer, due to the difference in methods, the time for us to get a solution for the 14x14 instance is more than 4000 times faster than that reported in [2].

Next, we performed the same process for monochromatic-square-free 2-colorings of $G_{14,15}$, and $G_{15,15}$. In both cases, the Minisat SAT solver determined that the SAT instance for the problem was unsatisfiable. This means that no monochromatic-square-free 2-colorings can exist for grid graphs of those sizes, and therefore the largest 2-colorable monochromatic-square-free grid graph is $G_{14,14}$. This is based on recognizing the symmetry between $G_{14,15}$ and $G_{15,14}$, and the fact that any 2-coloring of a grid graph larger than $G_{15,15}$ would also have as its subset a monochromatic-square-free 2-coloring of size $G_{15,15}$. Using the Minisat SAT solver it took 20.58 minutes to determine that the SAT instance for $G_{14,15}$ was unsatisfiable, and 22.18 minutes to determine that the SAT instance for $G_{15,15}$ was unsatisfiable.

IV. CONCLUSIONS

We have shown a polynomial time reduction from the monochromatic-square-free Ramsey Grid coloring problem to the Satisfiability problem (SAT). This allowed us to generate monochromatic-square-free 2-colorings for $G_{13,13}$ and $G_{14,14}$ in a substantially reduced amount of time when compared to the results in [2], which was previously the best known approach to solving the monochromatic-square-free Ramsey Grid coloring problem. In [2], Apon and Li found the lexicographically first 2-coloring of $G_{13,13}$ in about two weeks on a sequential machine, while our approach was able to find a 2-coloring of $G_{13,13}$ in 0.25 seconds on a comparable PC. And in [2], Apon and Li found a 2-coloring of $G_{14,14}$ in 2.5 hours using a parallel algorithm on 288 cores of a parallel super computer, while we were able to find a different 2-coloring of $G_{14,14}$ in 2.14 seconds on a sequential PC.

The monochromatic-square-free Ramsey grid coloring problem cannot be NP-complete unless P=NP [2], [10], therefore it may seem counterintuitive to reduce it to an NP-complete problem in order to solve it. However, the experimental results show that this method is currently the best way to solve the monochromatic-square-free Ramsey Grid coloring problem. Backtracking is applied by both the Minisat SAT solver and the methods in [2], but the proposed SAT solver approach is much faster. So, the open question is, why is this the case?

Future work could involve a detailed analysis of the SAT instances to see if there is something about their structure which makes them easier to solve. Or, perhaps there is something about the monochromatic-square-free Ramsey grid coloring problem itself that we could garner new knowledge about which would make solving it in its native form more efficient, possibly a sub-exponential algorithm which yields better performance than using a SAT solver.

Other research might also involve finding the maximum grid size for monochromatic-square-free 3-colorings, 4-colorings, etc., or any other Ramsey grid coloring problem in general.

V. Appendix Note

After writing and submitting the preliminary version of this paper for publication, it was brought to our attention by an anonymous referee the work done by Steinbach and Posthoff [12], which also used the approach of reduction to SAT and using a SAT solver to answer several computationally complex questions related to 4-colored monochromatic-rectangle-free grid coloring. As it turned out, the work reported in [12] and the work reported in this paper were done independently. Notice that a grid coloring is monochromatic-rectangle-free implies the grid coloring is also monochromatic-square-free. However, the converse is not true. Thus, the algorithm used in solving monochromatic-rectangle-free grid coloring cannot be used directly to solve monochromatic-square-free grid coloring. We would like to thank the anonymous reviewer for bringing the work of Steinbach and Posthoff to our attention.

References

[1] D. Apon. On the Complexity of Grid Coloring. Master's thesis, University of Arkansas, Fayetteville Arkansas, 2011.

[2] D. Apon and W. N. Li. Parallel search for ramsey grid colorings. In *Proceedings of the 50th Annual Southeast Regional Conference*, ACM-SE '12, pages 274–279, New York, NY, USA, 2012. ACM.

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

[4] W. Blochinger, W. Westje, W. Kuchlin, and S. Wedeniwski. Zetasat - boolean satisfiability solving on desktop grids. In *Proceedings of the Fifth IEEE International Symposium on Cluster Computing and the Grid (CCGrid'05) - Volume 2 - Volume 02*, CCGRID '05, pages 1079–1086, Washington, DC, USA, 2005. IEEE Computer Society.

[5] W. Chrabakh and R. Wolski. Gridsat: A chaff-based distributed sat solver for the grid. In *Supercomputing, 2003 ACM/IEEE Conference*, page 37, nov. 2003.

[6] W. Chrabakh and R. Wolski. Gridsat: a system for solving satisfiability problems using a computational grid. *Parallel Comput.*, 32(9):660–687, October 2006.

[7] N. Eén and N. Sörensson. An extensible sat-solver. In *SAT*, pages 502–518, 2003.

[8] Fenner, Gasarch, Glover, and Purewal. Rectangle free coloring of grids. Work in progress, University of Maryland, College Park, 2010.

[9] M. R. Garey and J. D. S. *Computers and Intractability, A Guide to the Theory of NP-completeness*. Freeman, San Francisco, CA, USA, 1979.

[10] S. R. Mahaney. Sparse complete sets for np: Solution of a conjecture of berman and hartmanis. *Journal of Computer and System Sciences*, 25(2):130 – 143, 1982.

[11] M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: engineering an efficient sat solver. In *Proceedings of the 38th annual Design Automation Conference*, DAC '01, pages 530–535, New York, NY, USA, 2001. ACM.

[12] B. Steinbach and C. Posthoff. Extremely complex 4-colored rectangle-free grids: Solution of open multiple-valued problems. In *Multiple-Valued Logic (ISMVL), 2012 42nd IEEE International Symposium on*, pages 37–44, 2012.