Computing the Ramsey Number R(4,3,3) using Abstraction and Symmetry breaking*

Michael Codish[1], Michael Frank[1], Avraham Itzhakov[1], and Alice Miller[2]

[1] Department of Computer Science, Ben-Gurion University of the Negev, Israel
[2] School of Computing Science, University of Glasgow, Scotland

Abstract. The number $R(4,3,3)$ is often presented as the unknown Ramsey number with the best chances of being found "soon". Yet, its precise value has remained unknown for almost 50 years. This paper presents a methodology based on *abstraction* and *symmetry breaking* that applies to solve hard graph edge-coloring problems. The utility of this methodology is demonstrated by using it to compute the value $R(4,3,3) = 30$. Along the way it is required to first compute the previously unknown set $\mathcal{R}(3,3,3;13)$ consisting of 78,892 Ramsey colorings.

1 Introduction

This paper introduces a general methodology that applies to solve graph edge-coloring problems and demonstrates its application in the search for Ramsey numbers. These are notoriously hard graph coloring problems that involve assigning colors to the edges of a complete graph. An $(r_1, \dots, r_k; n)$ Ramsey coloring is a graph coloring in k colors of the complete graph K_n that does not contain a monochromatic complete sub-graph K_{r_i} in color i for each $1 \leq i \leq k$. The set of all such colorings is denoted $\mathcal{R}(r_1, \dots, r_k; n)$. The Ramsey number $R(r_1, \dots, r_k)$ is the least $n > 0$ such that no $(r_1, \dots, r_k; n)$ coloring exists. In particular, the number $R(4,3,3)$ is often presented as the unknown Ramsey number with the best chances of being found "soon". Yet, its precise value has remained unknown for more than 50 years. It is currently known that $30 \leq R(4,3,3) \leq 31$. Kalbfleisch [7] proved in 1966 that $R(4,3,3) \geq 30$, Piwakowski [11] proved in 1997 that $R(4,3,3) \leq 32$, and one year later Piwakowski and Radziszowski [12] proved that $R(4,3,3) \leq 31$. We demonstrate how our methodology applies to computationally prove that $R(4,3,3) = 30$.

Our strategy to compute $R(4,3,3)$ is based on the search for a $(4,3,3;30)$ Ramsey coloring. If one exists, then because $R(4,3,3) \leq 31$, it follows that $R(4,3,3) = 31$. Otherwise, because $R(4,3,3) \geq 30$, it follows that $R(4,3,3) = 30$.

In recent years, Boolean SAT solving techniques have improved dramatically. Today's SAT solvers are considerably faster and able to manage larger instances than were previously possible. Moreover, encoding and modeling techniques are better understood and increasingly innovative. SAT is currently applied to solve

* Supported by the Israel Science Foundation, grant 182/13.

a wide variety of hard and practical combinatorial problems, often outperforming dedicated algorithms. The general idea is to encode a (typically, NP) hard problem instance, μ, to a Boolean formula, φ_μ, such that the satisfying assignments of φ_μ correspond to the solutions of μ. Given such an encoding, a SAT solver can be applied to solve μ.

Our methodology in this paper combines SAT solving with two additional concepts: *abstraction* and *symmetry breaking*. The paper is structured to let the application drive the presentation of the methodology in three steps. Section 2 presents: preliminaries on graph coloring problems, some general notation on graphs, and a simple constraint model for Ramsey coloring problems. Section 3 presents the first step in our quest to compute $R(4,3,3)$. We introduce a basic SAT encoding and detail how a SAT solver is applied to search for Ramsey colorings. Then we describe and apply a well known embedding technique, which allows to determine a set of partial solutions in the search for a $(4,3,3;30)$ Ramsey coloring such that if a coloring exists then it is an extension of one of these partial solutions. This may be viewed as a preprocessing step for a SAT solver which then starts from a partial solution. Applying this technique we conclude that if a $(4,3,3;30)$ Ramsey coloring exists then it must be $\langle 13,8,8\rangle$ regular. Namely, each vertex in the coloring must have 13 edges in the first color, and 8 edges in each of the other two colors. This result is already considered significant progress in the research on Ramsey numbers as stated in [19]. To further apply this technique to determine if there exists a $\langle 13,8,8\rangle$ regular $(4,3,3;30)$ Ramsey coloring requires to first compute the currently unknown set $\mathcal{R}(3,3,3;13)$.

Sections 4—7 present the second step: computing $\mathcal{R}(3,3,3;13)$. Section 4 illustrates how a straightforward approach, combining SAT solving with *symmetry breaking*, works for smaller instances but not for $\mathcal{R}(3,3,3;13)$. Then Section 5 introduces an *abstraction*, called degree matrices, Section 6 demonstrates how to compute degree matrices for $\mathcal{R}(3,3,3;13)$, and Section 7 shows how to use the degree matrices to compute $\mathcal{R}(3,3,3;13)$. Section 8 presents the third step re-examining the embedding technique described in Section 3 which given the set $\mathcal{R}(3,3,3;13)$ applies to prove that there does not exist any $(4,3,3;30)$ Ramsey coloring which is also $\langle 13,8,8\rangle$ regular. Section 9 presents a conclusion.

2 Preliminaries and Notation

In this paper, graphs are always simple, i.e. undirected and with no self loops. For a natural number n let $[n]$ denote the set $\{1,2,\ldots,n\}$. A graph coloring, in k colors, is a pair (G,κ) consisting of a simple graph $G=(V,E)$ and a mapping $\kappa\colon E\to[k]$. When G is clear from the context we refer to κ as the graph coloring. We typically represent $G=([n],E)$ as a (symmetric) $n\times n$ adjacency matrix, A, defined such that

$$A_{i,j}=\begin{cases}\kappa((i,j)) & \text{if } (i,j)\in E\\ 0 & \text{otherwise}\end{cases}$$

Given a graph coloring (G, κ) in k colors with $G = (V, E)$, the set of neighbors of a vertex $u \in V$ in color $c \in [k]$ is $N_c(u) = \{\ v \,|\, (u, v) \in E, \kappa((u, v)) = c\ \}$ and the color-c degree of u is $deg_c(u) = |N_c(u)|$. The color degree tuple of u is the k-tuple $deg(u) = \langle deg_1(u), \ldots, deg_k(u)\rangle$. The sub-graph of G on the c colored neighbors of $x \in V$ is the projection of G to vertices in $N_c(x)$ defined by $G^c_x = (N_c(x), \{\ (u, v) \in E \,|\, u, v \in N_c(x)\ \})$.

For example, take as G the graph coloring depicted by the adjacency matrix in Figure 3 with u the vertex corresponding to the first row in the matrix. Then, $N_1(u) = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$, $N_2(u) = \{14, 15, 16, 17, 18, 19, 20, 21\}$, and $N_3(u) = \{22, 23, 24, 25, 26, 27, 28, 29\}$. The subgraphs G^1_u, G^2_u, and G^3_u are highlighted by the boldface text in Figure 3.

An $(r_1, \ldots, r_k; n)$ Ramsey coloring is a graph coloring in k colors of the complete graph K_n that does not contain a monochromatic complete sub-graph K_{r_i} in color i for each $1 \leq i \leq k$. The set of all such colorings is denoted $\mathcal{R}(r_1, \ldots, r_k; n)$. The Ramsey number $R(r_1, \ldots, r_k)$ is the least $n > 0$ such that no $(r_1, \ldots, r_k; n)$ coloring exists. In the multicolor case ($k > 2$), the only known value of a nontrivial Ramsey number is $R(3, 3, 3) = 17$. Prior to this paper, it was known that $30 \leq R(4, 3, 3) \leq 31$. Moreover, while the sets of $(3, 3, 3; n)$ colorings were known for $14 \leq n \leq 16$, the set of colorings for $n = 13$ was never published.[3] More information on recent results concerning Ramsey numbers can be found in the electronic dynamic survey by Radziszowski [15].

$$\varphi^{n,k}_{adj}(A) = \bigwedge_{1\leq q<r\leq n} \big(1 \leq A_{q,r} \leq k \quad \wedge \quad A_{q,r} = A_{r,q} \quad \wedge \quad A_{q,q} = 0\big) \tag{1}$$

$$\varphi^{n,c}_{r}(A) = \bigwedge_{I\in\wp_r([n])} \bigvee \{\ A_{i,j} \neq c \,|\, i, j \in I, i < j\ \} \tag{2}$$

$$\varphi_{(r_1,\ldots,r_k;n)}(A) = \varphi^{n,k}_{adj}(A) \wedge \bigwedge_{1\leq c\leq k} \varphi^{n,c}_{r_c}(A) \tag{3}$$

Fig. 1. Constraints for graph labeling problems: Ramsey colorings $(r_1, \ldots, r_k; n)$

A graph coloring problem on k colors is about the search for a graph coloring which satisfies a given set of constraints. Formally, it is specified as a formula, $\varphi(A)$, where A is an $n \times n$ adjacency matrix of integer variables with domain $\{0\} \cup [k]$ and φ is a constraint on these variables. A solution is an assignment of integer values to the variables in A which satisfies φ and determines both the graph edges and their colors. We often refer to a solution as an integer adjacency matrix and denote the set of solutions as $sol(\varphi(A))$.

Figure 1 presents the k-color graph coloring problems we focus on in this paper: $(r_1, \ldots, r_k; n)$ Ramsey colorings. Constraint (1), $\varphi^{n,k}_{adj}(A)$, states that the

[3] Recently, the set $\mathcal{R}(3, 3, 3; 13)$ has also been computed independently by: Stanislaw Radziszowski, Richard Kramer and Ivan Livinsky [14].

graph represented by matrix A has n vertices, is k colored, and is simple. Constraint (2) $\varphi_r^{n,c}(A)$ states that the $n \times n$ matrix A has no embedded sub-graph K_r in color c. Each conjunct, one for each set I of r vertices, is a disjunction stating that one of the edges between vertices of I is not colored c. Notation: $\wp_r(S)$ denotes the set of all subsets of size r of the set S. Constraint (3) states that A is a $(r_1, \ldots, r_k; n)$ Ramsey coloring.

For graph coloring problems, solutions are typically closed under permutations of vertices and of colors. Restricting the search space for a solution modulo such permutations is crucial when trying to solve hard graph coloring problems. It is standard practice to formalize this in terms of graph (coloring) isomorphism.

Let $G = (V, E)$ be a graph (coloring) with $V = [n]$ and let π be a permutation on $[n]$. Then $\pi(G) = (V, \left\{ (\pi(x), \pi(y)) \,\middle|\, (x, y) \in E \right\})$. Permutations act on adjacency matrices in the natural way: If A is the adjacency matrix of a graph G, then $\pi(A)$ is the adjacency matrix of $\pi(G)$ and $\pi(A)$ is obtained by simultaneously permuting with π both rows and columns of A.

Definition 1 ((weak) isomorphism of graph colorings). *Let (G, κ_1) and (H, κ_2) be k-color graph colorings with $G = ([n], E_1)$ and $H = ([n], E_2)$. We say that (G, κ_1) and (H, κ_2) are weakly isomorphic, denoted $(G, \kappa_1) \approx (H, \kappa_2)$ if there exist permutations $\pi \colon [n] \to [n]$ and $\sigma \colon [k] \to [k]$ such that $(u, v) \in E_1 \iff (\pi(u), \pi(v)) \in E_2$ and $\kappa_1((u, v)) = \sigma(\kappa_2((\pi(u), \pi(v))))$. We denote such a weak isomorphism: $(G, \kappa_1) \approx_{\pi,\sigma} (H, \kappa_2)$. When σ is the identity permutation, we say that (G, κ_1) and (H, κ_2) are isomorphic.*

The following lemma emphasizes the importance of weak graph isomorphism as it relates to Ramsey numbers. Many classic coloring problems exhibit the same property.

Lemma 1 ($\mathcal{R}(r_1, r_2, \ldots, r_k; n)$ is closed under $\approx$). *Let (G, κ_1) and (H, κ_2) be graph colorings in k colors such that $(G, \kappa_1) \approx_{\pi,\sigma} (H, \kappa_2)$. Then,*

$$(G, \kappa_1) \in \mathcal{R}(r_1, r_2, \ldots, r_k; n) \iff (H, \kappa_2) \in \mathcal{R}(\sigma(r_1), \sigma(r_2), \ldots, \sigma(r_k); n)$$

We make use of the following theorem from [12].

Theorem 1. $30 \leq R(4, 3, 3) \leq 31$ *and,* $R(4, 3, 3) = 31$ *if and only if there exists a* $(4, 3, 3; 30)$ *coloring* κ *of* K_{30} *such that: (1) For every vertex* v *and* $i \in \{2, 3\}$, $5 \leq deg_i(v) \leq 8$, *and* $13 \leq deg_1(v) \leq 16$. *(2) Every edge in the third color has at least one endpoint* v *with* $deg_3(v) = 13$. *(3) There are at least 25 vertices* v *for which* $deg_1(v) = 13$, $deg_2(v) = deg_3(v) = 8$.

Corollary 1. *Let* $G = (V, E)$ *be a* $(4, 3, 3; 30)$ *coloring,* $v \in V$ *a selected vertex, and assume without loss of generality that* $deg_2(v) \geq deg_3(v)$. *Then,* $deg(v) \in \left\{ \langle 13, 8, 8\rangle, \langle 14, 8, 7\rangle, \langle 15, 7, 7\rangle, \langle 15, 8, 6\rangle, \langle 16, 7, 6\rangle, \langle 16, 8, 5\rangle \right\}$.

Consider a vertex v in a $(4, 3, 3; n)$ coloring and focus on the three subgraphs induced by the neighbors of v in each of the three colors. The following states that these must be corresponding Ramsey colorings.

Observation 1 *Let G be a $(4,3,3;n)$ coloring and v be any vertex with $deg(v) = \langle d_1, d_2, d_3 \rangle$. Then, $d_1 + d_2 + d_3 = n - 1$ and G_v^1, G_v^2, and G_v^3 are respectively $(3,3,3;d_1)$, $(4,2,3;d_2)$, and $(4,3,2;d_3)$ colorings.*

Note that by definition a $(4,2,3;n)$ coloring is a $(4,3;n)$ Ramsey coloring in colors 1 and 3 and likewise a $(4,3,2;n)$ Ramsey coloring is a $(4,3;n)$ coloring in colors 1 and 2. This is because the "2" specifies that the coloring does not contain a subgraph K_2 in the corresponding color and this means that it contains no edge with that color. For $n \in \{14,15,16\}$, the sets $\mathcal{R}(3,3,3;n)$ are known and consist respectively of 115, 2, and 2 colorings. Similarly, for $n \in \{5,6,7,8\}$ the sets $\mathcal{R}(4,3;n)$ are known and consist respectively of 9, 15, 9, and 3 colorings.

In this paper computations are performed using the CryptoMiniSAT [16] SAT solver. SAT encodings (CNF) are obtained using the finite-domain constraint compiler BEE [10]. The use of BEE facilitates applications to find a single (first) solution, or to find all solutions for a constraint, modulo a specified set of variables. When solving for all solutions, our implementation iterates with the SAT solver, adding so called *blocking clauses* each time another solution is found. This technique, originally due to McMillan [9], is simplistic but suffices for our purposes. All computations were performed on a cluster with a total of 228 Intel E8400 cores clocked at 2 GHz each, able to run a total of 456 parallel threads. Each of the cores in the cluster has computational power comparable to a core on a standard desktop computer. Each SAT instance is run on a single thread.

3 Basic SAT Encoding and Embeddings

Throughout the paper we apply a SAT solver to solve CNF encodings of constraints such as those presented in Figure 1. In this way it is straightforward to find a Ramsey coloring or prove its non-existence. Ours is a standard encoding to CNF. To this end: nothing new. For an n vertex graph coloring problem in k colors we take an $n \times n$ matrix A where $A_{i,j}$ represents in k bits the edge (i,j) in the graph: exactly one bit is true indicating which color the edge takes, or no bit is true indicating that the edge (i,j) is not in the graph. Already at the representation level, we use the same Boolean variables to represent the color in $A_{i,j}$ and in $A_{j,i}$ for each $1 \leq i < j \leq n$. We further fix the variables corresponding to $A_{i,i}$ to *false*. The rest of the SAT encoding is straightforward.

Constraint (1) is encoded to CNF by introducing clauses to state that for each $A_{i,j}$ with $1 \leq i < j \leq n$ at most one of the k bits representing the color of the edge (i,j) is true. In our setting typically $k = 3$. For three colors, if b_1, b_2, b_3 are the bits representing the color of an edge, then three clauses suffice: $(\bar{b}_1 \vee \bar{b}_2), (\bar{b}_1 \vee \bar{b}_3), (\bar{b}_2 \vee \bar{b}_3)$. Constraint (2) is encoded by a single clause per set I of r vertices expressing that at least one of the bits corresponding to an edge between vertices in I does not have color c. Finally Constraint (3) is a conjunction of constraints of the previous two forms.

In Section 4 we will improve on this basic encoding by introducing symmetry breaking constraints (encoded to CNF). However, for now we note that, even with symmetry breaking constraints, using the basic encoding, a SAT solver is currently not able to solve any of the open Ramsey coloring problems such as those considered in this paper. In particular, directly applying a SAT solver to search for a $(4,3,3;30)$ Ramsey coloring is hopeless.

To facilitate the search for $(4,3,3;30)$ Ramsey coloring using a SAT encoding, we apply a general approach where, when seeking a $(r_1,\ldots,r_k;n)$ Ramsey coloring one selects a "preferred" vertex, call it v_1, and based on its degrees in each of the k colors, embeds k subgraphs which are corresponding smaller colorings. Using this approach, we apply Corollary 1 and Observation 1 to establish that a $(4,3,3;30)$ coloring, if one exists, must be $\langle 13,8,8\rangle$ regular. Specifically, all vertices must have 13 edges in the first color and 8 each, in the second and third colors. This result is considered significant progress in the research on Ramsey numbers [19]. This "embedding" approach is often applied in the Ramsey number literature where the process of completing (or trying to complete) a partial solution (an embedding) to a Ramsey coloring is called *gluing*. See for example the presentations in [13,5,12].

Theorem 2. *Any* $(4,3,3;30)$ *coloring, if one exists, is* $\langle 13,8,8\rangle$ *regular.*

Proof. By computation as described in the rest of this section.

We seek a $(4,3,3;30)$ coloring of K_{30}, represented as a 30×30 adjacency matrix A. Let v_1 correspond to the the first row in A with $deg(v_1)=\langle d_1,d_2,d_3\rangle$ as prescribed by Corollary 1. For each possible triplet $\langle d_1,d_2,d_3\rangle$, except $\langle 13,8,8\rangle$, we take each of the known corresponding colorings for the subgraphs $G^1_{v_1}$, $G^2_{v_1}$, and $G^3_{v_1}$ and embed them into A. We then apply a SAT solver, to (try to) complete the remaining cells in A to satisfy $\varphi_{4,3,3;30}(A)$ as defined by Constraint (3) of Figure 1. If the SAT solver fails, then no such completion exists.

$$\begin{bmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
1 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 1 & 0 & 2 & 2 & 3 & 1 & 1 & 3 & 3 & 1 & 2 & 2 & 3 & 3 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 1 & 2 & 0 & 3 & 2 & 1 & 2 & 1 & 3 & 1 & 1 & 3 & 2 & 3 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 1 & 2 & 3 & 0 & 2 & 2 & 1 & 3 & 1 & 1 & 3 & 1 & 3 & 2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 1 & 3 & 2 & 2 & 0 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 1 & 1 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 2 & 1 & 1 & 2 & 1 & 0 & 3 & 3 & 1 & 2 & 3 & 1 & 2 & 3 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 2 & 1 & 2 & 1 & 1 & 3 & 0 & 1 & 3 & 2 & 1 & 3 & 3 & 2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 2 & 3 & 1 & 3 & 2 & 3 & 1 & 0 & 1 & 3 & 2 & 3 & 2 & 1 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 2 & 3 & 3 & 1 & 2 & 1 & 3 & 1 & 0 & 3 & 3 & 2 & 1 & 2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 3 & 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 0 & 2 & 2 & 1 & 1 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 3 & 2 & 1 & 3 & 3 & 3 & 1 & 2 & 3 & 2 & 0 & 1 & 1 & 2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 3 & 2 & 3 & 1 & 3 & 1 & 3 & 3 & 2 & 2 & 1 & 0 & 2 & 1 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 3 & 3 & 2 & 3 & 1 & 2 & 3 & 2 & 1 & 1 & 1 & 2 & 0 & 2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 3 & 3 & 3 & 2 & 1 & 3 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 0 & 3 & 3 & 3 & 1 & 1 & 1 & 1 & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 3 & 0 & 1 & 1 & 3 & 3 & 1 & 1 & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 3 & 1 & 0 & 1 & 3 & 1 & 3 & 1 & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 3 & 1 & 1 & 0 & 1 & 1 & 3 & 3 & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 1 & 3 & 3 & 1 & 0 & 1 & 1 & 3 & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 1 & 3 & 1 & 1 & 1 & 0 & 3 & 3 & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 1 & 1 & 3 & 3 & 1 & 3 & 0 & 1 & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 1 & 1 & 1 & 3 & 3 & 3 & 1 & 0 & _ & _ & _ & _ & _ & _ & _ \\
3 & _ & 0 & 2 & 2 & 1 & 1 & 1 & 1 \\
3 & _ & 2 & 0 & 1 & 2 & 1 & 1 & 1 \\
3 & _ & 2 & 1 & 0 & 1 & 2 & 1 & 1 \\
3 & _ & 1 & 2 & 1 & 0 & 1 & 2 & 1 \\
3 & _ & 1 & 1 & 2 & 1 & 0 & 1 & 2 \\
3 & _ & 1 & 1 & 1 & 2 & 1 & 0 & 2 \\
3 & _ & 1 & 1 & 1 & 1 & 2 & 2 & 0
\end{bmatrix}$$

Fig. 2. One embedding to search for a $(4,3,3;30)$ coloring when $deg(v_1)=\langle 14,8,7\rangle$.

To illustrate the approach, consider the case where $deg(v_1)=\langle 14,8,7\rangle$. Figure 2 details one of the embeddings corresponding to this case. The first row and column of A specify the colors of the edges of the 29 neighbors of v_1 (in bold). The symbol "_" indicates an integer variable that takes a value between 1 and 3. The neighbors of v_1 in color 1 form a submatrix of A embedded in rows (and columns) 2–15 of the matrix in the Figure. By Corollary 1 these are a $(3,3,3;14)$ Ramsey coloring and there are 115 possible such colorings modulo weak isomorphism. The Figure details one of them. Similarly, there are 3 possible $(4,2,3;8)$ colorings which are subgraphs for the neighbors of v_1 in color 2. In Figure 2, rows (and columns) 16–23 detail one such coloring. Finally, there are 9 possible $(4,3,2;7)$ colorings which are subgraphs for the neighbors of v_1 in color 3. In Figure 2, rows (and columns) 24–30 detail one such coloring.

v_1 degrees	# instances	# clauses (avg.)	# vars (avg.)	unsat (avg)	unsat (total)
(16,8,5)	54 = 2*3*9	32432	5279	51 sec.	0.77 hrs.
(16,7,6)	270 = 2*9*15	32460	5233	420 sec.	31.50 hrs.
(15,8,6)	90 = 2*3*15	33607	5450	93 sec.	2.32 hrs.
(15,7,7)	162 = 2*9*9	33340	5326	1554 sec.	69.94 hrs.
(14,8,7)	3105 = 115*3*9	34069	5324	294 sec.	253.40 hrs.

Table 1. Proving that any $(4,3,3;30)$ Ramsey coloring is $\langle 13,8,8\rangle$ regular (summary).

To summarize, Figure 2 is a partially instantiated adjacency matrix. The first row determines the degrees of v_1, in the three colors, and 3 corresponding subgraphs are embedded. The uninstantiated values in the matrix must be completed to obtain a solution that satisfies $\varphi_{4,3,3;30}(A)$ as specified in Constraint (3) of Figure 1. This can be determined using a SAT solver. For the specific example in Figure 2, the CNF generated using our tool set consists of 33,959 clauses, involves 5,318 Boolean variables, and is shown to be unsatisfiable in 52 seconds of computation time. For the case where v_1 has degrees $\langle 14,8,7\rangle$ in the three colors this is one of $115 \times 3 \times 9 = 3105$ instances that need to be checked.

Table 1 summarizes the experiment which proves Theorem 2. For each of the possible degrees of vertex 1 in a $(4,3,3;30)$ coloring as prescribed by Corollary 1, except $\langle 13,8,8\rangle$, and for each possible choice of colorings for the derived subgraphs $G^1_{v_1}$, $G^2_{v_1}$, and $G^3_{v_1}$, we apply a SAT solver to show that the instance $\varphi_{(4,3,3;30)}(A)$ of Constraint (3) of Figure 1 cannot be satisfied. The table details for each degree triple, the number of instances, their average size (number of clauses and Boolean variables), and the average and total times to show that the constraint is not satisfiable.

All of the SAT instances described in the experiment summarized by Table 1 are unsatisfiable. The solver reports "unsat". To gain confidence in our implementation, we illustrate its application on a satisfiable instance: to find a, known to exist, $(4,3,3;29)$ coloring. This experiment involves some reverse engineering. In 1966 Kalbfleisch [7] reported the existence of a circulant $(3,4,4;29)$ coloring. Encoding instance $\varphi_{(4,3,3;29)}(A)$ of Constraint (3) together with a constraint that states that the adjacency matrix A is circulant, results in a CNF with 146,506 clauses and 8,394 variables. Using a SAT solver, we obtain a corresponding $(4,3,3;29)$ coloring in less than two seconds of computation time. The solution is $\langle 12,8,8\rangle$ regular and isomorphic to the adjacency matrix depicted as Figure 3. Now we apply the embedding approach. We take the partial solution (the boldface elements) corresponding to the three subgraphs: $G^1_{v_1}$, $G^2_{v_1}$ and $G^3_{v_1}$ which

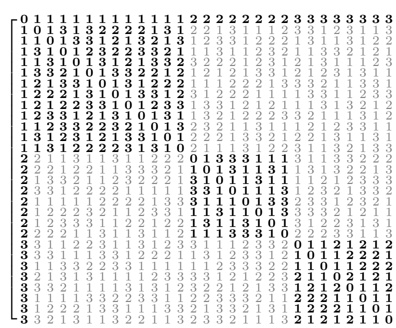

Fig. 3. Embedding (boldface) and solution (gray text) for a $(4,3,3;29)$ Ramsey coloring.

are respectively $(3,3,3;12)$, $(4,2,3;8)$ and $(4,3,2;8)$ Ramsey colorings. Applying a SAT solver to complete this partial solution to a $(4,3,3;29)$ coloring satisfying Constraint (3) involves a CNF with 30,944 clauses and 4,736 variables and requires under two hours of computation time. Figure 3 portrays the solution (the gray elements).

To apply the embedding approach described in this section to determine if there exists a $(4,3,3;30)$ Ramsey coloring which is $\langle 13,8,8\rangle$ regular would require access to the set $\mathcal{R}(3,3,3;13)$. We defer this discussion until after Section 7 where we describe how we compute the set of all 78,892 $(3,3,3;13)$ Ramsey colorings modulo weak isomorphism.

4 Symmetry Breaking: Computing $\mathcal{R}(r_1,\dots,r_k;n)$

In this section we prepare the ground to apply a SAT solver to find the set of all $(r_1,\dots,r_k;n)$ Ramsey colorings modulo weak isomorphism. The constraints are those presented in Figure 1 and their encoding to CNF is as described in Section 3. Our final aim is to compute the set of all $(3,3,3;13)$ colorings modulo weak isomorphism. Then we can apply the embedding technique of Section 3 to determine the existence of a $\langle 13,8,8\rangle$ regular $(4,3,3;30)$ Ramsey coloring. Given Theorem 2, this will determine the value of $R(4,3,3)$.

Solving hard search problems on graphs, and graph coloring problems in particular, relies heavily on breaking symmetries in the search space. When searching for a graph, the names of the vertices do not matter, and restricting the search modulo graph isomorphism is highly beneficial. When searching for a graph coloring, on top of graph isomorphism, solutions are typically closed under permutations of the colors: the names of the colors do not matter and the term often used is "weak isomorphism" [12] (the equivalence relation is weaker because both node names and edge colors do not matter). When the problem is to compute the set of all solutions modulo (weak) isomorphism the task is even more challenging. Often one first attempts to compute all the solutions of the coloring problem, and to then apply one of the available graph isomorphism tools, such as `nauty` [8] to select representatives of their equivalence classes modulo (weak) isomorphism. This is a *generate and test* approach. However, typically the number of solutions is so large that this approach is doomed to fail even though the number of equivalence classes itself is much smaller. The problem is that tools such as `nauty` apply after, and not during, generation. To this end, we follow [3] where Codish *et al.* show that the symmetry breaking approach of [2] holds also for graph coloring problems where the adjacency matrix consists of integer variables. This is a *constrain and generate approach.* But, as symmetry breaking does not break all symmetries, it is still necessary to perform some reduction using a tool like `nauty`.[4] This form of symmetry breaking is an important component in our methodology.

[4] Note that `nauty` does not directly handle edge colored graphs and weak isomorphism directly. We applied an approach called k-layering described by Derrick Stolee [17].

Instance	$\#\backslash_{\approx}$	no sym break				with sym break			
		vars	clauses	time	#	vars	clauses	time	#
(4,3;5)	9	10	15	0.02	322	24	85	0.01	13
(4,3;6)	15	15	35	0.35	2812	48	200	0.01	31
(4,3;7)	9	21	70	9.27	13842	85	390	0.01	45
(4,3;8)	3	28	126	19.46	17640	138	676	0.01	20
(3,3,3;16)	2	360	2160	∞	?	3328	17000	0.14	6
(3,3,3;15)	2	315	1785	∞	?	2707	13745	0.37	66
(3,3,3;14)	115	273	1456	∞	?	2169	10936	259.56	24635
(3,3,3;13)	?	234	1170	∞	?	1708	8540	∞	?

Table 2. Computing Ramsey colorings with and without the symmetry break Constraint (4) (time in seconds with 24 hr. timeout marked by ∞).

Definition 2. *[2]. Let A be an $n \times n$ adjacency matrix. Then,*

$$\mathsf{sb}^*_\ell(A) = \bigwedge \left\{ \ A_i \preceq_{\{i,j\}} A_j \,\middle|\, i < j \ \right\} \tag{4}$$

where $A_i \preceq_{\{i,j\}} A_j$ denotes the lexicographic order between the i^{th} and j^{th} rows of A (viewed as strings) omitting the elements at positions i and j (in both rows).

We omit the precise details of how Constraint (4) is encoded to CNF. In our implementation this is performed by the finite domain constraint compiler BEE and details can be found in [10]. Table 2 illustrates the impact of the symmetry breaking Constraint (4) on the search for the Ramsey colorings required in the proof of Theorem 2.

The first four rows in the table portray the required instances of the forms $(4, 3, 2; n)$ and $(4, 2, 3; n)$ which by definition correspond to $(4, 3; n)$ colorings (respectively in colors 1 and 3, and in colors 1 and 2). The next three rows correspond to $(3, 3, 3; n)$ colorings where $n \in \{14, 15, 16\}$. The last row illustrates our failed attempt to apply a SAT encoding to compute $\mathcal{R}(3, 3, 3; 13)$. The first column in the table specifies the instance. The column headed by "$\#\backslash_{\approx}$" specifies the known (except for the last row) number of colorings modulo weak isomorphism [15]. The columns headed by "vars" and "clauses" indicate, the numbers of variables and clauses in the corresponding CNF encodings of the coloring problems with and without the symmetry breaking Constraint (4). The columns headed by "time" indicate the time (in seconds) to find all colorings iterating with a SAT solver. The timeout assumed here is 24 hours. The column headed by "#" specifies the number of colorings found by iterated SAT solving.

In the first four rows, notice the impact of symmetry breaking which reduces the number of solutions by 1–3 orders of magnitude. In the next three rows the reduction is more acute. Without symmetry breaking the colorings cannot be computed within the 24 hour timeout. The sets of colorings obtained with symmetry breaking have been verified to reduce, using `nauty` [8], to the known number of colorings modulo weak isomorphism indicated in the second column.

5 Abstraction: Degree Matrices for Graph Colorings

This section introduces an abstraction on graph colorings defined in terms of *degree matrices*. The motivation is to solve a hard graph coloring problem by first searching for its degree matrices. Degree matrices are to graph coloring problems as degree sequences [4] are to graph search problems. A degree sequence is a monotonic nonincreasing sequence of the vertex degrees of a graph. A graphic sequence is a sequence which can be the degree sequence of some graph.

The idea underlying our approach is that when the combinatorial problem at hand is too hard, then possibly solving an abstraction of the problem is easier. In this case, a solution of the abstract problem can be used to facilitate the search for a solution of the original problem.

Definition 3 (degree matrix). *Let A be a graph coloring on n vertices with k colors. The* degree matrix *of A, denoted $dm(A)$ is an $n \times k$ matrix, M such that $M_{i,j} = deg_j(i)$ is the degree of vertex i in color j.*

Figure 4 illustrates the degree matrix of the graph coloring given as Figure 3. The three columns correspond to the three colors and the 29 rows to the 29 vertices. The degree matrix consists of 29 identical rows as the corresponding graph coloring is $\langle 12, 8, 8 \rangle$ regular.

$$\left.\begin{bmatrix} 12\ 8\ 8 \\ \vdots \\ 12\ 8\ 8 \end{bmatrix}\right\} \text{29 rows}$$

Fig. 4. A degree matrix.

A degree matrix M represents the set of graphs A such that $dm(A) = M$. Due to properties of weak-isomorphism (vertices as well as colors can be reordered) we can exchange both rows and columns of a degree matrix without changing the set of graphs it represents. In the rest of our construction we adopt a representation in which the rows and columns of a degree matrix are sorted lexicographically.

Definition 4 (lex sorted degree matrix). *For an $n \times k$ degree matrix M we denote by $lex(M)$ the smallest matrix with rows and columns in the lexicographic order (non-increasing) obtained by permuting rows and columns of M.*

Definition 5 (abstraction). *Let A be a graph coloring on n vertices with k colors. The* abstraction *of A to a degree matrix is $\alpha(A) = lex(dm(A))$. For a set $\mathcal{A}$ of graph colorings we denote $\alpha(\mathcal{A}) = \left\{ \ \alpha(A) \,\middle|\, A \in \mathcal{A} \ \right\}$.*

Note that if A and A' are weakly isomorphic, then $\alpha(A) = \alpha(A')$.

Definition 6 (concretization). *Let M be an $n \times k$ degree matrix. Then, $\gamma(M) = \left\{ \ A \,\middle|\, \alpha(A) = M \ \right\}$ is the set of graph colorings represented by M. For a set $\mathcal{M}$ of degree matrices we denote $\gamma(\mathcal{M}) = \cup \left\{ \ \gamma(M) \,\middle|\, M \in \mathcal{M} \ \right\}$.*

Let $\varphi(A)$ be a graph coloring problem in k colors on an $n \times n$ adjacency matrix, A. Our strategy to compute $\mathcal{A} = sol(\varphi(A))$ is to first compute an over-approximation $\mathcal{M}$ of degree matrices such that $\gamma(\mathcal{M}) \supseteq \mathcal{A}$ and to then use $\mathcal{M}$ to guide the computation of $\mathcal{A}$. We denote the set of solutions of the graph coloring problem, $\varphi(A)$, which have a given degree matrix, M, by $sol_M(\varphi(A))$. Then

$$sol(\varphi(A)) = \bigcup_{M \in \mathcal{M}} sol_M(\varphi(A)) \tag{5}$$

$$sol_M(\varphi(A)) = sol(\varphi(A) \wedge \alpha(A){=}M) \tag{6}$$

Equation (5) implies that, we can compute the solutions to a graph coloring problem $\varphi(A)$ by computing the independent sets $sol_M(\varphi(A))$ for any over approximation $\mathcal{M}$ of the degree matrices of the solutions of $\varphi(A)$. This facilitates the computation for two reasons: (1) The problem is now broken into a set of independent sub-problems for each $M \in \mathcal{M}$ which can be solved in parallel, and (2) The computation of each individual $sol_M(\varphi(A))$ is now directed using M.

The constraint $\alpha(A){=}M$ in the right side of Equation (6) is encoded to SAT by introducing (encodings of) cardinality constraints. For each row of the matrix A the corresponding row in M specifies the number of elements with value c (for $1 \leq c \leq k$) that must be in that row. We omit the precise details of the encoding to CNF. In our implementation this is performed by the finite domain constraint compiler BEE and details can be found in [10].

When computing $sol_M(\varphi(A))$ for a given degree matrix we can no longer apply the symmetry breaking Constraint (4) as it might constrain the rows of A in a way that contradicts the constraint $\alpha(A) = M$ in the right side of Equation (6). However, we can refine Constraint (4, to break symmetries on the rows of A only when the corresponding rows in M are equal. Then M can be viewed as inducing an ordered partition of A and Constraint (7) is, in the terminology of [2], a partitioned lexicographic symmetry break. In the following, M_i and M_j denote the i^{th} and j^{th} rows of matrix M.

$$\mathsf{sb}^*_\ell(A, M) = \bigwedge_{i<j} \big(\big(M_i = M_j \Rightarrow A_i \preceq_{\{i,j\}} A_j\big) \big) \tag{7}$$

The following refines Equation (6) introducing the symmetry breaking predicate.

$$sol_M(\varphi(A)) = sol(\varphi(A) \wedge (\alpha(A){=}M) \wedge \mathsf{sb}^*_\ell(A, M)) \tag{8}$$

To justify that Equations (6) and (8) both compute $sol_M(\varphi(A))$, modulo weak isomorphism, we must show that if $\mathsf{sb}^*_\ell(A, M)$ excludes a solution then there is another weakly isomorphic solution that is not excluded.

Theorem 3 (correctness of $\mathsf{sb}^*_\ell(A, M)$). *Let A be an adjacency matrix with $\alpha(A) = M$. Then, there exists $A' \approx A$ such that $\alpha(A') = M$ and $\mathsf{sb}^*_\ell(A', M)$ holds.*

6 Computing Degree Matrices for $R(3, 3, 3; 13)$

This section describes how we compute a set of degree matrices that approximate those of the solutions of instance $\varphi_{(3,3,3;13)}(A)$ of Constraint (3). We apply a strategy mixing SAT solving with brute-force enumeration as follows. The computation of the degree matrices is summarized in Table 3. In the first step, we compute bounds on the degrees of the nodes in any $R(3, 3, 3; 13)$ coloring.

Lemma 2. *Let A be an $R(3, 3, 3; 13)$ coloring then for every vertex x in A, and color $c \in \{1, 2, 3\}$, $2 \leq deg_c(x) \leq 5$.*

Proof. By solving instance $\varphi_{(3,3,3;13)}(A)$ of Constraint (3) seeking a graph with some degree less than 2 or greater than 5. The CNF encoding is of size 13,672 clauses with 2,748 Boolean variables and takes under 15 seconds to solve and yields an UNSAT result which implies that such a graph does not exist.

In the second step, we enumerate the degree sequences with values within the bounds specified by Lemma 2. Recall that the degree sequence of an undirected graph is the non-increasing sequence of its vertex degrees. Not every non-increasing sequence of integers corresponds to a degree sequence. A sequence that corresponds to a degree sequence is said to be graphical. The number of degree sequences of graphs with 13 vertices is 836,315 (see Sequence number `A004251` of The On-Line Encyclopedia of Integer Sequences published electronically at `http://oeis.org`). However, when the degrees are bound by Lemma 2 there are only 280.

Lemma 3. *There are 280 degree sequences with values between* 2 *and* 5*.*

Proof. Straightforward enumeration using the algorithm of Erdös and Gallai [4].

In the third step, we test the 280 degree sequences identified by Lemma 3 to determine which of them might occur as the left column in a degree matrix.

Lemma 4. *Let A be a $R(3,3,3;13)$ coloring and let $M = \alpha(A)$. Then, (a) the left column of M is one of the 280 degree sequences identified in Lemma 3; and (b) there are only 80 degree sequences from the 280 which are the left column of $\alpha(A)$ for some coloring A in $R(3,3,3;13)$.*

Proof. By solving instance $\varphi_{(3,3,3;13)}(A)$ of Constraint (3). For each degree sequence from Lemma 3, seeking a solution with that degree sequence in the first color. This involves 280 instances with average CNF size: 10861 clauses and 2215 Boolean variables. The total solving time is 375.76 hours and the hardest instance required about 50 hours. Exactly 80 of these instances were satisfiable.

In the fourth step we extend the 80 degree sequences identified in Lemma 4 to obtain all possible degree matrices.

Lemma 5. *Given the 80 degree sequences identified in Lemma 4 as potential left columns of a degree matrix, there are 11,933 possible degree matrices.*

Proof. By enumeration. For a degree matrix: the rows and columns are lex sorted, the rows must sum to 12, and the columns must be graphical (when sorted). We enumerate all such degree matrices and then select their smallest representatives under permutations of rows and columns. The computation requires a few seconds.

In the fifth step, we test the 11,933 degree matrices identified by Lemma 5 to determine which of them are the abstraction of some $R(3,3,3;13)$ coloring.

Lemma 6. *From the 11,933 degree matrices identified in Lemma 5, 999 are $\alpha(A)$ for a coloring A in $\mathcal{R}(3,3,3;13)$.*

Step	Notes	ComputationTimes	CNF Size	
1	compute degree bounds (Lemma 2) (1 instance, unsat)	12.52 sec.	#Vars 2748	#Clauses 13672
2	enumerate 280 possible degree sequences (Lemma 3)	Prolog, fast (seconds)		

3	test degree sequences (Lemma 4)
	(280 instances: 200 unsat, 80 sat)
		16.32 hrs.
		hardest: 1.34 hrs
			#Vars
			1215 (avg)
				#Clauses
				7729(avg)

4	enumerate 11,933 degree matrices (Lemma 5)	Prolog, fast (seconds)		
5	test degree matrices (Lemma 6) (11,933 instances: 10,934 unsat, 999 sat)	126.55 hrs. hardest: 0.88 hrs.	#Vars 1520 (avg)	#Clauses 7632 (avg)

Table 3. Computing the degree matrices for $\mathcal{R}(3,3,3;13)$ step by step.

Proof. By solving instance $\varphi_{(3,3,3;13)}(A)$ of Constraint (3) together with a given degree matrix to test if it is satisfiable. This involves 11,933 instances with average CNF size: 7632 clauses and 1520 Boolean variables. The total solving time is 126.55 hours and the hardest instance required 0.88 hours.

7 Computing $\mathcal{R}(3,3,3;13)$ from Degree Matrices

We describe the computation of the set $\mathcal{R}(3,3,3;13)$ starting from the 999 degree matrices identified in Lemma 6. Table 4 summarizes the two step experiment.

Step	Notes	Computation Times
1	compute all $(3,3,3;13)$ Ramsey colorings per degree matrix (999 instances, 129,188 solutions)	total: 136.31 hr. hardest: 4.3 hr.
2	reduce modulo $\approx$ (78,892 solutions)	`nauty`, fast (minutes)

Table 4. Computing $\mathcal{R}(3,3,3;13)$ step by step.

step 1: For each degree matrix we compute, using a SAT solver, all corresponding solutions of Equation (8), where $\varphi(A) = \varphi_{(3,3,3;13)}(A)$ of Constraint (3) and M is one of the 999 degree matrices identified in (Lemma 6). This generates in total 129,188 $(3,3,3;13)$ Ramsey colorings. Table 4 details the total solving time for these instances and the solving times for the hardest instance for each SAT solver. The largest number of graphs generated by a single instance is 3720.

step 2: The 129,188 $(3,3,3;13)$ colorings from step 1 are reduced modulo weak-isomorphism using `nauty` [8]. This process results in a set with 78,892 graphs.

We note that recently, the set $\mathcal{R}(3,3,3;13)$ has also been computed independently by Stanislaw Radziszowski, and independently by Richard Kramer and Ivan Livinsky [14].

8 There is no $\langle 13,8,8\rangle$ Regular $(4,3,3;30)$ Coloring

In order to prove that there is no $\langle 13,8,8\rangle$ regular $(4,3,3;30)$ coloring using the embedding approach of Section 3, we need to check that $78{,}892\times 3\times 3 = 710{,}028$

$$\left\{ \begin{array}{|cccccccc|}\hline 0&1&1&1&3&3&3&3\\ 1&0&3&3&1&1&3&3\\ 1&3&0&3&1&3&1&3\\ 1&3&3&0&3&3&1&1\\ 3&1&1&3&0&3&3&1\\ 3&1&3&3&3&0&1&1\\ 3&3&1&1&3&1&0&3\\ 3&3&3&1&1&1&3&0\\ \hline \end{array}, \begin{array}{|cccccccc|}\hline 0&1&1&1&3&3&3&3\\ 1&0&3&3&1&3&3&3\\ 1&3&0&3&3&1&1&3\\ 1&3&3&0&3&1&3&1\\ 3&1&3&3&0&1&1&3\\ 3&3&1&1&1&0&3&3\\ 3&3&1&3&1&3&0&1\\ 3&3&3&1&3&3&1&0\\ \hline \end{array}, \begin{array}{|cccccccc|}\hline 0&1&1&1&3&3&3&3\\ 1&0&3&3&1&3&3&3\\ 1&3&0&3&3&1&1&3\\ 1&3&3&0&3&1&3&1\\ 3&1&3&3&0&1&3&3\\ 3&3&1&1&1&0&3&3\\ 3&3&1&3&3&3&0&1\\ 3&3&3&1&3&3&1&0\\ \hline \end{array} \right\} \subseteq \left\{ \begin{array}{|cccccccc|}\hline 0&1&1&1&3&3&3&3\\ 1&0&3&3&1&\mathtt{A}&3&3\\ 1&3&0&3&\mathtt{A}&\mathtt{B}&1&3\\ 1&3&3&0&3&\mathtt{B}&\mathtt{A}&1\\ 3&1&\mathtt{A}&3&0&\mathtt{B}&\mathtt{C}&\mathtt{A}\\ 3&\mathtt{A}&\mathtt{B}&\mathtt{B}&\mathtt{B}&0&\mathtt{A}&\mathtt{A}\\ 3&3&1&\mathtt{A}&\mathtt{C}&\mathtt{A}&0&\mathtt{B}\\ 3&3&3&1&\mathtt{A}&\mathtt{A}&\mathtt{B}&0\\ \hline \end{array} \;\middle|\; \begin{array}{l} \mathtt{A},\mathtt{B},\mathtt{C} \in \{1,3\} \\ \mathtt{A} \neq \mathtt{B} \end{array} \right\}$$

Fig. 5. Approximating the three (4,2,3;8) colorings by a single matrix with constraints.

corresponding instances are unsatisfiable. These correspond to the elements in the cross product of $\mathcal{R}(3,3,3;13)$, $\mathcal{R}(4,2,3;8)$ and $\mathcal{R}(4,3,2)$.

To decrease the number of instances by a factor of 9, we approximate the three $(4,2,3;8)$ colorings by a single description as demonstrated in Figure 5. The constrained matrix on the right has four solutions which include the three $(4,2,3;8)$ colorings on the left. We apply a similar approach for the $(4,3,2;8)$ colorings. So, in fact we have a total of only 78,892 embedding instances to consider.

In addition to the constraints in Figure 1, we add constraints to specify that each row of the adjacency matrix has the prescribed number of edges in each color (13, 8 and 8). By application of a SAT solver, we have determined all 78,892 instances to be unsatisfiable. The average size of an instance is 36,259 clauses with 5187 variables. The total solving time is 128.31 years (running in parallel on 456 threads). The average solving time is 14 hours while the median is 4 hours. Only 797 instances took more than one week to solve. The worst-case solving time is 96.36 days. The two hardest instances are detailed in Appendix B. Table 5 specifies, in the second column, the total number of instances that can be shown unsatisfiable within the time specified in the first column. The third column indicates the increment in percentage (within 10 hours we solve 71.46%, within 20 hours we solve an additional 12.11%, etc). The last rows in the table indicate that there are 4 instances which require between 1500 and 2000 hours of computation, and 2 that require between 2000 and 2400 hours.

time (hrs)	# instances	% instances (Δ)
10	56,363	71.443 %
20	65,914	12.106 %
100	77,263	14.385 %
500	78,791	1.937 %
1000	78,869	0.099 %
1500	78,886	0.022 %
2000	78,890	0.005 %
2400	78,892	0.003 %

Table 5. Time required per instance for proof that there are no $(4,3,3;30)$ colorings with degrees $(13,8,8)$

9 Conclusion

We have applied SAT solving techniques together with a methodology using abstraction and symmetry breaking to construct a computational proof that the Ramsey number $R(4,3,3) = 30$. Our strategy is based on the search for a $(4,3,3;30)$ Ramsey coloring, which we show does not exist. This implies that $R(4,3,3) \leq 30$ and hence, because of known bounds, that $R(4,3,3) = 30$.

The precise value $R(4,3,3)$ has remained unknown for almost 50 years. We have applied a methodology involoving SAT solving, abstraction, and symmetry to compute $R(4,3,3) = 30$. We expect this methodology to apply to a range of other hard graph coloring problems.

The question of whether a computational proof constitutes a *proper* proof is a controversial one. Most famously the issue caused much heated debate after publication of the computer proof of the Four Color Theorem [1]. It is straightforward to justify an existence proof (i.e. a *SAT* result), as it is easy to verify that the witness produced satisfies the desired properties. Justifying an *UNSAT* result is more difficult. If nothing else, we are certainly required to add the proviso that our results are based on the assumption of a lack of bugs in the entire tool chain (constraint solver, SAT solver, C-compiler etc.) used to obtain them.

Most modern SAT solvers, support the option to generate a proof certificate for UNSAT instances (see e.g. [6]), in the DRAT format [18], which can then be checked by a Theorem prover. This might be useful to prove the lack of bugs originating from the SAT solver but does not offer any guarantee concerning bugs in the generation of the CNF. Moreover, the DRAT certificates for an application like that described in this paper are expected to be of unmanageable size.

Our proofs are based on two main "computer programs". The first was applied to compute the set $\mathcal{R}(3,3,3;13)$ with its 78,892 Ramsey colorings. The fact that at least two other groups of researchers (Stanislaw Radziszowski, and independently Richard Kramer and Ivan Livinsky) report having computed this set and quote [14] the same number of elements is reassuring. The second program, was applied to complete partially instantiated adjacency matrices, embedding smaller Ramsey colorings, to determine if they can be extended to Ramsey colorings. This program was applied to show the non-existence of a $(4,3,3;30)$ Ramsey coloring. Here we gain confidence from the fact that the same program does find Ramsey colorings when they are known to exist. For example, the $(4,3,3;29)$ coloring depicted as Figure 3.

All of the software used to obtain our results is publicly available, as well as the individual constraint models and their corresponding encodings to CNF. For details, see the appendix.

Acknowledgments

We thank Stanislaw Radziszowski for his guidance and comments which helped improve the presentation of this paper. In particular Stanislaw proposed to show that our technique is able to find the $(4,3,3;29)$ coloring depicted as Figure 3.

References

1. K. Appel and W. Haken. Every map is four colourable. *Bulletin of the American Mathematical Society*, 82:711–712, 1976.
2. M. Codish, A. Miller, P. Prosser, and P. J. Stuckey. Breaking symmetries in graph representation. In F. Rossi, editor, *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China*. IJCAI/AAAI, 2013.
3. M. Codish, A. Miller, P. Prosser, and P. J. Stuckey. Constraints for symmetry breaking in graph representation. Full version of [2] (in preparation)., 2014.
4. P. Erdös and T. Gallai. Graphs with prescribed degrees of vertices (in Hungarian). *Mat. Lapok*, pages 264–274, 1960. Available from `http://www.renyi.hu/~p_erdos/1961-05.pdf`.
5. S. E. Fettes, R. L. Kramer, and S. P. Radziszowski. An upper bound of 62 on the classical Ramsey number r(3, 3, 3, 3). *Ars Comb.*, 72, 2004.
6. M. Heule, W. A. H. Jr., and N. Wetzler. Bridging the gap between easy generation and efficient verification of unsatisfiability proofs. *Softw. Test., Verif. Reliab.*, 24(8):593–607, 2014.
7. J. G. Kalbfleisch. *Chromatic Graphs and Ramsey's Theorem*. PhD thesis, University of Waterloo, January 1966.
8. B. McKay. *nauty* user's guide (version 1.5). Technical Report TR-CS-90-02, Australian National University, Computer Science Department, 1990.
9. K. L. McMillan. Applying SAT methods in unbounded symbolic model checking. In E. Brinksma and K. G. Larsen, editors, *Computer Aided Verification, 14th International Conference, Proceedings*, volume 2404 of *Lecture Notes in Computer Science*, pages 250–264. Springer, 2002.
10. A. Metodi, M. Codish, and P. J. Stuckey. Boolean equi-propagation for concise and efficient SAT encodings of combinatorial problems. *J. Artif. Intell. Res. (JAIR)*, 46:303–341, 2013.
11. K. Piwakowski. On Ramsey number r(4, 3, 3) and triangle-free edge-chromatic graphs in three colors. *Discrete Mathematics*, 164(1-3):243–249, 1997.
12. K. Piwakowski and S. P. Radziszowski. $30 \leq R(3,3,4) \leq 31$. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 27:135–141, 1998.
13. K. Piwakowski and S. P. Radziszowski. Towards the exact value of the Ramsey number r(3, 3, 4). In *Proceedings of the 33-rd Southeastern International Conference on Combinatorics, Graph Theory, and Computing*, volume 148, pages 161–167. Congressus Numerantium, 2001. `http://www.cs.rit.edu/~spr/PUBL/paper44.pdf`.
14. S. P. Radziszowski. Personal communication. January, 2015.
15. S. P. Radziszowski. Small Ramsey numbers. *Electronic Journal of Combinatorics*, 1994. Revision #14: January, 2014.
16. M. Soos. CryptoMiniSAT, v2.5.1. `http://www.msoos.org/cryptominisat2`, 2010.
17. D. Stolee. Canonical labelings with nauty. Computational Combinatorics (Blog), Entry from September 20, 2012. `http://computationalcombinatorics.wordpress.com` (viewed October 2015).
18. N. Wetzler, M. Heule, and W. A. H. Jr. Drat-trim: Efficient checking and trimming using expressive clausal proofs. In C. Sinz and U. Egly, editors, *Theory and Applications of Satisfiability Testing, 17th International Conference, Proceedings*, volume 8561 of *Lecture Notes in Computer Science*, pages 422–429. Springer, 2014.
19. X. Xu and S. P. Radziszowski. On some open questions for Ramsey and Folkman numbers. *Graph Theory, Favorite Conjectures and Open Problems*, 2015. (to appear).

A Selected Proofs

Lemma 1. [$\mathcal{R}(r_1, r_2, \ldots, r_k; n)$ **is closed under** $\approx$]
Let (G, κ_1) and (H, κ_2) be graph colorings in k colors such that $(G, \kappa_1) \approx_{\pi,\sigma} (H, \kappa_2)$. Then,

$$(G, \kappa_1) \in \mathcal{R}(r_1, r_2, \ldots, r_k; n) \iff (H, \kappa_2) \in \mathcal{R}(\sigma(r_1), \sigma(r_2), \ldots, \sigma(r_k); n)$$

Proof (of Lemma 1). Assume that $(G, \kappa_1) \in \mathcal{R}(r_1, r_2, \ldots, r_k; n)$ and in contradiction that $(H, \kappa_2) \notin \mathcal{R}(\sigma(r_1), \sigma(r_2), \ldots, \sigma(r_k); n)$. Let R denote a monochromatic clique of size r_s in H and R^{-1} the inverse of R in G. From Definition 1, $(u, v) \in R \iff (\pi^{-1}(u), \pi^{-1}(v)) \in R^{-1}$ and $\kappa_2(u, v) = \sigma^{-1}(\kappa_1(u, v))$. Consequently R^{-1} is a monochromatic clique of size r_s in (G, κ_1) in contradiction to $(G, \kappa_1) \in \mathcal{R}(r_1, r_2, \ldots, r_k; n)$.

Theorem 3. [**correctness of** $\mathsf{sb}^*_\ell(A, M)$] Let A be an adjacency matrix with $\alpha(A) = M$. Then, there exists $A' \approx A$ such that $\alpha(A') = M$ and $\mathsf{sb}^*_\ell(A', M)$ holds.

Proof (of Theorem 3). Let $C = \left\{ A' \,\middle|\, A' \approx A \wedge \alpha(A') = M \right\}$. Obviously $C \neq \emptyset$ because $A \in C$ and therefore there exists a $A_{min} = min_{\preceq} C$. Therefore, $A_{min} \preceq A'$ for all $A' \in C$. Now we can view M as inducing an ordered partion on A: vertices u and v are in the same component if and only if the corresponding rows of M are equal. Relying on Theorem 4 from [2], we conclude that $\mathsf{sb}^*_\ell(A_{min}, M)$ holds.

B The Two Hardest Instances

The following partial adjacency matrices are the two hardest instances described in Section 8, from the total 78,892. Both include the constraints: $\mathtt{A}, \mathtt{B}, \mathtt{C}, \in \{1, 3\}$, $\mathtt{D}, \mathtt{E}, \mathtt{F} \in \{1, 2\}$, $\mathtt{A} \neq \mathtt{B}$, $\mathtt{D} \neq \mathtt{E}$. The corresponding CNF representations consist in 5204 Boolean variables (each), 36,626 clauses for the left instance and 36,730 for the right instance. SAT solving times to show these instances UNSAT are 8,325,246 seconds for the left instance and 7,947,257 for the right.

$$\begin{bmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
1 & 0 & 1 & 1 & 2 & 2 & 3 & 3 & 1 & 1 & 2 & 3 & 3 & 1 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 1 & 0 & 2 & 1 & 3 & 1 & 2 & 2 & 3 & 3 & 1 & 2 & 3 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 1 & 2 & 0 & 3 & 2 & 1 & 1 & 3 & 3 & 1 & 2 & 3 & 2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 2 & 1 & 3 & 0 & 1 & 2 & 3 & 1 & 1 & 3 & 3 & 2 & 2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 2 & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 2 & 1 & 3 & 3 & 1 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 3 & 1 & 1 & 2 & 1 & 0 & 2 & 2 & 3 & 3 & 2 & 1 & 3 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 3 & 2 & 1 & 3 & 2 & 2 & 0 & 3 & 3 & 2 & 1 & 1 & 1 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 1 & 2 & 3 & 1 & 3 & 2 & 3 & 0 & 2 & 3 & 1 & 1 & 3 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 1 & 3 & 3 & 1 & 2 & 3 & 3 & 2 & 0 & 1 & 3 & 1 & 2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 2 & 3 & 1 & 3 & 1 & 3 & 2 & 3 & 1 & 0 & 1 & 3 & 2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 3 & 1 & 2 & 3 & 3 & 2 & 1 & 1 & 3 & 1 & 0 & 2 & 3 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 3 & 2 & 3 & 2 & 3 & 1 & 1 & 1 & 1 & 3 & 2 & 0 & 3 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 1 & 3 & 2 & 2 & 1 & 3 & 1 & 3 & 2 & 2 & 3 & 3 & 0 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 0 & 1 & 1 & 1 & 3 & 3 & 3 & 3 & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 1 & 0 & 3 & 3 & 1 & \mathtt{A} & 3 & 3 & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 1 & 3 & 0 & 3 & \mathtt{A} & \mathtt{B} & 1 & 3 & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 1 & 3 & 3 & 0 & 3 & \mathtt{B} & \mathtt{A} & 1 & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 3 & 1 & \mathtt{A} & 3 & 0 & \mathtt{B} & \mathtt{C} & \mathtt{A} & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 3 & \mathtt{A} & \mathtt{B} & \mathtt{B} & \mathtt{B} & 0 & \mathtt{A} & \mathtt{A} & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 3 & 3 & 1 & \mathtt{A} & \mathtt{C} & \mathtt{A} & 0 & \mathtt{B} & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 3 & 3 & 3 & 1 & \mathtt{A} & \mathtt{A} & \mathtt{B} & 0 & _ & _ & _ & _ & _ & _ & _ & _ \\
3 & _ & 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\
3 & _ & 1 & 0 & 2 & 2 & 1 & \mathtt{D} & 2 & 2 \\
3 & _ & 1 & 2 & 0 & 2 & \mathtt{D} & \mathtt{E} & 1 & 2 \\
3 & _ & 1 & 2 & 2 & 0 & 2 & \mathtt{E} & \mathtt{D} & 1 \\
3 & _ & 2 & 1 & \mathtt{D} & 2 & 0 & \mathtt{E} & \mathtt{F} & \mathtt{D} \\
3 & _ & 2 & \mathtt{D} & \mathtt{E} & \mathtt{E} & \mathtt{E} & 0 & \mathtt{D} & \mathtt{D} \\
3 & _ & 2 & 2 & 1 & \mathtt{D} & \mathtt{F} & \mathtt{D} & 0 & \mathtt{E} \\
3 & _ & 2 & 2 & 2 & 1 & \mathtt{D} & \mathtt{D} & \mathtt{E} & 0
\end{bmatrix}
\quad
\begin{bmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
1 & 0 & 1 & 1 & 2 & 1 & 2 & 3 & 3 & 3 & 2 & 1 & 2 & 3 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 1 & 0 & 3 & 1 & 3 & 2 & 1 & 1 & 3 & 2 & 2 & 3 & 2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 1 & 3 & 0 & 3 & 2 & 1 & 2 & 3 & 1 & 2 & 3 & 2 & 1 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 2 & 1 & 3 & 0 & 1 & 1 & 3 & 2 & 3 & 1 & 2 & 3 & 2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 1 & 3 & 2 & 1 & 0 & 2 & 3 & 3 & 1 & 3 & 2 & 1 & 3 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 2 & 2 & 1 & 1 & 2 & 0 & 3 & 3 & 3 & 3 & 1 & 1 & 3 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 3 & 1 & 2 & 3 & 3 & 3 & 0 & 2 & 1 & 1 & 3 & 1 & 2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 3 & 1 & 3 & 2 & 3 & 3 & 2 & 0 & 1 & 2 & 1 & 3 & 1 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 3 & 3 & 1 & 3 & 1 & 3 & 1 & 1 & 0 & 2 & 3 & 2 & 2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 2 & 2 & 2 & 1 & 3 & 3 & 1 & 2 & 2 & 0 & 3 & 3 & 1 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 1 & 2 & 3 & 2 & 2 & 1 & 3 & 1 & 3 & 3 & 0 & 2 & 3 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 2 & 3 & 2 & 3 & 1 & 1 & 1 & 3 & 2 & 3 & 2 & 0 & 3 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
1 & 3 & 2 & 1 & 2 & 3 & 3 & 2 & 1 & 2 & 1 & 3 & 3 & 0 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 0 & 1 & 1 & 1 & 3 & 3 & 3 & 3 & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 1 & 0 & 3 & 3 & 1 & \mathtt{A} & 3 & 3 & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 1 & 3 & 0 & 3 & \mathtt{A} & \mathtt{B} & 1 & 3 & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 1 & 3 & 3 & 0 & 3 & \mathtt{B} & \mathtt{A} & 1 & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 3 & 1 & \mathtt{A} & 3 & 0 & \mathtt{B} & \mathtt{C} & \mathtt{A} & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 3 & \mathtt{A} & \mathtt{B} & \mathtt{B} & \mathtt{B} & 0 & \mathtt{A} & \mathtt{A} & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 3 & 3 & 1 & \mathtt{A} & \mathtt{C} & \mathtt{A} & 0 & \mathtt{B} & _ & _ & _ & _ & _ & _ & _ & _ \\
2 & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & _ & 3 & 3 & 3 & 1 & \mathtt{A} & \mathtt{A} & \mathtt{B} & 0 & _ & _ & _ & _ & _ & _ & _ & _ \\
3 & _ & 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\
3 & _ & 1 & 0 & 2 & 2 & 1 & \mathtt{D} & 2 & 2 \\
3 & _ & 1 & 2 & 0 & 2 & \mathtt{D} & \mathtt{E} & 1 & 2 \\
3 & _ & 1 & 2 & 2 & 0 & 2 & \mathtt{E} & \mathtt{D} & 1 \\
3 & _ & 2 & 1 & \mathtt{D} & 2 & 0 & \mathtt{E} & \mathtt{F} & \mathtt{D} \\
3 & _ & 2 & \mathtt{D} & \mathtt{E} & \mathtt{E} & \mathtt{E} & 0 & \mathtt{D} & \mathtt{D} \\
3 & _ & 2 & 2 & 1 & \mathtt{D} & \mathtt{F} & \mathtt{D} & 0 & \mathtt{E} \\
3 & _ & 2 & 2 & 2 & 1 & \mathtt{D} & \mathtt{D} & \mathtt{E} & 0
\end{bmatrix}$$

(1) `new_int`$(\mathtt{I}, \mathtt{c}_1, \mathtt{c}_2)$	declare integer: $\mathtt{c}_1 \leq \mathtt{I} \leq \mathtt{c}_2$
(2) `bool_array_or`$([\mathtt{X}_1, \ldots, \mathtt{X}_n])$	clause: $\mathtt{X}_1 \vee \mathtt{X}_2 \cdots \vee \mathtt{X}_n$
(3) `bool_array_sum_eq`$([\mathtt{X}_1, \ldots, \mathtt{X}_n],\ \mathtt{I})$	Boolean cardinality: $(\Sigma\ \mathtt{X}_i) = \mathtt{I}$
(4) `int_eq_reif`$(\mathtt{I}_1, \mathtt{I}_2,\ \mathtt{X})$	reified integer equality: $\mathtt{I}_1 = \mathtt{I}_2 \Leftrightarrow \mathtt{X}$
(5) `int_neq`$(\mathtt{I}_1, \mathtt{I}_2)$	$\mathtt{I}_1 \neq \mathtt{I}_2$
(6) `int_gt`$(\mathtt{I}_1, \mathtt{I}_2)$	$\mathtt{I}_1 > \mathtt{I}_2$

Fig. 6. Selected BEE constraints

C Making the Instances Available

The statistics from the proof that $R(4, 3, 3) = 30$ are available from the domain:

`http://cs.bgu.ac.il/~mcodish/Benchmarks/Ramsey334`.

Additionally, we have made a small sample (30) of the instances available. Here we provide instances with the degrees $\langle 13, 8, 8\rangle$ in the three colors. The selected instances represent the varying hardness encountered during the search. The instances numbered {27765, 39710, 42988, 36697, 13422, 24578, 69251, 39651, 43004, 75280} are the hardest, the instances numbered {4157, 55838, 18727, 43649, 26725, 47522, 9293, 519, 23526, 29880} are the median, and the instances numbered {78857, 78709, 78623, 78858, 28426, 77522, 45135, 74735, 75987, 77387} are the easiest. A complete set of both the BEE models and the DIMACS CNF files are available upon request. Note however that they weight around 50GB when zipped.

The files in `bee_models.zip` detail constraint models, each one in a separate file. The file named `r433_30_Instance#.bee` contains a single Prolog clause of the form

```
model(Instance#,Map,ListOfConstraints) :- {...details...} .
```

where `Instance#` is the instance number, `Map` is a partially instantiated adjacency matrix associating the unknown adjacency matrix cells with variable names, and `ListOfConstraints` are the finite domain constraints defining their values. The syntax is that of BEE, however the interested reader can easily convert these to their favorite fininte domain constraint language. Note that the Boolean values *true* and *false* are represented in BEE by the constants 1 and −1. Figure 6 details the BEE constraints which occur in the above mentioned models.

The files in `cnf_models.zip` correspond to CNF encodings for the constraint models. Each instance is associated with two files: `r433_30_instance#.dimacs` and `r433_30_instance#.map`. These consist respectively in a DIMACS file and a map file which associates the Booleans in the DIMACS file with the integer variables in a corresponding partially instantiated adjacency matrix. The map file specifies for each pair (i, j) of vertices a triplet $[B_1, B_2, B_3]$ of Boolean variables (or values) specifying the presence of an edge in each of the three colors. Each such B_i is either the name of a DIMACS variable, if it is greater than 1, or a truth value 1 (*true*), or −1 (*false*).