# $R(5,5) \leq 48$

VIGLEIK ANGELTVEIT[1] AND BRENDAN D. MCKAY[2]

[1]*Mathematical Sciences Institute;* [2]*Research School of Computer Science*
*Australian National University, Canberra, ACT 2601, Australia*

Abstract. We improve the upper bound on the Ramsey number $R(5,5)$ from $R(5,5) \leq 49$ to $R(5,5) \leq 48$. We also complete the catalogue of extremal graphs for $R(4,5)$.

## 1. Introduction

The Ramsey number $R(s,t)$ is defined to be the smallest $n$ such that every graph of order $n$ contains either a clique of $s$ vertices or an independent set of $t$ vertices.

**Theorem 1.1.** *The Ramsey number $R(5,5)$ is less than or equal to* 48.

The history of bounds on $R(5,5)$ was provided in [4]. The lower bound of 43 established constructively by Exoo [1] in 1989 is still the best. The previous best upper bound of 49 was proved by McKay and Radziszowski [4]. By Theorem 1.1 we now have $43 \leq R(5,5) \leq 48$.

The actual value of $R(5,5)$ is widely believed to be 43, because a lot of computer resources have been expended in an unsuccessful attempt to construct a Ramsey(5,5)-graph of order 43 [4]. Moreover, all attempts to expand the list of 656 known Ramsey(5,5)-graphs [4] of order 42 have failed. In this regard, we mention unpublished 2014 work of Lieby and the second author which computed every Ramsey(5,5)-graph which shares a 37-vertex subgraph with one of the just-mentioned 656. No additional Ramsey(5,5)-graphs of order 42 were found.

Our proof of Theorem 1.1 uses a combination of theory and computation. Because the computation is quite long and the possibility of subtle errors is ever-present, each author implemented and ran separate programs using somewhat different approaches.

## 2. Outline of the proof of Theorem 1.1

Let $\mathcal{R}(s,t,n)$ denote the set of isomorphism classes of graphs of order $n$ without an $s$-clique or independent $t$-set, and $\mathcal{R}(s,t) = \bigcup_n \mathcal{R}(s,t,n)$. The

*E-mail address*: vigleik.angeltveit@anu.edu.au, brendan.mckay@anu.edu.au.

main idea is that given any graph $F \in \mathcal{R}(5,5,48)$, a large subgraph of it must be obtained by gluing together two graphs in $\mathcal{R}(4,5,24)$ along a graph in $\mathcal{R}(3,5,d)$ for some $d$.

A list of 350,904 graphs in $\mathcal{R}(4,5,24)$ was compiled by McKay and Radziszowski [3] in 1995, and our first task was to complete their list. This was actually the most time-consuming part of the project; see Section 3.

**Theorem 2.1.** $|\mathcal{R}(4,5,24)| = 352{,}366$.

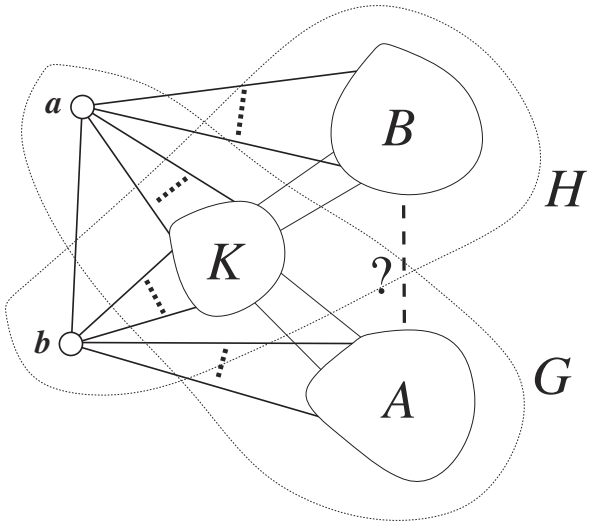


FIGURE 1. $G$ and $H$ are input graphs from $\mathcal{R}(4,5,24)$ that overlap in a graph $K$. The problem is to choose the edges between $A$ and $B$ so that the whole is in $\mathcal{R}(5,5)$.

We now explain the main idea for the proof of Theorem 1.1 in more detail. For a graph $F$, $VF$ is the vertex-set of $F$, $N_F(w)$ is the neighbourhood of vertex $w \in VF$ (which doesn't include $w$) and $F[W]$ is the subgraph of $F$ induced by $W \subseteq VF$. First, note that because $R(4,5) = 25$ [3], every vertex in a graph $F \in \mathcal{R}(5,5,48)$ must have degree 23 or 24. By replacing $F$ by its complement if necessary we can assume that $F$ has at least 24 vertices of degree 24. Hence $F$ must have two adjacent vertices $a, b$ of degree 24. Define

$$\begin{aligned} G &= F[N_F(b)], \\ H &= F[N_F(a)], \\ K &= F[VG \cap VH]. \end{aligned}$$

In words, $G$ is the subgraph of $F$ induced by the 24 vertices adjacent to $b$ (this includes $a$ but not $b$), $H$ is the subgraph induced by the vertices adjacent to $a$, and $K$ is the intersection of $G$ and $H$; see Figure 1. Note that $G, H \in \mathcal{R}(4,5,24)$ and that $K \in \mathcal{R}(3,5,d)$ for some $d$. Because $R(3,5) = 14$ we

must have $d \leq 13$, and $d$ is also equal to the degree of $a$ in $G$ and the degree of $b$ in $H$.

To reconstruct $F[VG \cup VH]$, which is a graph with $48 - d$ vertices, from $G$, $H$ and $K$, it suffices to specify how $K$ is a subgraph of $G$ and $H$, and whether or not we have an edge between $x$ and $y$ for all $x \in VG - VK - \{a\}$ and $y \in VH - VK - \{b\}$; i.e. between the parts labelled $A$ and $B$ in Figure 1. We call this procedure *gluing $G$ to $H$ along $K$*. For each inclusion of $K$ into $G$ and $H$ there are $2^{(23-d)^2}$ ways of gluing $G$ to $H$ along $K$, but we will only consider gluings that could give a graph in $\mathcal{R}(5, 5, 48 - d)$.

For $K \in \mathcal{R}(3, 5, d)$, define

$$\mathcal{R}(4, 5, 24, K) = \{(G, a) \mid G \in \mathcal{R}(4, 5, 24), a \in VG, G[N_G(a)] \cong K\},$$

where $\cong$ denotes isomorphism. We will call $(G, a)$ a *pointed graph of type $K$*. Our proof of Theorem 1.1 consists of the following steps.

**Step 1:** We completed the list of graphs in $\mathcal{R}(4, 5, 24)$ compiled by McKay and Radziszowski, thereby proving Theorem 2.1. This was done by a straightforward (but computationally expensive) extension of the method in [3]. While that calculation would have taken too long in 1995, it was doable in 2016.

**Step 2:** For each $K \in \mathcal{R}(3, 5, d)$ with $d \leq 11$ and for each pair $(G, a), (H, b) \in \mathcal{R}(4, 5, 24, K)$, we used computer programs to calculate all ways of gluing $G$ to $H$ along $K$. Due to automorphisms of $K$, there can be more than one distinct way to overlap $G$ and $H$ so that their intersection is $K$, so several gluing computations can be required for given $G, H, K$.

**Step 3:** For each graph generated in Step 2, we used another program which attempts in all possible ways to add one vertex while staying within $\mathcal{R}(5, 5)$. Since this was never possible, none of the graphs generated in Step 2 is a subgraph of a graph in $\mathcal{R}(5, 5, 48)$.

**Lemma 2.2.** *Execution of Steps 1–3 is sufficient to prove Theorem 1.1.*

*Proof.* Suppose $F \in \mathcal{R}(5, 5, 48)$. We first prove that either $F$ or its complement has a vertex of degree 24 adjacent to at least 12 other vertices of degree 24. Suppose that $F$ is a counterexample to this claim, and let $W \subseteq VF$ be its vertices of degree 24. Since $F[W]$ has maximum degree 11, there are at least $e_1 = 13\,|W|$ edges between $W$ and $VF \setminus W$ in $F$. Similarly, there are at least $e_2 = 13\,(48 - |W|)$ edges between $W$ and $VF \setminus W$ in $\bar{F}$. However, this is impossible since $e_1 + e_2 = 13 \times 48 = 624$ and $|W|(48 - |W|) \leq 24^2 = 576$.

So let $b$ be a vertex of $F$ of degree 24 that is adjacent to at least 12 other vertices of degree 24 and define $G = F[N_F(b)]$. From the $\mathcal{R}(4, 5, 24)$ catalogue we find that $G$ has at most 8 vertices of degree more than 11, so we can choose $a \in N_F(b)$ that has degree 24 in $F$ and degree at most 11 in $G$. Define $H = F[N_F(a)]$. Then the gluing of $(G, a)$ and $(H, b)$ in Step 2 will find a subgraph of $F$ and the failure of one point extension in Step 3 will show that $F$ doesn't exist. □

| $e$ | $i_3$ | $i_4$ | $c_3$ | $\delta$ | $\Delta$ | count |
|---|---|---|---|---|---|---|
| 116 | 356–368 | 225–262 | 123–128 | 8–9 | 10–11 | 9 |
| 117 | 346–362 | 216–253 | 122–132 | 8–9 | 10–11 | 90 |
| 118 | 340–360 | 206–251 | 120–136 | 6–9 | 10–12 | 806 |
| 119 | 332–356 | 198–247 | 124–140 | 6–9 | 10–13 | 4358 |
| 120 | 324–352 | 186–243 | 127–144 | 6–10 | 10–13 | 16346 |
| 121 | 319–344 | 181–232 | 130–146 | 6–10 | 11–13 | 43457 |
| 122 | 314–337 | 178–223 | 133–149 | 6–10 | 11–13 | 79678 |
| 123 | 310–330 | 171–215 | 136–152 | 6–10 | 11–13 | 92504 |
| 124 | 304–324 | 163–208 | 140–154 | 6–10 | 11–13 | 67209 |
| 125 | 302–318 | 161–201 | 144–157 | 6–10 | 11–13 | 31996 |
| 126 | 296–312 | 155–195 | 147–160 | 7–10 | 11–12 | 11485 |
| 127 | 291–301 | 152–177 | 152–162 | 8–10 | 11–12 | 3401 |
| 128 | 286–296 | 149–171 | 156–164 | 8–10 | 11–12 | 843 |
| 129 | 281–290 | 146–165 | 162–166 | 9–10 | 11–12 | 147 |
| 130 | 276–282 | 143–155 | 166–169 | 9–10 | 11–12 | 32 |
| 131 | 270–270 | 143–149 | 172–172 | 10–10 | 11–11 | 3 |
| 132 | 264–264 | 138–144 | 176–176 | 11–11 | 11–11 | 2 |
| all | 264–368 | 138–262 | 120–176 | 6–11 | 10–13 | 352366 |

TABLE 1. Statistics for all $(4, 5, 24)$-graphs

## 3. Step 1: Completing the list of graphs in $\mathcal{R}(4, 5, 24)$

McKay and Radziszowski [3] produced a list of 350,904 such graphs, and proved that the list contains all graphs in $\mathcal{R}(4, 5, 24)$ with minimum degree 6, 7 or 8, or maximum degree 12 or 13, or if the graph is regular of degree 11.

To complete the catalogue it suffices to find those graphs with minimum degree 9 or 10. We did this using the well-tested code from [3] to glue together graphs of type $\mathcal{R}(3, 5, 9)$ and $\mathcal{R}(4, 4, 14)$, and of types $\mathcal{R}(3, 5, 10)$ and $\mathcal{R}(4, 4, 13)$. Although this requires a very large number of graph pairs to be glued, it is feasible when the graphs of type $\mathcal{R}(3, 5, 9)$ and $\mathcal{R}(3, 5, 10)$ are arranged in a tree structure that exhibits common subgraphs and symmetries. See [3] for details. All graphs in $\mathcal{R}(4, 5, 24)$ with a vertex of degree 9 or 10 were found, to increase the overlap with [3] for checking purposes. This took about 1.5 core-years of computer time and discovered 1462 new graphs in $\mathcal{R}(4, 5, 24)$; recall that the search in [3] was not intended to be complete.

Although there was no reason to suspect the catalogue of $\mathcal{R}(4, 5, 24)$ was not now complete, we devoted another 6 core-months to checking it. We mention two such checks. In the first, let $\mathcal{A}'$ be the set of all neighbourhoods of a vertex of degree 9 or 10 in the 1462 new graphs, and let $\mathcal{B}'$ be the set of all complementary neighbourhoods of the same vertices in those graphs. Then, using a completely separate program, we constructed all graphs in $\mathcal{R}(4, 5, 24)$ with a vertex having a neighbourhood in $\mathcal{A}'$ and a complementary

neighbourhood in $\mathcal{B}'$. Only known graphs appeared. For the second check, we computed all Ramsey(4,5)-graphs that share a 21-vertex subgraph with a known Ramsey(4,5,24)-graph. Again, no additional Ramsey(4,5,24)-graphs appeared.

Summary statistics of the catalogue, to complete [3, Table 4], are provided in Table 1; $e$ is the number of edges, $i_k$ is the number of independent sets of size $k$, $c_3$ is the number of triangles, and $\delta$, $\Delta$ are the minimum and maximum degrees. The graphs themselves are available at [2].

## 4. The structure of $\mathcal{R}(4, 5, 24, K)$

The neighbourhood of a vertex $a$ of degree $d$ in a pointed graph $(G, a) \in \mathcal{R}(4, 5, 24, K)$ is the graph $K \in \mathcal{R}(3, 5, d)$. However not all graphs in $\mathcal{R}(3, 5)$ appear in pointed graphs.

In Table 2, we show the number of graphs $K$ which occur at least once and the total number of pointed graphs for each $d$. Each graph $G \in \mathcal{R}(4, 5, 24)$ contributes 24 to the table, even though automorphisms of $G$ mean that some of the 24 pointed graphs formed from $G$ may be isomorphic. The great majority of graphs in $\mathcal{R}(4, 5, 24)$ have trivial automorphism group, so we gave up the small available speedup (estimated at 3%) in order to have fewer steps in the computation.

| $d$ | $\lvert\mathcal{R}(3, 5, d)\rvert$ | occurring | count |
|---|---|---|---|
| 1–5 | 21 | 0 | 0 |
| 6 | 32 | 2 | 1979 |
| 7 | 71 | 11 | 7497 |
| 8 | 179 | 88 | 64395 |
| 9 | 290 | 240 | 832288 |
| 10 | 313 | 294 | 4651124 |
| 11 | 105 | 103 | 2800499 |
| 12 | 12 | 11 | 97968 |
| 13 | 1 | 1 | 1034 |
| all | 1029 | 750 | 8456784 |

Table 2. Overcounts (see text) of pointed graphs

The number of pointed graphs in $\mathcal{R}(4, 5, 24, K)$ for $K \in \mathcal{R}(3, 5, \leq 11)$ varies greatly: from 0 to 526,073, the latter from a rather irregular graph of order 11 and 21 edges. For Step 2 we take two pointed graphs $(G, a)$, $(H, b) \in \mathcal{R}(4, 5, 24, K)$ and overlap them so that their common subgraph $K$ coincides. This can be done in one distinct way for each automorphism of $K$ (again ignoring some small reductions arising from automorphisms of $G$ and $H$). Most graphs $K$ have only trivial automorphisms but some have large automorphism groups, the largest having order 1152 (a vertex-transitive quartic graph of order 8).

Taking the wildly varying sizes of $\mathcal{R}(4,5,24,K)$ as well as the automorphism groups of the various $K$ into account we needed to solve approximately 2 trillion gluing problems. While that is certainly a lot, we were able to perform hundreds of thousands of such gluings per second per core. This part of the computation took approximately six core-months for one implementation and two core-months for the other.

## 5. Step 2. Finding all ways to glue

In order to ensure correctness, the list of pointed graphs was prepared independently by the two authors and all the gluings were performed by two programs written independently using different methods.

Now we will describe the two different methods for gluing $(G,a),(H,b) \in \mathcal{R}(4,5,24,K)$ after they are overlapped at the common subgraph $K$. Because of the large number of calculations needed, the naive approach of deciding one unknown adjacency at a time takes far too long.

Define $d' = 23 - d$. Suppose $K$ has vertices $v_0,\dots,v_{d-1}$, $G$ has vertices $v_0,\dots,v_{d-1},a,a_1,\dots,a_{d'}$ and $H$ has vertices $v_0,\dots,v_{d-1},b,b_1,\dots,b_{d'}$. Note that the vertices $a$ and $b$ cannot participate in any 5-cliques or independent 5-sets by the construction. To specify a gluing it suffices to specify whether or not $a_i$ and $b_j$ are connected by an edge for $1 \le i,j \le d'$. We will record this data in a $d' \times d'$ matrix $M$ with entries 0 (for no edge) and 1 (for edge).

Define a *potential* $(r,s,t)$*-clique* to be $r$ vertices $w_1,\dots,w_r$ in $VK$, $s$ vertices $x_1,\dots,x_s$ in $VG - VK - \{a\}$, and $t$ vertices $y_1,\dots,y_t$ in $VH - VK - \{b\}$ such that

$$\{w_1,\dots,w_r,x_1,\dots,x_s\}$$

is an $(r+s)$-clique in $G$ and

$$\{w_1,\dots,w_r,y_1,\dots,y_t\}$$

is an $(r+t)$-clique in $H$. Note that $r+s, r+t \le 3$, since $G,H \in \mathcal{R}(4,5)$. Define a *potential independent* $(r,s,t)$*-set* similarly. The following lemma is immediate.

**Lemma 5.1.** *A $d' \times d'$ 0-1 matrix $M = (m_{ij})$ defines a gluing if and only if*

1. *For each potential $(r,s,t)$-clique with $r+s+t = 5$, $m_{x_i y_j} = 0$ for some $1 \le i \le s, 1 \le j \le t$. (This is needed for potential $(r,s,t)$-cliques $(1,2,2)$, $(0,2,3)$ and $(0,3,2)$.)*
2. *For each potential independent $(r,s,t)$-set with $r+s+t = 5$, $m_{x_i y_j} = 1$ for some $1 \le i \le s, 1 \le j \le t$. (This is needed for potential independent $(r,s,t)$-sets $(3,1,1)$, $(2,1,2)$, $(2,2,1)$, $(1,1,3)$, $(1,2,2)$, $(1,3,1)$, $(0,2,3)$ and $(0,3,2)$.)*

*Proof.* Please refer to Figure 1 and consider a set $W$ of size 5. For $W$ to be a clique in the completed graph, it must overlap both $K \cup A$ and $K \cup B$, and both $(K \cup A) \cap W$ and $(K \cup B) \cap W$ must be cliques. That implies $W$ is one of the potential $(r,s,t)$-cliques listed in part (1), and to prevent $W$

from being a clique in the completed graph we need to include a non-edge in $(A \times B) \cap W$. The case of an independent set is similar. $\square$

The two gluing methods are logically similar but were implemented very differently. The first gluing method expands on the method in [3]. Define an *interval* to be a set of the form $I = \{X \mid B \subseteq X \subseteq T\}$, where $B$ and $T$ are subsets of $\{a_1, \ldots, a_{d'}\} \times \{b_1, \ldots, b_{d'}\}$. We write $I = [B, T]$. We represent $I$ by two $d' \times d'$ matrices with coefficients in $\{0, 1\}$.

Given an interval $[B, T]$, we define collapsing rules as follows. There are 11 in total, one for each of the triples $(r, s, t)$ in Lemma 5.1 above. The special event FAIL means that there is no $X \in [B, T]$ which corresponds to a proper gluing.

**Rule** $K_{1,2,2}$. Suppose $\{w_1, x_1, x_2, y_1, y_2\}$ is a potential $(1, 2, 2)$-clique.

**if** $(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2) \in B$ **then** FAIL
**else if** $(x_1, y_1), (x_1, y_2), (x_2, y_1) \in B$ **then** $T := T - (x_2, y_2)$
**else if** $(x_1, y_1), (x_1, y_2), (x_2, y_2) \in B$ **then** $T := T - (x_2, y_1)$
**else if** $(x_1, y_1), (x_2, y_1), (x_2, y_2) \in B$ **then** $T := T - (x_1, y_2)$
**else if** $(x_1, y_2), (x_2, y_1), (x_2, y_2) \in B$ **then** $T := T - (x_1, y_1)$
**end if**

The collapsing rules for $K_{0,2,3}$ and $K_{0,3,2}$ are similar. In each case, the rule says that if 5 vertices include 9 edges, then the remaining vertex pair must not be an edge.

**Rule** $E_{3,1,1}$. Suppose $\{w_1, w_2, w_3, x_1, y_1\}$ is a potential independent $(3, 1, 1)$-set.

**if** $(x_1, y_1) \notin T$ **then** FAIL
**else** $B := B \cup (x_1, y_1)$.
**end if**

The collapsing rules for the other potential independent sets from Lemma 5.1 are similar.

We start the search with a single interval $I = [B, T]$ with $B = \varnothing$ and $T = \{a_1, \ldots, a_{d'}\} \times \{b_1, \ldots, b_{d'}\}$. It was commonly true that collapsing rule $E_{3,1,1}$ could be applied immediately. Each time we add an edge to $B$ or remove an edge from $T$ the number of possible gluings is cut in half.

Given an interval $[B, T]$, we apply the collapsing rules repeatedly until either FAIL occurs or a stable situation is reached in which FAIL has not occurred but no further collapsing rules apply. By the discussion in [3], which is valid here too, the final state is independent of the order of the application of the collapsing rules.

In the case that FAIL is reached, we have proved that no solution lies in $[B, T]$. Suppose on the other hand that FAIL was not reached and that $[B', T']$ is the final stable state of the interval. Then we can infer that any solution which lies in $[B, T]$ also lies in $[B', T']$. If $B' = T'$ there is exactly one solution, which we can output. Otherwise, we pick some $(a_i, b_j) \in$

$T' - B'$ and split $[B', T']$ into the two disjoint intervals $[B', T' - (a_i, b_j)]$ and $[B' \cup (a_i, b_j), T']$, which are then processed independently by the same method.

The second gluing method applies an equivalent procedure using data structures familiar from the constraint satisfaction discipline. Each entry $m_{ij}$ of $M$ is a *variable*, with value FALSE, TRUE or UNKNOWN, meaning an edge, a non-edge, or undecided status. Each set $\{x_1, \ldots, x_s\} \times \{y_1, \ldots, y_t\}$ corresponding to a potential $(r, s, t)$-clique or independent $(r, s, t)$-set is a *clause*. To avoid 5-cliques and independent 5-sets, clauses from potential $(r, s, t)$-cliques can't have all their variables TRUE, while clauses from potential independent $(r, s, t)$-sets can't have all their variables FALSE.

Each variable $\alpha$ is associated with a list $\mathcal{C}(\alpha)$ of the clique clauses which contain $\alpha$, and a list $\mathcal{I}(\alpha)$ of the independent set clauses which contain $\alpha$. There is also a stack $S$ which maintains a set of distinct variables on a last-in first-out basis. Informally, at each moment $S$ contains those variables which have been assigned FALSE or TRUE, but their clause lists have not yet been scanned.

```
while S ≠ ∅ do
    Pop the top variable α off S
    if α = FALSE then
        for each clause C ∈ I(α) do
            if all variables in C are FALSE then
                exit FAIL
            else if all variables in C are FALSE
                                except for β = UNKNOWN then
                Set β := TRUE and push β onto S
            end if
        end for
    else
        for each clause C ∈ C(α) do
            if all variables in C are TRUE then
                exit FAIL
            else if all variables in C are TRUE
                                except for β = UNKNOWN then
                Set β := FALSE and and push β onto S
            end if
        end for
    end if
end while
```

FIGURE 2. Second algorithm for gluing

Initially, variables are initialised to TRUE if they are the (sole) variable in a potential independent $(3, 1, 1)$-set. All other variables are initialised to

UNKNOWN. The variables equal to TRUE are put onto $S$ (which will be empty if there are no TRUE variables at this stage). Then we execute the algorithm of Figure 2 until it terminates.

Note that variables are assigned TRUE or FALSE as they are put onto $S$. Although it would be mathematically valid to delay the assignment until the variable is popped off the stack, the experimental efficiency is severely degraded.

If the algorithm terminates with "**exit** FAIL", there is no solution. Otherwise, all the variables with value FALSE or TRUE have those values in all solutions, so if there are no variables with value UNKNOWN we have found a unique solution. On the other hand, if the algorithm terminates without "**exit** FAIL" and some variable $\alpha$ has value UNKNOWN, we divide the problem into two subproblems according to whether $\alpha$ is TRUE or FALSE. In both cases, all other variables keep their present values and $S$ is initialisd to $\{\alpha\}$. The subproblems are processed by the same method, leading to further subproblems, and so on until all subproblems have been completed.

Both methods were very fast for $d \geq 8$, often performing 100,000 gluings per second per core, primarily because failure occurred early most of the time.

For $d \leq 7$, the methods as described could take much longer since extremely large search trees with many useless branches could be generated. For those values of $d$ we used additional techniques.

For the first method, two techniques were used. First, for each pair $(a_i, b_j) \in T - B$ we applied the collapsing rules to both $[B, T - (a_i, b_j)]$ and $[B \cup (a_i, b_j), T]$. If for some pair $(a_i, b_j)$ we arrived at FAIL in both cases we then concluded that there were no gluings. If $[B, T - (a_i, b_j)]$ led to FAIL then we replaced $[B, T]$ by $[B \cup (a_i, b_j), T]$, and if $[B \cup (a_i, b_j), T]$ led to FAIL then we replaced $[B, T]$ by $[B, T - (a_i, b_j)]$. This is of course more expensive than the original algorithm at each node of the search tree, but we found that for $6 \leq d \leq 7$ it was worth it.

Second, we ordered the pairs $(a_i, b_j)$ according to how many independent sets of type $(2, 2, 1)$ and $(2, 1, 2)$ they were contained in and started the binary search with a pair $(a_i, b_j)$ which was maximal in this sense. The advantage is that when considering $[B, T-(a_i, b_j)]$ the collapsing rules $E_{2,2,1}$ and $E_{2,1,2}$, which require only a single edge to be missing from $T$ in order to modify $B$, come into play as much as possible.

For the second method, instead of choosing an arbitrary UNKNOWN variable to branch on, we used an UNKNOWN variable which occurred in the greatest number of clique clauses with all TRUE variables except two UNKNOWN variables, or independent set clauses with all FALSE variables except two UNKNOWN variables. This is a heuristic for how beneficial it is to assign FALSE or TRUE to the variable.

In both cases, these enhancements made the cost per node of the search tree much greater but this was more than compensated for by the reduction in the number of nodes.

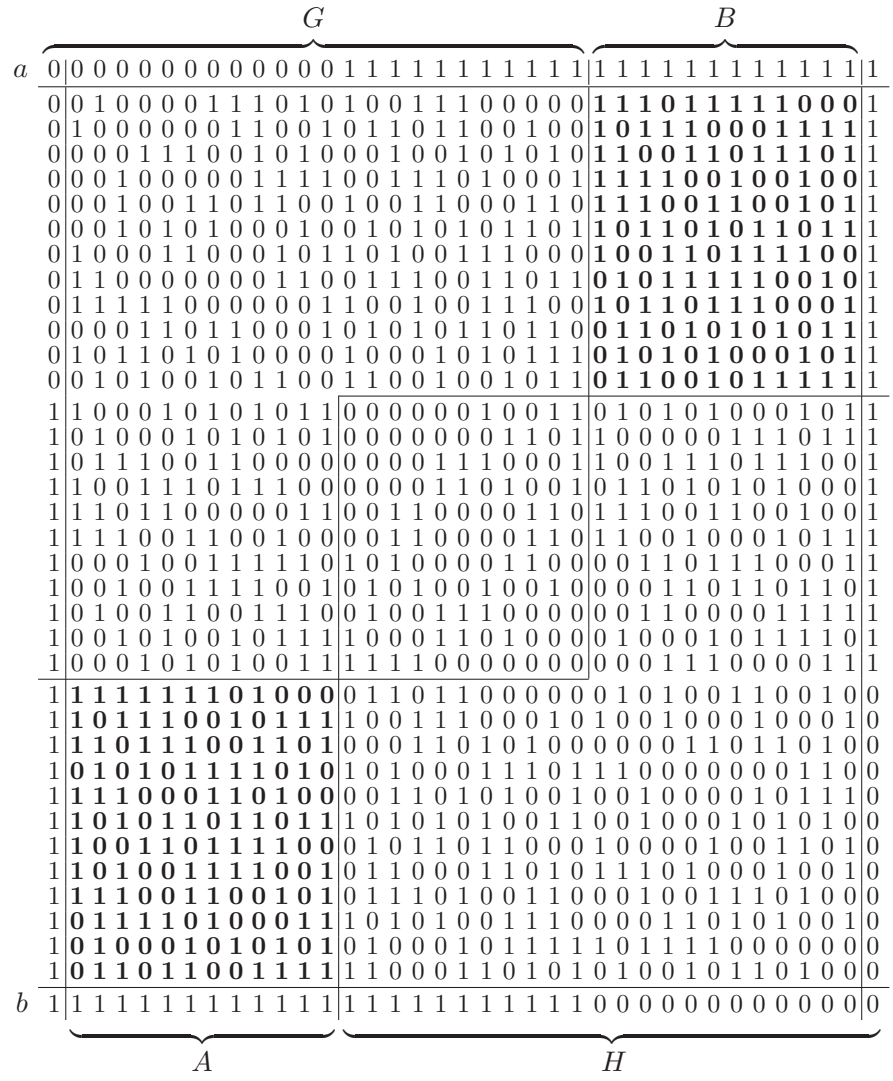


FIGURE 3. An example of $G, H \in \mathcal{R}(4,5,24)$ glued along $K \in \mathcal{R}(3,5,11)$ (the square in the centre), to make $F \in \mathcal{R}(5,5,37)$. The bold entries are those which need to be computed by the gluing procedure.

## 6. Step 3. Empirical results

For $6 \le d \le 9$, no gluings produced any output graphs, so Step 3 was unnecessary. For $d = 10$ we found a total of 647,424 graphs (81,936 nonisomorphic) in $\mathcal{R}(5, 5, 38)$, all of them from a single $K \in \mathcal{R}(3, 5, 10)$. For $d = 11$ we found a total of 15,244 graphs in $\mathcal{R}(5, 5, 37)$, with 15,152 graphs (14,412 nonisomorphic) coming from one $K \in \mathcal{R}(3, 5, 11)$ and 92 graphs (84 nonisomorphic) coming from another $K$. An example is shown in Figure 3. None of these graphs could be extended by one more vertex while staying within $\mathcal{R}(5, 5)$, so Step 3 was completed successfully.

By Lemma 2.2, we do not need gluings for $d \ge 12$, which is fortunate since the number of successful gluings is around 57 billion for $d = 12$ and perhaps even larger for $d = 13$. This would make Step 3 very onerous. Of course, these considerations are the reason we sought to eliminate $d \ge 12$ theoretically.

We wish to acknowledge useful comments from Staszek Radziszowski.

## References

[1] Geoffrey Exoo. A lower bound for $R(5, 5)$. *J. Graph Theory*, 13(1):97–98, 1989.

[2] Brendan D. McKay. Ramsey Graphs. Website at `http://users.cecs.anu.edu.au/~bdm/data/ramsey.html`

[3] Brendan D. McKay and Stanisław P. Radziszowski. $R(4, 5) = 25$. *J. Graph Theory*, 19(3):309–322, 1995.

[4] Brendan D. McKay and Stanisław P. Radziszowski. Subgraph counting identities and Ramsey numbers. *J. Combin. Theory Ser. B*, 69(2):193–209, 1997.