R(4,5) = 25

Brendan D. McKay+
Department of Computer Science
Australian National University
GPO Box 4, ACT 2601, Australia
bdm@cs.anu.edu.au

Stanisław P. Radziszowski*
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623, USA
spr@cs.rit.edu

JGT 19 (1995) 309–322

Abstract.

The Ramsey number $R(4,5)$ is defined to be the least positive integer n such that every n-vertex graph contains either a clique of order 4 or an independent set of order 5. With the help of a long computation using novel techniques, we prove that $R(4,5) = 25$.

1. Introduction.

We shall only consider graphs without multiple edges or loops. For $s, t, n \geq 1$, an (s,t)-graph is a graph without cliques of order s or independent sets of order t, and an (s,t,n)-graph is an (s,t)-graph of order n. Let $\mathcal{R}(s,t)$ and $\mathcal{R}(s,t,n)$ denote the set of all (s,t)-graphs and all (s,t,n)-graphs, respectively.

The Ramsey number $R(s,t)$ is defined to be the least $n > 0$ such that there is no (s,t,n)-graph. The existence of $R(s,t)$ is a corollary of the celebrated paper of Ramsey [11], but some decades passed before serious work began on determining actual values. A good early survey appears in the book [2].

At the time of the paper of Greenwood and Gleason in 1955 [3], only the elementary bound $R(4,5) \leq 31$ was known; this follows from the values $R(3,5) = 14$ and $R(4,4) = 18$ immediately. The first significant lower bound $R(4,5) \geq 25$ was found by Kalbfleisch in 1965 [4], who constructed a $(4,5,24)$-graph with a circular symmetry. Kalbfleisch also stated that he had established the upper bound $R(4,5) \leq 30$. Substantial progress on upper bounds began with Walker [12, 13], who established $R(4,5) \leq 29$ and later $R(4,5) \leq 28$ using some simple linear programs derived from counting subgraphs in an interesting way. This last bound held for 20 years until the present authors reduced it to $R(4,5) \leq 27$ [7]. Firstly, we extended Walker's method with more subgraph identities and exact

+ Supported in part by a grant from the Australia-US Bilateral Cooperative Science and Technology Program

* Supported in part by a grant from the Australian National University Mathematical Sciences Research Visitors Program

data about $(3,5)$-graphs and $(4,4)$-graphs, to make much more powerful linear programs. Then we used those programs to restrict the space of possible $(4,5,27)$-graphs sufficiently that a computer could exhaustively search it. Essentially the same method, using better algorithms, produced the bound $R(4,5) \leq 26$ in 1992 (unpublished).

Unfortunately, as the number of vertices is brought down, the usefulness of the linear program decreases dramatically, to the extent that for the work described in the current paper we did not use it at all. Instead we needed to rely primarily on computational strategies. It was necessary to improve the previous algorithms by more than an order of magnitude before the computation became feasible, even though we were willing to expend more than a decade of computer time.

An extensive summary of current knowledge of graph Ramsey numbers, including generalisations of the classical numbers considered here, can be found in [9].

2. Problem decomposition.

If F is a graph, $v \in VF$ and $W \subseteq VF$, then $N_F(v,W) = \{w \in W \mid vw \in EF\}$. The subgraph of F induced by W will be denoted by $F[W]$. The special case $F[\,VF - \{v\}]$ will also be written as $F - v$.

Suppose that x is a vertex of F. Define the induced subgraphs $G_x = G_x(F) = F[N_F(x, VF)]$ and $H_x = H_x(F) = F[\,VF - N_F(x, VF) - \{x\}]$. If F is a $(4,5,25)$-graph, and $x \in VF$ has degree d, it is clear that G_x is a $(3,5,d)$-graph and H_x is a $(4,4,24-d)$-graph. Since $R(3,5) = 14$ and $R(4,4) = 18$, we must have that $7 \leq d \leq 13$. Complete catalogues of $(3,5)$-graphs and $(4,4)$-graphs have been previously compiled [7, 8, 10]; for the present work they were checked extensively. Counts of these graphs according to their order appear in Table 1. The two (4,4,16)-graphs were previously found by Brass and Mengersen [1]. They also claimed a third, but unfortunately they overlooked the isomorphism of the first two graphs of their Figure 4. (Note that in the second graph that diagonal of length two which determines a vertex of degree 9 is obviously misdrawn.)

In principle we could construct all $(4,5,25)$-graphs directly by taking potential pairs (G_x, H_x) and searching for valid ways to place edges between them, but we avoided this approach for several reasons. Firstly, the number of such pairs is in the hundreds of millions, as can be seen from Table 1. Secondly, we expected this computation would give a null result; i.e., that there are no $(4,5,25)$-graphs. In that event we would have little output that could be subject to consistency checks or could serve to compare two implementations. Instead, we will aim to construct a family of $(4,5,24)$-graphs, defined such that any $(4,5,25)$-graph must be a 1-vertex extension of at least one of the graphs in our family.

For $k = 7,8,9,10$, let $\mathcal{R}'(3,5,k)$ be a set of $(3,5)$-graphs of order less than k such that every $(3,5,k)$-graph contains at least one of them. Similarly, for $k = 11,12$, let $\mathcal{R}'(4,4,k)$ be a set of $(4,4)$-graphs of order less than k such that every $(4,4)$-graph contains at least

n	$\lvert\mathcal{R}(3,5,n)\rvert$	$\lvert\mathcal{R}(4,4,n)\rvert$
1	1	1
2	2	2
3	3	4
4	7	9
5	13	24
6	32	84
7	71	362
8	179	2079
9	290	14701
10	313	103706
11	105	546356
12	12	1449166
13	1	1184231
14		130816
15		640
16		2
17		1

Table 1. Counts of $(3,5)$-graphs and $(4,4)$-graphs

one of them. The actual choice of these sets is important for efficiency, as we will explain later. Suppose that F^* is a $(4,5,25)$-graph. Choose a vertex v of F^* thus:

(i) If x is a vertex of F^* of degree $d \leq 10$, let v be a vertex of $G_x(F^*)$ such that $G_x(F^*) - v$ contains some member of $\mathcal{R}'(3,5,d)$.

(ii) If x has degree $d \geq 12$, let v be a vertex of $H_x(F^*)$ such that $H_x(F^*) - v$ contains some member of $\mathcal{R}'(4,4,24-d)$.

Since F^* cannot be regular of degree 11, having odd order, at least one choice of v is possible. Hence, at least one subgraph of F^* occurs in the set of all $(4,5,24)$-graphs F such that for some x, $G_x(F)$ and $H_x(F)$ are represented by some row of Table 2.

G_x	H_x
$\mathcal{R}'(3,5,7)$	$\mathcal{R}(4,4,17)$
$\mathcal{R}'(3,5,8)$	$\mathcal{R}(4,4,16)$
$\mathcal{R}'(3,5,9)$	$\mathcal{R}(4,4,15)$
$\mathcal{R}'(3,5,10)$	$\mathcal{R}(4,4,14)$
$\mathcal{R}(3,5,12)$	$\mathcal{R}'(4,4,12)$
$\mathcal{R}(3,5,13)$	$\mathcal{R}'(4,4,11)$

Table 2. Required choices for (G_x, H_x)

The process of constructing F from a pair (G_x, H_x) will be called *gluing*. The number of gluings required by Table 2 is far too large for naive gluing methods, as about 50-80 edges between G_x and H_x must be chosen. In the following section, we will describe a gluing algorithm of sufficient efficiency that all the required gluings can be performed within an acceptable time.

3. Gluing.

Of the six rows of Table 2, the most difficult were the fourth and fifth. In this section we will provide a detailed description of the gluing algorithm used for the first four rows, and a brief outline of that used for the other two rows.

For $k = 7, 8, 9$, we took $\mathcal{R}'(3, 5, k) = \mathcal{R}(3, 5, k-1)$. Although smaller sets would have been more efficient, we wished to take the opportunity to compute the complete set of $(4, 5, 24)$-graphs having a vertex of degree 8 or less. For $\mathcal{R}'(3, 5, 10)$ we took a set of 53 $(3, 5, 9)$-graphs chosen, in accord with experiment, to be as sparse as possible.

Suppose that G and H are a $(3, 5)$-graph and a $(4, 4)$-graph, respectively. Define $\mathcal{F}(G, H)$ to be the set of all $(4, 5)$-graphs F such that for some vertex $x \in VF$, $G_x(F) = G$ and $H_x(F) = H$. We will use F to refer to a representative member of $\mathcal{F}(G, H)$. For definiteness, we will suppose that the vertices of G are labelled with contiguous integers $0, 1, 2, \ldots$ and that induced subgraphs of G are also labelled contiguously $0, 1, 2, \ldots$ in the order induced from the labelling of G.

Define a *feasible cone* to be a subset of VH which covers no clique of order 3. If H is a $(4, 4, 14)$-graph, there are typically about 4000 feasible cones. The relevance of feasible cones is that $N_F(v, VH)$ must be a feasible cone for every vertex $v \in VG$. Our problem is to choose feasible cones $C_0, C_1, \ldots,$ one for each vertex of G, such that no cliques of order 4 or independent sets of order 5 appear in F. The various positions in which these forbidden subgraphs might occur are as follows.

K_2: Two adjacent vertices $v, w \in VG$ have $C_v \cap C_w$ covering some edge of H.

E_t: For some independent set $w_0, \ldots, w_{t-1}$ of G, there is an independent set of order $5-t$ in H which is completely missed by $C_{w_0} \cup C_{w_1} \cup \cdots C_{w_{t-1}}$ $(t = 2, 3, 4)$.

The gluing operation can be achieved by a direct backtrack search of depth $|VG|$, but this is insufficiently efficient due to the very large number of feasible cones. Instead, we will partition the set of feasible cones into well-structured families which can be processed in parallel. An *interval of feasible cones*, for brevity called an *interval*, is a set of feasible cones of the form $\{X \mid B \subseteq X \subseteq T\}$ for some feasible cones $B \subseteq T$. This interval will be denoted by $[B, T]$, and we will call B and T its *bottom* and *top*, respectively. Obviously, $[B, T]$ contains $2^{|T|-|B|}$ feasible cones. Using a simple heuristic search, the typical set of 4000 feasible cones when H is a $(4, 4, 14)$-graph can be written as the disjoint union of about 100 intervals. The dimensions $|T| - |B|$ range from 0 to 8.

Suppose $m = |VG|$. If $C_0, \ldots, C_{m-1}$ are feasible cones, then $F(G, H; C_0, \ldots, C_{m-1})$ denotes the graph F with vertex x such that $G_x(F) = G$, $H_x(F) = H$, and $C_i = N_F(i, VH)$ for $0 \le i \le m-1$. Clearly, this is a $(4,5,24)$-graph if and only if conditions K_2, E_2, E_3, E_4 are avoided. Similarly, if $I_0, I_1, \ldots, I_{m-1}$ are intervals, then $\mathcal{F}(G, H; I_0, \ldots, I_{m-1})$ represents the set of all $(4,5,24)$-graphs $F(G, H; C_0, \ldots, C_{m-1})$ such that $C_i \in I_i$ for $0 \le i \le m-1$.

Given H, we define three functions $H_1, H_2, H_3 : 2^{VH} \to 2^{VH}$. Namely, for $X \subseteq VH$ let

$$H_1(X) = \{w \in VH \mid vw \in EH \text{ for some } v \in X\};$$

$$H_2(X) = \{w \in VH \mid vw \notin EH \text{ for some } v \notin X\};$$

$$H_3(X) = \{w \in VH \mid \{u, v, w\} \text{ is an independent 3-set of } H \text{ for some } u, v \notin X\}.$$

These functions can be computed by means of simple recursions. Using them, we can define some *collapsing rules* that apply to sequences $I_0, \ldots, I_{m-1}$ of intervals. The rules depend on the graphs G and H. In each case, either some interval is replaced by an interval contained in it, or the special event FAIL occurs. Suppose $I_i = [B_i, T_i]$ for each i, and define collapsing rules (a)–(d) as follows:

(a) Suppose $\{u, v\} \in EG$.
  if $B_u \cap B_v \cap H_1(B_u \cap B_v) \neq \emptyset$ **then** FAIL
  else $T_u := T_u - \big(H_1(B_u \cap B_v) \cap B_v\big)$

(b) Suppose $\{u, v\} \notin EG$, where u, v are distinct vertices of G.
  if $H_3(T_u \cup T_v) \not\subseteq T_u \cup T_v$ **then** FAIL
  else $B_u := B_u \cup \big(H_3(T_u \cup T_v) - T_v\big)$

(c) Suppose $\{u, v, w\}$ is an independent 3-set of G.
  if $H_2(T_u \cup T_v \cup T_w) \not\subseteq T_u \cup T_v \cup T_w$ **then** FAIL
  else $B_u := B_u \cup \big(H_2(T_u \cup T_v \cup T_w) - (T_v \cup T_w)\big)$

(d) Suppose $\{u, v, w, z\}$ is an independent 4-set of G.
  if $T_u \cup T_v \cup T_w \cup T_z \neq VH$ **then** FAIL
  else $B_u := B_u \cup \big(VH - (T_v \cup T_w \cup T_z)\big)$

In all of the above it should be noted that there is a different collapsing rule for each vertex in the given edge or independent set. For example, in rule (c) the stated role for u could equally be played by v or w. The reason these collapsing rules are useful is the following lemma.

Lemma 1. *Suppose that some collapsing rule is applied to $I_0, \ldots, I_{m-1}$.*
If FAIL *occurs, then $\mathcal{F}(G, H; I_0, \ldots, I_{m-1}) = \emptyset$.*

Otherwise, $\mathcal{F}(G,H;I_0,\ldots,I_{m-1}) = \mathcal{F}(G,H;I'_0,\ldots,I'_{m-1})$,
where $I'_0,\ldots,I'_{m-1}$ *is the sequence of intervals after the rule application.*

Proof. Consider Rule (a), for example. Let $\{y,z\} \in EH$ be an edge such that $y \in B_u \cap B_v$ and $z \in T_u \cap B_v$. Clearly u cannot be adjacent to z, since that would imply a 4-clique $\{u,v,w,z\}$ (condition K_2). Thus, $\mathcal{F}(G,H;I_0,\ldots,I_{m-1}) = \emptyset$ if $z \in B_u$; otherwise z can be removed from T_u. Application of this idea simultaneously to all such edges $\{y,z\}$ is precisely Rule (a).

Rules (b)–(d) apply similar ideas to avoid independent 5-sets (conditions E_2–E_4, respectively). ∎

If the collapsing rules are applied repeatedly, we must eventually encounter either a FAIL condition or a stable situation where no collapsing rule can FAIL or reduce an interval strictly. It turns out that the final state is independent of the order of application of the collapsing rules. This is a special case of the following elementary result.

Let $(X,\leq)$ be a partially ordered set, and let Φ be a family of functions from X to X. Suppose that, for $x,x' \in X$ and $\phi \in \Phi$ we have $\phi(x) \leq x$ and $x \leq x' \Rightarrow \phi(x) \leq \phi(x')$. Call $x \in X$ *Φ-stable* if $\phi(x) = x$ for all $\phi \in \Phi$. Let $\Phi^*(x)$ denote the closure of $\{x\}$ under Φ.

Lemma 2. *For each* $x \in X$, $\Phi^*(x)$ *contains at most one* Φ*-stable element.*

Proof. Suppose that for $\phi_1,\ldots,\phi_r,\phi'_1,\ldots,\phi'_s \in \Phi$, both $y = \phi_r(\cdots\phi_1(x)\cdots)$ and $y' = \phi'_s(\cdots\phi'_1(x)\cdots)$ are Φ-stable. Then $y = \phi'_s(\cdots(\phi'_1(\phi_r(\cdots\phi_1(x)\cdots)\cdots)$ since y is Φ-stable, and so $y \leq y'$ by the stated conditions on Φ. Similarly $y' \leq y$, and so $y' = y$. ∎

To apply Lemma 2 for given G and H, let X be the set of all m-tuples $(I_0,\ldots,I_{m-1})$ of intervals, together with the special value FAIL. Define $x \leq x'$ if either $x =$ FAIL or $x = (I_0,\ldots,I_{m-1})$, $x' = (I'_0,\ldots,I'_{m-1})$ and $I_i \subseteq I'_i$ for $0 \leq i < m$. Let Φ be the set of all available collapsing rules, extended to map FAIL onto FAIL always. The requirements for Lemma 2 are now easily checked, noting that the functions H_1, H_2 and H_3 are monotonically nondecreasing, nonincreasing and nonincreasing, respectively, and that the finiteness of $\Phi^*(x)$ guarantees that it contains at least one Φ-stable element.

The result of applying collapsing rules until either a FAIL condition or stability occurs will be called *collapsing*; it replaces $(I_0,\ldots,I_{m-1})$ by $C(G,H;I_0,\ldots,I_{m-1})$, where the latter is either FAIL or a sequence $(I'_0,\ldots,I'_{m-1})$ such that $I'_i \subseteq I_i$ for $0 \leq i \leq m-1$. The latter stable sequence is said to be *fully collapsed* (for G and H).

The fundamental theorem about collapsing is as follows.

Theorem 1. *If* $C(G,H;I_0,\ldots,I_{m-1}) =$ FAIL *then* $\mathcal{F}(G,H,I_0,\ldots,I_{m-1}) = \emptyset$.
Otherwise, define $(I'_0,\ldots,I'_{m-1}) = C(G,H;I_0,\ldots,I_{m-1})$. *Then* $\mathcal{F}(G,H,I'_0,\ldots,I'_{m-1}) =$

$\mathcal{F}(G, H, I_0, \ldots, I_{m-1})$ *and if, in addition,* $|I'_0| = |I'_1| = \cdots = |I'_{m-1}| = 1$, *then* $\mathcal{F}(G, H; I'_0, \ldots, I'_{m-1})$ *consists of a single* $(4, 5)$*-graph.*

Proof. All but the final claim follows by repeated application of Lemma 1. For last part, note that the existence of any 4-clique or independent 5-set would lead to the corresponding collapsing rule causing condition FAIL. ∎

Note that every time a collapsing rule modifies an interval, the number of feasible cones it represents is divided by a power of two. This ability of collapsing to purge many infeasible configurations at the same time is the primary reason for the success of this approach.

We can now see a search procedure for gluing using intervals. Suppose inductively that we have all fully collapsed sequences of intervals $(I'_0, \ldots, I'_{r-1})$ for $G[\{0, 1, \ldots, r-1\}]$. Those for $G[\{0, 1, \ldots, r\}]$ have the form $C(G[\{0, 1, \ldots, r\}], H; I'_0, \ldots, I'_{r-1}, I_r)$, where I_r is some interval and choices causing FAIL are rejected.

Given all fully collapsed sequences $(I'_0, \ldots, I'_{m-1})$ for G, we can easily find $\mathcal{F}(G, H)$. Sequences with $|I'_0| = \cdots = |I'_{m-1}| = 1$ yield a single solution, as shown in Theorem 1. Those which have some $I'_i = [B'_i, T'_i]$ with $B'_i \neq T'_i$ can be recursively split into the disjoint configurations $C(G, H; I'_0, \ldots, [B'_i \cup \{w\}, T'_i], \ldots, I'_{m-1})$ and $C(G, H; I'_0, \ldots, [B'_i, T'_i - \{w\}], \ldots, I'_{m-1})$ for some $w \in T'_i - B'_i$, with values causing FAIL being rejected as usual.

This algorithm is already more than one order of magnitude faster than simple backtracking with feasible cones, but further substantial improvement is possible. One source of unnecessary inefficiency is that there are typically 100 intervals that might be chosen for I_r, and most of them lead to FAIL conditions for moderate r. We can reduce the required number of collapsing operations markedly with the help of a simple look-ahead. This is achieved with the help of a data-structure which has a further great advantage: many different G's can be processed simultaneously.

Suppose $1 \leq a_2 \leq a_3 \leq \cdots$ are integers such that $a_i < i$ for $i \geq 2$. We will define two relations on the set of labelled graphs, where the labels are the integers $\{0, 1, \ldots, m-1\}$ if the order is m. Suppose that J is such a graph with $m \geq 2$ vertices. Then

$$\mathrm{parent}(J) = J[\{0, 1, \ldots, m-2\}]$$

and

$$\mathrm{adjunct}(J) = J[\{0, 1, \ldots, a_m - 2, m-1\}],$$

where the final vertex of adjunct(J) is given the label $a_m - 1$ in accordance with our convention for labelling subgraphs. It follows from the definitions that parent(J) and adjunct(J) have $m-1$ and a_m vertices, respectively.

From Lemma 2 we easily have the following.

Lemma 3. *Let $I_0, \ldots, I_{m-1}$ be intervals. If $C(\text{parent}(J), H; I_0, \ldots, I_{m-2})$ is FAIL or $C(\text{adjunct}(J), H; I_0, \ldots, I_{a_m-2}, I_{m-1})$ is FAIL, then $C(J, H; I_0, \ldots, I_{m-1})$ is FAIL. Otherwise, suppose $(I'_0, \ldots, I'_{m-2}) = C(\text{parent}(J), H; I_0, \ldots, I_{m-2})$ and $(I''_0, \ldots, I''_{a_m-1}) = C(\text{adjunct}(J), H; I_0, \ldots, I_{a_m-2}, I_{m-1})$. Then,*

$$C(J, H; I_0, \ldots, I_{m-1}) = C(J, H; I'_0 \cap I''_0, \ldots, I'_{a_m-2} \cap I''_{a_m-2}, I'_{a_m-1}, \ldots, I'_{m-2}, I''_{a_m-1}),$$

where the value is taken as FAIL if any of the intersections are empty. ∎

The primary usage of Lemma 3 is to reduce the number of collapsing operations needed to deduce the fully collapsed configurations for J from those for parent(J). Instead of perhaps 100 possibilities for I_{m-1} we typically have only a few, the others having caused FAIL when tried for adjunct(J). Furthermore, the collapsing operations required for adjunct(J) are independent of I_{m-2} and so need not be repeated if only I_{m-2} changes.

To implement these ideas efficiently we construct an object consisting of a pair of superimposed rooted trees. We will call it a *double tree*. For definiteness, suppose we wish to glue a single H to a family $\mathcal{R}' = \mathcal{R}'(3, 5, k)$. The nodes of the double tree are labelled graphs, namely the members of $\mathcal{R}'$ and recursively the parent and adjunct of every node. The graph K_1 has no parent or adjunct; call it the *root*. The edges $(J, \text{parent}(J))$ form a rooted tree called the *parent tree*, of which those nodes and edges on a path from a member of $\mathcal{R}'$ to the root constitute the *main branches* (a subtree of the parent tree). The edges $(J, \text{adjunct}(J))$ form another tree, called the *adjunct tree*. Since $\text{parent}(\text{adjunct}(J)) = \text{parent}^{\,m-a_m+1}(J)$, where $m = |\,VJ|$ and $a_m \geq 2$, the total number of nodes in the double tree is at most $k-1$ times the number in the main branches.

In order to reduce the total number of nodes in the double tree, especially close to the root, the members of $\mathcal{R}'$ were labelled so that their leading subgraphs coincided as much as possible. Experiments showed that placing sparse subgraphs first was the most efficient on average for our purposes.

Now the primary gluing algorithm can be described. Beginning at the root, the main branches of the double tree are scanned in depth-first order. For the root, all intervals are collapsed. For other nodes, the collapsed configurations for the parent and the adjunct are combined as described in Lemma 3. A time-stamping system is used to detect when the configurations for the adjunct are known already or need to be computed. Computation at the adjunct might recursively require computations at the adjunct of the adjunct, and so on. Typical experience was that whenever the adjunct configurations were required they were already valid about 90% of the time.

Further speed-up, of a factor of 2–3, was achieved by employing the symmetries of the low-order nodes in the double-tree. We found it adequate to employ all the symmetries of the nodes of order 4 or less.

For the final two rows of Table 2, an alternate but similar algorithm was used. Instead of searching in a space of structure derived from a set of possibilities for G, using intervals in H, the roles of G and H were reversed. Intervals were defined in the power set of VG, and the search space structure was derived from the set of possibilities for H. Since H is a $(4,4)$-graph, collapsing rule (d) was not needed, but a new collapsing rule for triangles in H was used. The much larger number of $(4,4)$-graphs compared to $(3,5)$-graphs led us to choose $\mathcal{R}'(4,4,k)$ to consist of graphs smaller than $k-1$ vertices. For $k=12$ we used 23 $(4,4,7)$-graphs and 51 $(4,4,8)$-graphs. For $k=11$ we used 28 $(4,4,8)$-graphs and 113 $(4,4,9)$-graphs. Experience led us to favour dense graphs, and label them with their densest subgraphs first. In each case, the gluing operation as previously described takes us to graphs on 21 or 22 vertices. These were extended in all possible ways to 24 vertices using a method that applies collapsing rules to determine the edges incident with the remaining vertices of H as well as those between G and H.

4. One-vertex extensions.

Our final requirement is an algorithm for extending $(4,5)$-graphs by a single vertex. Suppose F is a $(4,5,n)$-graph. We wish to find all manners in which a new vertex v can be joined to F to make a $(4,5,n+1)$-graph. Clearly it is necessary and sufficient that $N(v, VF)$ does not cover any triangle of F and hits every independent 4-set of F.

Let $X_1, X_2, \ldots, X_r$ be a list containing all the triangles and independent 4-sets of F, in some order. Similarly to the gluing algorithm, we will consider intervals $[B,T]$ of subsets of VF. The extension algorithm uses a set $\mathcal{I}$ of such intervals.

$\mathcal{I} := \{[\emptyset, VF]\}$
for $i := 1$ **to** r **do**
    if X_i is a triangle **then**
        for each $[B,T] \in \mathcal{I}$ such that $X_i \subseteq T$ **do**
            if $X_i \subseteq B$ **then**
                Delete $[B,T]$ from $\mathcal{I}$.
            else
                Replace $[B,T]$ by $[B \cup \{y_1, \ldots, y_{j-1}\}, T - \{y_j\}]$ for $j = 1, \ldots, k$,
                        where $X_i - B = \{y_1, \ldots, y_k\}$.
            endif
        endfor
    else [if X_i is an independent 4-set]
        for each $[B,T] \in \mathcal{I}$ such that $X_i \cap B = \emptyset$ **do**
            if $X_i \cap T = \emptyset$ **then**
                Delete $[B,T]$ from $\mathcal{I}$.
            else
                Replace $[B,T]$ by $[B \cup \{y_j\}, T - \{y_1, \ldots, y_{j-1}\}]$ for $j = 1, \ldots, k$,
                        where $X_i \cap T = \{y_1, \ldots, y_k\}$.
            endif
        endfor
    endif
endfor

At completion of the algorithm, $\mathcal{I}$ will contain a set of disjoint intervals whose union is the set of possible neighbourhoods $N(v, VF)$. We will leave the proof of this claim to the reader.

The efficiency of the algorithm depends considerably on the order of the elements in the list $X_1, \ldots, X_r$. A reasonably good method is to sort on greatest element, then second greatest, and so on, with the triangles and independent 4-sets being sorted in together. With careful implementation this algorithm can achieve typical extensions from 24 to 25 vertices in about 10 milliseconds, which is more than one order of magnitude faster than straightforward searching without using intervals.

5. Computations and verification.

Two separate implementations of the algorithms described in Sections 3 and 4 were constructed, one by each author. The general structure of the two implementations was similar, but the detail was quite different. For example, different partitions of the set of feasible cones into intervals, and different adjunct orders a_i were used. The generated $(4, 5, 24)$-graphs were compared for each individual gluing operation, or sometimes in small groups, with no discrepancy being found. Some representative gluings were also performed using independent unsophisticated searches, again with identical results. Isomorphism testing was performed using the program *nauty* distributed by the first author [5].

The two implementations required 3.2 years and 6 years of cpu time on Sun Microsystems computers (mostly Sparcstation SLC). This was achieved without undue delay by employing a large number of computers (up to 110 at once).

As a result of these computations, about 250000 $(4, 5, 24)$-graphs were found. These were shown to not be induced subgraphs of $(4, 5, 25)$-graphs using two independent programs for extending $(4, 5, n)$-graphs to $(4, 5, n+1)$-graphs. Thus we have our main theorem.

Theorem 2. $R(4, 5) = 25$. ∎

Due to the great length of the computation, and the complexity of the algorithms, we sought further evidence of the correctness of the output. Firstly, we did a large number of additional gluing operations, including enough to find all $(4, 5, 24)$-graphs with maximum degree at least 12 and those regular of degree 11. The latter computation required additional techniques and will be described in another paper. These results imply the bound $R(5, 5) \le 49$. We also found many additional $(4, 5, 24)$-graphs by random search. We then closed our catalogue of $(4, 5, 24)$-graphs under small perturbations, as follows.

For integer k, define the relation $\sim_k$ on the set of $(4, 5, 24)$-graphs by $F_1 \sim_k F_2$ if F_1 and F_2 have a common k-vertex subgraph (up to isomorphism). Using one of the extension programs mentioned above, we found the closure of our catalogue of $(4, 5, 24)$-graphs under $\sim_{22}$. This was achieved by taking each $(4, 5, 24)$-graph and for each pair of vertices removing those vertices then extending the resulting subgraph back to 24 vertices using the algorithm of Section 4. This process was repeated until it stabilised. Some large subfamilies were closed under the weaker relation $\sim_{21}$.

Eventually, a collection of 350904 $(4, 5, 24)$-graphs had been formed, none of which extend to a $(4, 5, 25)$-graph. Some statistics about this collection will be given in the next section. We then searched the collection for all those graphs which should have been generated by each of the original gluing operations and compared the results to the solutions found by the gluing programs. The comparison succeeded in every case; i.e., our extra work (perhaps another two years of cpu time) failed to demonstrate incompleteness in the gluing programs. We believe that this is strong evidence for the correctness of those programs.

6. The structure of $\mathcal{R}(4, 5)$.

In the course of the project, and through some auxiliary computations, we collected a significant amount of information on $\mathcal{R}(4, 5)$. Since it is of importance to our further investigations, for example into bounds on higher Ramsey numbers, we will record some of it here.

Firstly, we consider the number of $(4, 5, n)$-graphs for each n. The one-vertex extension method of Section 4 can be fitted easily into the general scheme of [6] to make a program generating $(4, 5)$-graphs without isomorphs. For our purposes it is sufficient to note that the method of [6] represents the isomorphism classes as the nodes of a rooted tree with root K_1. The parent of each $(4, 5, n)$-graph is a $(4, 5, n - 1)$-subgraph $(n \geq 2)$.

Using this method we generated all $(4, 5, n)$-graphs for $n \leq 11$. It would be feasible for $n = 12$ but we did not do it. For $n \geq 12$ we used a simple procedure to estimate the number of graphs without exhaustive generation. Let $p_2, p_3, \ldots, p_{24}$ be probabilities. Each time an isomorphism class of order n is generated, reject it with probability $1 - p_n$, each such rejection being independent of the others. Then the total number of $(4, 5, n)$-graphs, multiplied by $p_2 p_3 \cdots p_n$, is the expectation of the number of nodes generated and accepted at order n. A large number of such experiments were performed. In Table 3, the values are exact for $n \leq 11$ and are estimates based on our random sampling for $12 \leq n \leq 22$. Few graphs of order $n > 22$ were generated, due to the small number of them relative to the large hump at order 19. The value for $n = 23$ is a barely more than a guess. We expect that the correct value for $n = 24$ is at most a few hundred beyond the number given.

Also in Table 3 we record our current best bounds on $e(4, 5, n)$ and $E(4, 5, n)$, the minimum and maximum number of edges in $(4, 5, n)$-graphs, respectively. This updates Table II in [7]. The exact values were found by direct computation. The upper bounds on $e(4, 5, n)$ and the lower bounds on $E(4, 5, n)$ were proved by constructing examples. Otherwise, the bounds are derived from linear programming as explained in [7], with the exception that the bound $E(4, 5, 18) \leq 88$ is a trivial consequence of $E(4, 5, 17) = 79$.

In Table 4, we give some further information on the known $(4, 5, 24)$-graphs. The table contains the numbers of known graphs, and known ranges for some of their parameters.

n	$\|\mathcal{R}(4,5,n)\|$	$e(4,5,n)$	$E(4,5,n)$
1	1	0	0
2	2	0	1
3	4	0	3
4	10	0	5
5	28	1	8
6	114	2	12
7	627	3	16
8	5588	4	21
9	81321	6	27
10	1915582	8	33
11	67445833	10	40
12	3.22×10^9	12	48
13	1.85×10^{11}	17	53
14	1.11×10^{13}	22	60
15	5.96×10^{14}	27	66
16	2.32×10^{16}	32	72
17	5.17×10^{17}	41	79
18	4.97×10^{18}	48-50	85-88
19	1.46×10^{19}	56-57	92-97
20	8.5×10^{18}	66-68	100-105
21	5.5×10^{17}	75-77	107-114
22	1.9×10^{15}	86-88	114-122
23	10^{11}	98-102	121-130
24	≥ 350904	109-116	132
all	2.91×10^{19}		

Table 3. Statistics for $(4,5)$-graphs

There might exist graphs lying outside those ranges, but it is certain that the catalogue is complete if the minimum degree is 6, 7, 8 (1979, 7491, 51803 graphs respectively), or the maximum degree is 12 or 13 (74375, 961 graphs, respectively), or the graph is regular of degree 11 (2 graphs). For each number of edges (e), the table gives ranges for the number of independent 3-sets and 4-sets (i_3 and i_4), the number of triangles (c_3), and the minimum and maximum degree (δ and Δ). Of those 350904 graphs, 10872 have nontrivial automorphism groups, and five have vertex-transitive automorphism groups.

References.

[1] P. Brass and I. Mengersen, Construction of special edge colorings, *Utilitas Mathematica*, **36** (1989) 201–210.

e	i_3	i_4	c_3	δ	Δ	number
116	356–368	232–262	123–128	8–9	10–11	7
117	346–362	216–253	122–132	8–9	10–11	86
118	340–360	206–251	120–136	6–9	10–12	785
119	332–356	198–247	124–140	6–9	10–13	4293
120	324–352	186–243	127–144	6–10	10–13	16165
121	319–344	181–232	130–146	6–10	11–13	43110
122	314–337	178–223	133–149	6–10	11–13	79257
123	310–330	171–215	136–152	6–10	11–13	92217
124	304–324	163–208	140–154	6–10	11–13	67101
125	302–318	161–201	144–157	6–10	11–13	31974
126	296–312	155–195	147–160	7–10	11–12	11481
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	350904

Table 4. Further statistics for known $(4, 5, 24)$-graphs

[2] R. L. Graham, B. L. Rothschild and J. H. Spencer, Ramsey Theory, John Wiley and Sons, 1990.

[3] R. E. Greenwood and A. M. Gleason, Combinatorial relations and chromatic graphs, *Canadian J. Math.*, **7** (1955) 1–7.

[4] J. G. Kalbfleisch, Construction of special edge-chromatic graphs, *Canadian Math. Bulletin*, **8** (1965) 575–584.

[5] B. D. McKay, nauty users' guide (version 1.5), Technical report TR-CS-90-02, Computer Science Dept., Australian National University, 1990.

[6] B. D. McKay, Isomorph-free exhaustive generation, in preparation.

[7] B. D. McKay and S. P. Radziszowski, Linear programming in some Ramsey problems, *J. Combinatorial Theory, Ser. B*, to appear.

[8] B. D. McKay and K. M. Zhang, The value of the Ramsey number $R(3, 8)$, *J. Graph Theory*, **16** (1992) 99–105.

[9] S. P. Radziszowski, Small Ramsey numbers, Technical Report RIT-TR-93-009, Dept. of Computer Science, Rochester Institute of Technology, 1993.

[10] S. P. Radziszowski and D. L. Kreher, On $R(3, k)$ Ramsey graphs: theoretical and computational results, *J. Combinatorial Math. and Combinatorial Computing*, **4** (1988) 37–52.

[11] F. P. Ramsey, On a problem of formal logic, *Proc. London Math. Soc.*, **30** (1930) 264–286.

[12] K. Walker, Dichromatic graphs and Ramsey numbers, *J. Combinatorial Theory*, **5** (1968) 238–243.

[13] K. Walker, An upper bound for the Ramsey number $M(5,4)$, *J. Combinatorial Theory*, **11** (1971) 1–10.