Proper edge-colourings of complete graphs

M.J. Grannell, T.S. Griggs, K.A.S. Quinn
Department of Pure Mathematics
The Open University
Walton Hall
Milton Keynes MK7 6AA
UNITED KINGDOM

W.L. Kocay, R.G. Stanton
Department of Computer Science
University of Manitoba
Winnipeg
CANADA R3T 2N2

Dedicated to Ernie Cockayne on his 60^{th} birthday

This is a preprint of an article published in the Journal of Combinatorial Mathematics and Combinatorial Computing, 33, 2000, p103-116

Abstract

This paper considers the following question: how many non-isomorphic proper edge-colourings (with any number of colours) are there of the complete graph K_n? We prove an asymptotic result and enumerate the solutions for $n \leq 6$.

1 Introduction

Amongst the vast literature which exists on edge-colourings of graphs, the following easily stated problem seems never to have been considered: given the complete graph on $n \geq 2$ vertices, K_n, determine the number of non-isomorphic proper edge-colourings. The extreme cases in which the number of colours is a minimum, $n-1$ if n is even and n if n is odd, are respectively one-factorizations and near one-factorizations, and these problems have attracted considerable attention. For the former, [5] and [8] are excellent surveys. See also the entry in the Handbook [1]. Asymptotic results are given in [2]. But there is silence on the situation in which the number of colours can take any value between the minimum value m and the maximum value, $M = n(n-1)/2$, the total number of edges.

The reason for this may be twofold. First, the well-known phenomenon of combinatorial explosion which occurs in many problems of this nature is particularly severe in this case. For $m \leq i \leq M$, denote the number of non-isomorphic proper edge-colourings of K_n using i colours by $c(i,n)$. Further, let $C(n) = \sum_{i=m}^{M} c(i,n)$. Then trivially $C(2) = c(1,2) = 1$ and $C(3) = c(3,3) = 1$. Almost as easy are the results $c(i,4) = 1$, $3 \leq i \leq 6$, giving $C(4) = 4$; and a few moments' work establishes that $c(5,5) = 1$, $c(6,5) = 2$, $c(7,5) = 4$, $c(8,5) = 2$, $c(9,5) = 1$, and $c(10,5) = 1$, giving $C(5) = 11$. The colourings are listed in [3]. However, the same problem for $n = 6$ is already a significant challenge.

The second reason may be that there was no previously known application of the results which merited their consideration. But they are related to a well-known problem in design theory. The quantity $g^{(k)}(v)$ was introduced in [7] as the minimum number of blocks necessary in a pairwise balanced design on v elements, subject to the condition that the longest block has length k. In [9], Woodall showed that, in effect, $g^{(k)}(v) \geq 1+(v-k)(3k-v+1)/2$. It was later shown (Theorem 3.3 of [7]) that if $v \equiv 1 \pmod 4$ then this bound is achieved for $k > (v-1)/2$ and otherwise for $k \geq (v-1)/2$. Moreover (Theorem 3.4 of [7]), the only designs which achieve these bounds are those which use pairs and triples as blocks other than the 'long' block of length k. For values of k and v satisfying the relevant inequality above, the number, $N^{(k)}(v)$, of pairwise non-isomorphic designs achieving the lower bound is equal to the total number of non-isomorphic proper edge-colourings of the complete graph K_{v-k} using any number of colours from the minimum m to k' where $k' = \min(k, (v-k)(v-k-1)/2)$, i.e. $N^{(k)}(v) = \sum_{i=m}^{k'} c(i, v-k)$. The design is constructed from the proper edge-colouring as follows. Assign a different element of the 'long' block to each colour class of the proper edge-colouring. Now, for each element, say x, so assigned, form triples of x with the vertices incident to each edge of that colour class. Complete the design with the missing pairs. See [3] for details.

We are particularly interested in the values of $N^{(v-6)}(v)$, $v \geq 11$, and hence in all non-isomorphic proper edge-colourings of the complete graph K_6. The determination of these is the main result of this paper. However, first we present an asymptotic result.

2 Asymptotic result

Theorem 2.1 *The number $C(n)$ of non-isomorphic proper edge-colourings of K_n satisfies*

$$\lim_{n\to\infty}\left(\frac{\log_e C(n)}{n^2\log_e n}\right)=1,$$

that is

$$C(n)=n^{n^2(1+o(1))}\ \text{as } n\to\infty.$$

Proof First we obtain an upper bound for $C(n)$. A proper edge-colouring of K_n with up to k colours may be represented by a symmetric $n\times n$ array with diagonal cells blank and all other cells occupied by numbers from the set $\{1,2,\ldots,k\}$ such that no single row or single column contains a repeated entry. By considering entries in the upper triangle it is clear that the number of distinct proper edge-colourings of K_n with up to k colours is bounded above by $k^{n(n-1)/2}$. Hence

$$C(n)\le(n(n-1)/2)^{n(n-1)/2}\le(n^2)^{n^2/2}=n^{n^2}.$$

So

$$\frac{\log_e C(n)}{n^2\log_e n}\le 1.$$

We now obtain a lower bound for $C(n)$. We again take the representation of a proper edge-colouring by a symmetric $n\times n$ array as above, and complete the rows of the upper triangle one at a time. Denote the symmetric $n\times n$ array by A and, for each cell $A[i,j]$, let $b[i,j]$ be the minimum number of choices for a number from the set $\{1,2,\ldots,k\}$, where we assume that $k\ge 2n-3$. Then, working row by row and from left to right,

$$\begin{aligned}b[1,j]&=k-(j-2),\quad 2\le j\le n,\\ b[2,j]&\ge k-(j-1),\quad 3\le j\le n,\end{aligned}$$

and in general,

$$b[i,j]\ge k-(i+j-3),\quad 1\le i<j\le n.$$

Hence $b[i,j]\ge k-2n+4$ for $1\le i<j\le n$. It follows that the number of non-isomorphic proper edge-colourings of K_n with up to k colours is at least

$$\frac{(k-2n+4)^{n(n-1)/2}}{n!\,k!}$$

for any $k\ge 2n-3$, where the denominator deals with possible isomorphisms (all possible permutations of the n vertices and k colours). Hence

$$C(n)\ge\frac{(k-2n+4)^{n(n-1)/2}}{n!\,k!}$$

for all $k \geq 2n-3$. Now assume that $n \geq 2$ and put $k = k^* = \lceil n^2/\log_e n \rceil$. Then $k^* \geq 2n-3$ and so

$$C(n) \geq \frac{\left(\dfrac{n^2}{\log_e n} - 2n + 4\right)^{n(n-1)/2}}{n!\,\Gamma(k^*+1)}$$

which gives

$$C(n) \geq \frac{\left(\dfrac{n^2}{\log_e n} - 2n + 4\right)^{n(n-1)/2}}{n!\,\Gamma\left(\dfrac{n^2}{\log_e n} + 2\right)}.$$

Thus

$$\begin{aligned}
&\log_e C(n) \geq \\
&\frac{n(n-1)}{2}\left(\log_e n^2 + \log_e\left(1 - \frac{2\log_e n}{n} + \frac{4\log_e n}{n^2}\right) - \log_e \log_e n\right) \\
&- \left(n + \frac{1}{2}\right)\log_e n + n + O(1) \\
&- \left(\frac{n^2}{\log_e n} + \frac{3}{2}\right)\left(\log_e n^2 + \log\left(1 + \frac{2\log_e n}{n^2}\right) - \log_e \log_e n\right) \\
&+ \frac{n^2}{\log_e n} + 2 + O(1),
\end{aligned}$$

using Stirling's formula $\Gamma(x) = x^{x-1/2}e^{-x}\sqrt{2\pi}(1+o(1))$, and the fact that $n! = \Gamma(n+1)$. It follows that

$$\frac{\log_e C(n)}{n^2 \log_e n} \geq 1 + o(1).$$

Combining the upper and lower bounds gives the result.

It is interesting to compare Theorem 2.1 with the result of Theorem 4.2 and Remark #1 on page 66 of [2], which can be stated as follows.

Theorem 2.2 [2] *For n even, let $F(n)$ be the number of pairwise non-isomorphic one-factorizations of the complete graph K_n. Then*

$$\lim_{n\to\infty}\left(\frac{\log_e F(n)}{n^2 \log_e n}\right) = \frac{1}{2},$$

that is

$$F(n) = n^{n^2(1/2+o(1))} \text{ as } n \to \infty.$$

3 Enumeration results

The complete graph K_6 has 15 edges, and each colour class in a proper edge-colouring contains at most 3 edges. Our method of attacking the problem is to list all partitions of the number 15 into parts no greater than 3 and then determine the number of non-isomorphic proper edge-colourings for each partition. We use the standard terminology $3^a2^b1^c$ to represent the partition of 15 into a parts of 3, b parts of 2, and c parts of 1. The results are summarized in Table 3.1 below.

# colour classes	# non-isomorphic proper edge-colourings	Partition type	# non-isomorphic proper edge-colourings	Note
5	1	3^5	1	P
6	5	$3^4 2^1 1^1$	1	P
		$3^3 2^3$	4	P
7	28	$3^4 1^3$	1	P
		$3^3 2^2 1^2$	6	P
		$3^2 2^4 1^1$	11	P
		$3^1 2^6$	10	P
8	94	$3^3 2^1 1^4$	3	E
		$3^2 2^3 1^3$	25	G
		$3^1 2^5 1^2$	48	C
		$2^7 1$	18	C
9	162	$3^3 1^6$	2	E
		$3^2 2^2 1^5$	14	G
		$3^1 2^4 1^4$	74	C
		$2^6 1^3$	72	C
10	140	$3^2 2^1 1^7$	4	E
		$3^1 2^3 1^6$	43	G
		$2^5 1^5$	93	C
11	66	$3^2 1^9$	1	E
		$3^1 2^2 1^8$	13	G
		$2^4 1^7$	52	G
12	21	$3^1 2^1 1^{10}$	2	E
		$2^3 1^9$	19	G
13	5	$3^1 1^{12}$	1	E
		$2^3 1^{11}$	4	E
14	1	$2^1 1^{13}$	1	E
15	1	1^{15}	1	E

Table 3.1

Thus $C(6) = 1 + 5 + 28 + 94 + 162 + 140 + 66 + 21 + 5 + 1 + 1 = 524$. Returning to the original design theory problem which motivated this work, values of $N^{(v-6)}(v)$ are given below.

v:	11	12	13	14	15	16	17	18	19	20	≥ 21
$N^{(v-6)}(v)$:	1	6	34	128	290	430	496	517	522	523	524

The meanings of the letters P (previous), E (easy), G (graph listings) and C (computer) in the Note column of Table 3.1 are as follows. The letter P indicates that these values, for 5, 6 or 7 colour classes, have been obtained previously (Theorem 2.7 of [3]). The reader is referred to that paper for a complete derivation and listing of these results. The fact that the number of non-isomorphic proper edge-colourings corresponding to the partition type 3^5 is 1 is a statement of the fact that, to within isomorphism, the one-factorization of the complete graph K_6 is unique. The letter E indicates that these results are easy to obtain and we leave their determination to the reader. Note that the union of any two one-factors of the complete graph K_6 forms a Hamiltonian cycle and that there are precisely two non-isomorphic sets of three disjoint one-factors. On the vertex set $\{1,2,3,4,5,6\}$ these are $\{12,34,56\}$, $\{23,45,61\}$, $\{14,26,35\}$ and $\{12,34,56\}$, $\{23,45,61\}$, $\{14,25,36\}$. The former extends to a one-factorization whilst the latter does not. The letter G indicates that these values were obtained with the help of listings of graphs given in the *Atlas of Finite Graphs* [6]; details are given in the next section. Finally the letter C indicates that these results were obtained by computer; details of the algorithm are given in Section 5.

4 Graph listings

Consider the partitition $3^a 2^b 1^c$. We make the simple observation that once the a colour classes of 3 edges and the b colour classes of 2 edges have been chosen, the edge colouring is determined because all the remaining edges form colour classes by themselves. Thus the problem is equivalent to determining all non-isomorphic proper edge-colourings, with a colour classes of 3 edges and b colour classes of 2 edges, of each graph with 6 vertices and $3a + 2b$ edges. All graphs on 6 vertices are known; they are illustrated in the *Atlas of Finite Graphs* [6]. This method is particularly useful when the value of c is relatively large and there are a number of pairwise non-isomorphic graphs with 6 vertices and $3a + 2b$ edges. We have employed this method for the six entries marked G in Table 3.1. With respect to explicit listings of the results we adopt an intermediate approach. For each graph, reference number as in [6], we give the number of non-isomorphic proper edge-colourings with a colour classes of 3 edges and b colour classes of 2 edges. It is then a relatively simple matter for the reader to obtain the explicit edge-colourings. The results are given in Table 4.1 below.

$2^3 1^9$ (number of proper edge-colourings = 19)

graph #	86	87	88	89	90	91	92
# colourings	1	0	1	0	1	1	0

graph #	93	94	95	96	97	98	99
# colourings	0	1	0	0	1	2	2

graph #	100	101	102	103	104	105	106
# colourings	1	1	1	1	2	2	1

$3^1 2^2 1^8$ (number of proper edge-colourings = 13)

graph #	107	108	109	110	111	112	113	114
# colourings	0	0	0	0	0	0	0	0

graph #	115	116	117	118	119	120	121	122
# colourings	1	1	0	0	0	0	0	1

graph #	123	124	125	126	127	128	129	130
# colourings	1	1	1	0	1	3	2	1

$2^4 1^7$ (number of proper edge-colourings = 52)

graph #	131	132	133	134	135	136	137	138
# colourings	1	1	0	2	0	0	3	1

graph #	139	140	141	142	143	144	145	146
# colourings	3	2	3	1	2	0	2	2

graph #	147	148	149	150	151	152	153	154
# colourings	4	4	3	1	5	6	3	3

$3^1 2^3 1^6$ (number of proper edge-colourings = 43)

graph #	155	156	157	158	159	160	161
# colourings	0	0	1	0	1	2	0

graph #	162	163	164	165	166	167	168
# colourings	0	1	0	0	5	4	1

graph #	169	170	171	172	173	174	175
# colourings	5	2	4	4	5	6	2

$3^2 2^2 1^5$ (number of proper edge-colourings = 14)

graph #	176	177	178	179	180	181	182	183
# colourings	0	0	0	0	0	1	0	0

graph #	184	185	186	187	188	189	190
# colourings	1	2	1	0	6	1	2

$3^2 2^3 1^3$ (number of proper edge-colourings = 25)

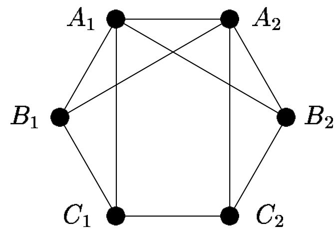
graph #	200	201	202	203	204
# colourings	1	1	8	9	6

Table 4.1

To illustrate the method used in obtaining these results we give more details of the working for the penultimate partition above.

Consider the partition $3^2 2^2 1^5$. Since there are two colour classes of 3 edges, any graph having a vertex of valency 0 or 1 can be eliminated. Similarly, since there are four colour classes of 2 or 3 edges, any graph having a vertex of valency 5 can also be eliminated. This immediately removes graphs 176 to 180, 182, 183 and 187 from consideration. In graphs 181, 186 and 189 there is, up to isomorphism, a unique way of choosing the two colour classes of 3 edges (a Hamiltonian cycle), and the two colour classes of 2 edges are then also uniquely determined. In graphs 184 and 185 there are two non-isomorphic Hamiltonian cycles, but in the former only one of these can be extended to the required edge-colouring with the two colour classes of 2 edges, whilst in the latter both can be so extended. In graph 190, the Hamiltonian cycle is unique up to isomorphism but there are two non-isomorphic extensions with the two colour classes of 2 edges.

This leaves only graph 188, which is more intricate to analyse, and is shown below.



Up to isomorphism there are four Hamiltonian cycles:

1. A1, B1, C1, C2, B2, A2, A1
2. A1, B2, C2, C1, B1, A2, A1
3. A1, C1, B1, A2, C2, B2, A1
4. A1, B1, A2, B2, C2, C1, A1

The first two Hamiltonian cycles each have two non-isomorphic extensions with the two colour classes of 2 edges, whilst the last two each have just one extension. Hence, the number of non-isomorphic proper edge-colourings with two colour classes of 3 edges and two colour classes of 2 edges for this graph is six.

The other partitions can be handled in the same way and are straightforward, with the possible exception of $3^2 2^3 1^3$ where isomorph rejection is more difficult. An alternative approach to this partition follows.

Consider the complete graph K_6 on the vertex set $\{1, 2, 3, 4, 5, 6\}$. Without loss of generality, the two colour classes of 3 edges form a Hamiltonian cycle and can be taken to be 12, 34, 56 and 23, 45, 61. To within isomorphism, a single colour class of 2 edges can be adjoined in four ways: 13, 46 (type a); 13, 25 (type b); 13, 24 (type c); 14, 25 (type d). This is in fact the number of partitions $3^2 2^1 1^7$. Using this classification, the number of non-isomorphic ways of adjoining three colour classes of two edges can be evaluated. These are given in Table 4.2 together with the types of the colour classes and the graph formed with the Hamiltonian cycle.

#	Colour classes			Type	Graph #
1.	13, 46;	24, 51;	35, 62	aaa	204
2.	13, 46;	24, 51;	35, 41	aab	202
3.	13, 46;	24, 51;	36, 52	aad	204
4.	13, 46;	24, 51;	36, 41	aad	202
5.	13, 46;	24, 36;	35, 41	abb	202
6.	13, 46;	24, 36;	26, 41	abb	200
7.	13, 46;	24, 36;	26, 51	abc	203
8.	13, 46;	24, 36;	25, 41	abd	203
9.	13, 46;	24, 35;	15, 26	acc	204
10.	13, 46;	24, 35;	14, 36	acd	202
11.	13, 46;	24, 35;	14, 25	acd	203
12.	13, 25;	36, 51;	35, 41	bbb	201
13.	13, 25;	36, 42;	35, 41	bbb	203
14.	13, 25;	36, 42;	35, 46	bbc	203
15.	13, 25;	36, 42;	15, 46	bbc	204
16.	13, 25;	36, 51;	35, 42	bbc	202
17.	13, 25;	35, 46;	15, 26	bcc	203
18.	13, 25;	35, 42;	15, 26	bcc	202
19.	13, 25;	35, 42;	36, 41	bcd	203
20.	13, 25;	35, 46;	36, 41	bcd	203
21.	13, 24;	35, 46;	51, 62	ccc	204
22.	13, 24;	35, 46;	36, 41	ccd	202
23.	13, 24;	35, 46;	36, 52	ccd	203
24.	13, 24;	46, 51;	36, 52	ccd	204
25.	13, 24;	46, 51;	14, 25	ccd	202

Table 4.2

5 Computer program

In this section we outline the algorithm used by the computer program to determine the proper edge-colourings of K_6. The algorithm is described for the colouring of any complete graph K_n. However for $n > 6$ it generates too many duplicates and a more sophisticated program would be needed. For $k = n-1, n, \ldots, \binom{n}{2}$, we find the number of proper edge-colourings with k colours. Given a value of k, let the colours be $1, 2, \ldots, k$. For a given edge-colouring, let the frequency of colour j be f_j. Without loss of generality, we can assume that f_1 is the greatest frequency in the colouring, and that vertex 1 is incident with edges of colour $1, 2, \ldots, n-1$, where $f_1 \geq f_2 \geq \ldots \geq f_{n-1}$. Since each edge has two endpoints, we can also say that $f_j \leq \lfloor n/2 \rfloor$, for all j. We recursively construct all colourings with colours $1, 2, \ldots, k$ such that vertex 1 is incident with colours $1, 2, \ldots, n-1$, where

1. $f_1 \geq f_2 \geq \ldots \geq f_{n-1}$ and
2. $f_1 \geq f_j$ for $j = n, \ldots, k$.

To do this, we begin with the partial edge-colouring just described (vertex 1 incident with edges of colour $1, 2, \ldots, n-1$) and recursively colour one edge at a time, following the procedure given below. For each vertex u, we maintain a list $R(u)$ of the remaining colours currently allowed at u. We also maintain a sequence $f_1, f_2, \ldots, f_k$ of the current frequencies of the colours. Initially $f_1 = f_2 = \ldots = f_{n-1} = 1$ and $f_j = 0$ if $j \geq n$. As the graph is coloured, the conditions (1) and (2) above may not always be satisfied. However when the colouring is complete, we must be sure that they are indeed satisfied. Given a sequence $f_1, f_2, \ldots, f_k$, we define the *excess* to be the minimum number of edges which must be coloured with colours i currently in use (ie, whose current frequency $f_i > 0$), in order to transform the sequence into a sequence which satisfies conditions (1) and (2) above. The excess can be easily computed by the following statements.

```
Excess := 0
Max := f_{n-1}
for i := n - 1 downto 1 do begin
   if f_i <Max then Excess := Excess + Max - f_i
   else if f_i > Max then Max := f_i
end
Max := f_k
for i := k downto n do begin
   if f_i <Max then Excess := Excess + Max - f_i
   else if f_i > Max then Max := f_i
end
if k > n then if f_1 < Max then Excess := Excess + Max - f_1
```

Max becomes the largest of the f_i, and the number of edges which must be coloured with colour i is accumulated into *Excess*. The edges are coloured in lexicographic order, $(1, 2), (1, 3), \ldots (n-1, n)$. We use a recursive procedure ColourEdge(u, v) to colour the edge uv. Upon entry, if the number of unused colours equals the number of uncoloured edges, then the current partial edge-colouring has a unique completion, and this is constructed. If the edge to be coloured is the last edge, $(n-1, n)$, then it is assigned in turn all remaining colours in $R(n-1) \cap R(n)$. Otherwise we have to proceed recursively. If the sum of the excess and the number of unused colours is greater than the number of uncoloured edges, then the current partial edge-colouring cannot be completed, and we backtrack. Otherwise we try all allowed colours for the edge uv. Every time a colouring is completed, a procedure ConstructGraph is called which constructs a graph encoding the colouring, and outputs it to a file for future processing. We store the colour of edge uv in an array $C[u, v]$. We also store the number of edges currently uncoloured (E_u) and the number of colours currently unused (C_u) as global variables.

```
Procedure ColourEdge(u, v)
{ colour the edge uv, where u < v, in all possible ways }
begin
  if E_u = C_u then begin
    { each remaining edge must receive an unused colour }
    { the unused colours are all equivalent to each other }
    { check that f_1 ≥ f_2 ≥ ... ≥ f_{n-1} }
    { and ensure that f_j ≥ 1, for all j ≥ n }
    tail_of_sequence := false
    for i := 2 to k do begin
      if f_i = 0 then begin
        tail_of_sequence := true
        C[u, v] := i
        { find next edge to colour }
        v := v + 1
        if v > n then begin
          u := u + 1;    v := u + 1
        end
      end
      else begin   { f_i > 0 }
        if f_{i-1} < f_i then return   { frequencies are invalid }
        if tail_of_sequence then if f_i ≠ 1 then return   { frequencies are invalid }
      end
    end
    ConstructGraph
    return   { done }
  end
  { check if uv is the last edge in the graph }
  if v = n and u = n - 1 then begin
    { run through remaining colours at u }
    for each i ∈ R(u) do if i ∈ R(v) then begin
      C[u, v] := i
      ConstructGraph
```

```
    end
    return   { done }
  end
  { check if the Excess will allow a colouring }
  calculate Excess
  if Excess + E_u > C_u then return   { no colouring possible }
  { at this point the Excess is OK, and uv is not the last edge }
  { |R(v)| must be at least n − u, since all edges vx,
    where x > u must still be coloured }
  if |R(v)| + u < n then return   { no colouring possible }
  { try all possible colours for edge uv }
  for each i ∈ R(u) do if i ∈ R(v) then begin
    if f_i > 0 then begin
      { colour i has already been used –
        check that all colours can eventually be used }
      if E_u ≤ C_u then goto 1   { try next colour }
    end
    { OK to assign colour i to edge uv }
    C[u, v] := i
    if f_i = 0 then decrease C_u
    decrease E_u
    f_i := f_i + 1
    { find next edge to be coloured }
    u' := u;   v' := v + 1
    if v' > n then begin
      u' := u + 1;   v' := u' + 1
    end
    remove colour i from R(u) and R(v)
    ColourEdge(u', v')   { recursive call }
    restore colour i to R(u) and R(v)
    f_i := f_i − 1
    if f_i = 0 then increment C_u
    increment E_u
  1:{ try next colour }
  end   { for }
end   { ColourEdge }
```

This program produces a file of graphs encoding the colourings found. The conversion of each colouring to a graph is carried out as follows. A new vertex x_i is created for each colour $i = 1, 2, \ldots, k$. Each edge of K_n is subdivided by a new vertex, and this new vertex is joined to x_i if the edge was coloured i. This creates a graph with $n + k + \binom{n}{2}$ vertices. Two isomorphic edge-colourings produce isomorphic graphs, where colourings equivalent up to permutations of the colours and vertices are considered to be isomorphic (as they are throughout this paper). Non-isomorphic colourings produce non-isomorphic graphs. The file of graphs thus constructed is input to an isomorphism program (see [4]) which produces a certificate for each one. The file of certificates is then sorted, and duplicates removed. The final output consists of a graph and colour matrix for each non-isomorphic edge-colouring. The edge colourings are then sorted according to their partition type. This provides a check on all the values in Table 3.1 which were worked out using methods P, E, or G, and provides an answer in the remaining cases (labelled C in the table).

References

[1] L.D. Andersen, Factorizations of graphs, in *CRC Handbook of Combinatorial Designs* (ed. C.J. Colbourn and J.H. Dinitz), CRC Press (1996), 653–667.

[2] P.J. Cameron, Parallelisms of complete designs, *London Math. Soc. Lecture Notes* **23**, Cambridge University Press (1976), 63–80.

[3] M.J. Grannell, T.S. Griggs, K.A.S. Quinn and R.G. Stanton, A census of minimal pair-coverings with restricted largest block length, *Ars Combinatoria* **52** (1999), 71–96.

[4] W.L. Kocay, On writing isomorphism programs, in *Computational and Constructive Design Theory* (ed. W.D. Wallis), Kluwer Academic Publishers (1996), 135–175.

[5] E. Mendelsohn and A. Rosa, One-factorizations of the complete graph—a survey, *J. Graph Theory* **9** (1985), 43–65.

[6] R.C. Read and R.J. Wilson, *An Atlas of Graphs*, Clarendon Press, Oxford (1998), 9–11.

[7] R.G. Stanton, J.L. Allston and D.D. Cowan, Pair-coverings with restricted largest block length, *Ars Combinatoria* **11** (1981), 85–98.

[8] W.D. Wallis, One-factorizations of complete graphs, in *Contemporary Design Theory: a Collection of Surveys* (ed. J.H. Dinitz and D.R. Stinson), John Wiley and Sons, New York (1992), 593–631.

[9] D.R. Woodall, The $\lambda - \mu$ problem, *J. London Math. Soc.* **1** (1968), 505–519.