THE SEARCH FOR THE SMALLEST 3-E.C. GRAPHS

PRZEMYSŁAW GORDINOWICZ AND PAWEŁ PRAŁAT

Abstract. A graph G is 3-existentially closed (3-e.c.) if each 3-set of vertices can be extended in all of the possible eight ways. Results which improve the lower bound of the minimum order of a 3-e.c. graph are reported. It has been shown that $m_{ec}(3) \geq 24$ where $m_{ec}(3)$ is defined to be the minimum order of a 3-e.c. graph.

1. Introduction

Adjacency properties of graphs have received much attention since Erdős and Rényi [5] first studied them in their pioneering work on random graphs. One such adjacency property is the n-e.c. property. For a positive integer n, a graph is *n-existentially closed* or *n-e.c.*, if for all disjoint sets of vertices A and B with $|A \cup B| = n$ (one of A or B can be empty), there is a vertex z not in $A \cup B$ joined to each vertex of A and no vertex of B. We say that z is *correctly joined* to A and B. Hence, for all n-subsets S of vertices, there exist 2^n vertices joined to S in all possible ways. For example, a graph is 2-e.c. if for each pair of distinct vertices u and v, there are four vertices (distinct from u and v) joined to them in all four possible ways. See Figure 1 for the unique isomorphism type of 2-e.c. graph with the least possible number of vertices. (In [3] it was first noted that $m_{ec}(2) = 9$; in [2] it was observed that $K_3 \square K_3$ is in fact the unique isomorphism

The authors gratefully acknowledge support from ACEnet, NSERC, MITACS, and SHARCNET.

type of 2-e.c. graphs of order 9.) For completeness, every (non-null, that is, with non-empty vertex set) graph is 0-e.c. For a recent survey of n-e.c. graphs, see [1].

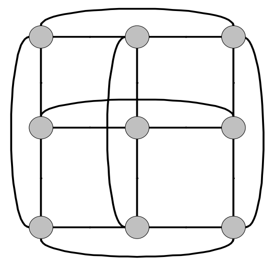


FIGURE 1. The smallest order 2-e.c. graph.

For a positive integer n, the probability that a binomial random graph $G(v, 1/2)$ is *not* n-e.c. is bounded from above by

$$\binom{v}{n} 2^n \left(1 - \frac{1}{2^n}\right)^{v-n}$$

which is smaller than one for v sufficiently large. This guarantees that there is a graph that satisfies this property. Therefore we can define $m_{ec}(n)$ to be the minimum order of an n-e.c. graph. It is not difficult to show that $m_{ec}(1) = 4$ (P_4, C_4, and $K_2 \cup K_2$ are the only 1-e.c. graphs with least possible number of vertices) and, as we already mentioned, $m_{ec}(2) = 9$, but no other values of this function are known. For example, $20 \leq m_{ec}(3) \leq 28$; the upper bound was determined by searching through the vertex-transitive graphs of order 20 and up listed on Gordon Royle's website (see [2] for more details). Analogous to the well-known Ramsey numbers (see a dynamic survey of Radziszowski [7]) it is difficult to compute the exact value of $m_{ec}(n)$, even for $n = 3$, and very little progress has been made up to date. The goal of the present paper is to improve the lower bound, namely, it will be shown that $m_{ec}(3) \geq 24$. In order to obtain this result, a computer support was required to verify that a necessary condition for a graph to be n-e.c. (see Section 2) is not satisfied.

Some additional subtle approaches specific for a given case are discussed in Section 3. We conclude the paper with a few open problems (see Section 4).

All graphs considered are simple, undirected, and finite unless otherwise stated. We denote the complement of the graph G by $\overline{G}$. Let $G[X]$ denote the graph induced by the set $X \subseteq V(G)$. Let $N(v)$ and $N^c(v)$ denote the neighbourhood an non-neighbourhood of v, respectively.

2. A necessary condition

We start with the following necessary condition for a graph to be n-e.c. For $m = n$, the condition we consider is equivalent to one in the definition of being n-e.c.

Theorem 2.1. *Let $n \geq m \geq 1$. If G is n-e.c., then for all disjoint sets of vertices X and Y with $|X \cup Y| = m$ (one of X or Y can be empty), a graph induced by vertex set $Z = Z(X,Y)$ defined as*

$$Z = \left(\bigcap_{x \in X} N(x) \right) \cap \left(\bigcap_{y \in Y} N^c(y) \right)$$

is $(n-m)$-e.c.

Proof. Let $X, Y \subseteq V(G)$, $X \cap Y = \emptyset$, and $|X \cup Y| = m$. We will show that $G[Z]$ is $(n-m)$-e.c. First note that $|Z| \geq 2^{n-m}$. Indeed, since G is n-e.c., for any $(n-m)$-subset $S \subseteq V(G) \setminus (X \cup Y)$ of vertices, there exist 2^{n-m} vertices joined to each vertex of X, no vertex of Y, and S in all possible ways. In particular, $|Z| \geq 1$ and the proof is complete if $n = m$.

Suppose now that $n > m$ and let $A, B \subseteq Z$, $A \cap B = \emptyset$, and $|A \cup B| = n - m$. In order to finish the proof it is enough to show that there is a vertex $z \in Z \setminus (A \cup B)$ that is correctly joined to A and B. But, since G is n-e.c., $X \cap Z = \emptyset$, $Y \cap Z = \emptyset$, there is a vertex $z' \in V(G)$ correctly joined to $A \cup X$ and $B \cup Y$, and clearly $z' \in Z \setminus (A \cup B)$. □

It was an open problem (personal communication of the second author) to determine whether the condition described in

```
The adjacency list:
  0 : 4 5 7 8 10;
  1 : 5 8 9 10;
  2 : 6 7 9 10;
  3 : 6 8 9 10;
  4 : 0 7 9 10;
  5 : 0 1 8 9;
  6 : 2 3 7 8;
  7 : 0 2 4 6;
  8 : 0 1 3 5 6;
  9 : 1 2 3 4 5 10;
 10 : 0 1 2 3 4 9;
```

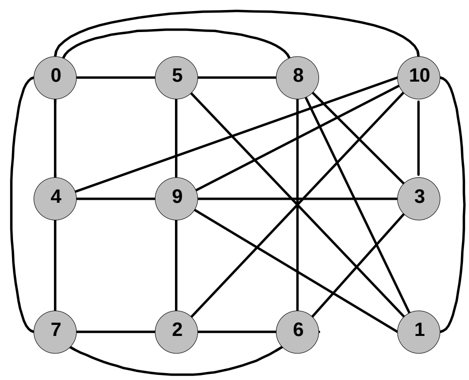


FIGURE 2. The counterexample on 11 vertices.

Theorem 2.1 with $m = 1$ is also a sufficient one. With a computer support, we verified that all graphs on at most 10 vertices satisfying the condition with $n = 2$, $m = 1$ are 2-e.c. but there are counterexamples on 11 vertices (20,058 graphs satisfy the property but only 12,078 of them are 2-e.c.). Below, we present one counterexample; it is easy to check that neighbourhood and non-neighbourhood of each vertex induce 1-e.c. graph but there is no vertex adjacent to 1 and 7. Another counterexamples (on larger number of vertices) can be constructed by taking a disjoint union of at least two 2-e.c. graphs.

3. Improving a lower bound

Since the neighbourhood and non-neighbourhood of each vertex of a 3-e.c. graph induce a 2-e.c. graph and $m_{ec}(2) = 9$, we get a trivial lower bound of 19 for $m_{ec}(3)$. But, clearly, there is no 9-regular graph on 19 vertices (since the number of vertices of odd degree must be even) which yields a lower bound of 20. Until this paper, no progress had been made so far and it seems that there is no 'proof from the Book' that would determine the value of $m_{ec}(3)$. Therefore, we are content with eliminating a few values with computer support. The rest of this section is devoted to prove the following theorem.

Theorem 3.1. $m_{ec}(3) \geq 24$.

Before we move to improving a lower bound, we present a little bit more details about 2-e.c. graphs of small order. As we already mentioned, there is a unique 2-e.c. graph on 9 vertices (up to isomorphism). Below we present more information verified by computer. Let $\mathcal{H}(n)$ be a family of graphs on n vertices, $\mathcal{H}_2(n) \subseteq \mathcal{H}(n)$ be a subfamily containing 2-e.c. graphs only, $\mathcal{E}_{\min}(n) = \min\{|E(H)| : H \in \mathcal{H}_2(n)\}$, and $\mathcal{E}_{\max}(n) = \max\{|E(H)| : H \in \mathcal{H}_2(n)\}$.

n	$\lvert\mathcal{H}(n)\rvert$	$\lvert\mathcal{H}_2(n)\rvert$	$\mathcal{E}_{\min}(n)$	$\mathcal{E}_{\max}(n)$
9	274,668	1	18	18
10	12,005,168	16	21	24
11	1,018,997,864	12,078	23	32
12	165,091,172,592	11,688,811	26	40

TABLE 1. 2-e.c. graphs of small order.

3.1. **Eliminating** 20**.** Suppose that G has 20 vertices and is 3-e.c. Since the neighbourhood and non-neighbourhood of each vertex induce 2-e.c. graph, each vertex has degree 9 or 10. Moreover, there is at least one vertex of degree 10. Suppose for a contradiction that all vertices have degree 9 so $|E(G)| = 90$. But for any vertex $v \in V(G)$, $G[N(v)]$ and $G[N^c(v)]$ are 2-e.c. so $|E(N(v), N(v))| = 18$ and $|E(N^c(v), N^c(v))| \leq 24$ (see Table 1) where $E(X, Y) = \{uv \in E(G) : u \in X, v \in Y\}$. Since

$$\begin{aligned} |E(N(v), N^c(v))| &= \sum_{u \in N(v)} \deg(u) - |N(v)| - 2|E(N(v), N(v))| \\ &= 36, \end{aligned}$$

we get

$$\begin{aligned} |E(G)| &= |N(v)| + |E(N(v), N(v))| \\ &\quad + |E(N(v), N^c(v))| + |E(N^c(v), N^c(v))| \\ &\leq 9 + 18 + 36 + 24 = 87 < 90. \end{aligned}$$

Let v_0 denote a vertex of degree 10 in G, $X = N(v_0)$, $Y = N^c(v_0)$; both $G[X]$ and $G[Y]$ are 2-e.c. Since each vertex has degree 9 or 10, we obtain that

$$81 \leq \sum_{v \in Y} \deg(v) = 2|E(Y,Y)| + |E(X,Y)|$$

so $|E(X,Y)| \geq 45$. Similarly

$$100 \geq \sum_{v \in X} \deg(v) = |X| + 2|E(X,X)| + |E(X,Y)|$$

so $|E(X,Y)| \leq 90 - 2|E(X,X)|$. Note also that a graph G is n-e.c. if and only if $\overline{G}$ is. Hence, without loss of generality, we can assume that

$$\begin{aligned} |E(G)| &= |X| + |E(X,X)| + |E(X,Y)| + |E(Y,Y)| \\ &\leq \binom{20}{2}/2 = 95, \end{aligned}$$

which implies that $|E(X,X)| = 21$ or 22. If $|E(X,X)| = 21$, then $E(X,Y) = 45$ or 46; $E(X,Y) = 45$ otherwise.

In order to show that there is no 3-e.c. graph on 20 vertices we put a 2-e.c. graph on 10 vertices (one out of 8 (not 16) possible ones due to the additional condition) on set X and the only 2-e.c. graph on 9 vertices on set Y. It remains to check that it is not possible to distribute edges between X and Y to satisfy the necessary condition stated in Theorem 2.1.

If $E(X,Y) = 46$, then we have exactly one vertex of degree 10 in Y. Since all vertices in Y are undistinguishable (at this point), we can take any vertex $v_1 \in Y$ and assign to this vertex 6 neighbours from X so that $G[N^c(v_1)]$ is isomorphic to the only 2-e.c. graph on 9 vertices. This, of course, can be done in many different ways. Next, we take any other vertex $v_2 \in Y$ and *try* to assign 5 neighbours from X so that $G[N(v_2)]$ is 2-e.c. (note that this time we check the neighbourhood so, again, the only chance for the necessary condition to hold is that the graph induced by $N(v_2)$ is the only 2-e.c. graph on 9 vertices). We repeat this process to discover that there is no chance for a graph to be 3-e.c. The argument can be repeated for the case $E(X,Y) = 45$;

this time all vertices in Y have degree 9 so one should check neighbourhoods only.

It has been verified that it is not even possible to satisfy three vertices, that is, it is not possible to distribute edges between X and Y so that three vertices in Y satisfy the necessary condition. As we already mentioned, there are 8 possible 2-e.c. graphs we can start with; after verifying the condition for v_1 we get 816 (possibly some of them are isomorphic) configurations; finally, there are only 120 configurations satisfying the necessary condition for v_1, v_2. The running time was below a second.

3.2. **Eliminating 21.** Suppose that G has 21 vertices and is 3-e.c.; each vertex has degree 9, 10, or 11. Let v_0 denote a vertex of degree 10 (note that there is at least one such vertex since the number of vertices of odd degree must be even), $X = N(v_0)$, $Y = N^c(v_0)$.

Similarly as before, we assume, without loss of generality, that

$$\begin{aligned} |E(G)| &= |X| + |E(X,X)| + |E(X,Y)| + |E(Y,Y)| \\ &\le \binom{21}{2}/2 = 105, \end{aligned}$$

which gives $|E(X,Y)| \le 95 - |E(X,X)| - |E(Y,Y)|$. This implies that at least $5 + |E(X,X)| - |E(Y,Y)|$ vertices in Y have degree 9.

In order to check that there is no 3-e.c. graph on 21 vertices we consider 16^2 possible embeddings of 2-e.c. graphs on sets X, Y and all possible distributions of $5 + |E(X,X)| - |E(Y,Y)|$ vertices among those from Y. For each such a vertex v_Y we generate all possible distributions of edges between X and Y so that $N(v_Y)$ is isomorphic to the only 2-e.c. graph on 9 vertices. (Note that sometimes we are guaranteed to have only 2 such vertices and the condition can be satisfied easily but sometimes the number of such vertices is as large as 8 and there is no way to do that.) For each such a distribution of edges, we take a vertex v_X from set X for which the number of determined incident edges and non-edges is maximized and we fix a degree (we have to consider three cases: 9,10,11). If $\deg(v_X) = 9$, then we check the condition for the neighbourhood; if $\deg(v_X) = 11$, then we verify the condition for the non-neighbourhood; otherwise, we deal with both cases in order to decrease the number of configurations that stay to the next round (since the number of 2-e.c. graphs on 11 vertices is large, we avoid considering this case here). We repeat the operation for a next candidate until there is no way to satisfy the necessary condition. It has been verified that 37 pairs (out of 256) of 2-e.c. graphs can be used to satisfy vertices in Y but only 8 pairs can satisfy an additional condition for the first vertex from X; 2 pairs survive to the next round and the process ends. It took a few minutes for a computer to eliminate order 21.

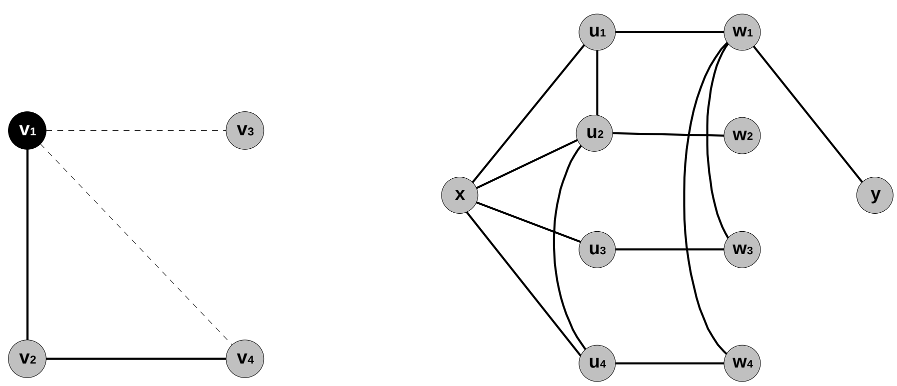
3.3. **Eliminating 22.** Suppose that G has 22 vertices and is 3-e.c.; each vertex has degree 9, 10, 11, or 12. Since if G is 3-e.c. so is $\overline{G}$, we can assume that there is a vertex v_0 of degree at most 10, $X = N(v_0)$. If $\deg(v_0) = 9$, then we put the only 2-e.c. graph on 9 vertices in the neighbourhood of v_0; otherwise we have to check all 16 graphs on 10 vertices independently. Experiments show that it is better to resign with determining graphs in the non-neighbourhood of v_0 at this point due to the large number of initial configurations to consider. Moreover, without fixing this, the number of non-isomorphic configurations we have to deal with in a first few steps of the process is much smaller.

Similarly as before, at each step of the process, we sort vertices that are not satisfied yet with respect to the number of determined edges and non-edges (so that the number of branches into we split the process is as small as possible), and *try* to distribute edges and non-edges so that the necessary conditions are satisfied. We avoid considering 2-e.c. graphs on more than 10 vertices by checking the condition for neighbourhood (or non-neighbourhood if vertex has degree 11 or 12) only.

It is possible that we can satisfy a vertex using different edge distributions which can yield a large number of configurations that are isomorphic to each other. The main improvement here, required to solve the problem, is the following. After considering all possibilities for X, fixing the degree of the first vertex, and satisfying this vertex, we store all possible configurations and remove any isomorphic copies. Next, for each configuration, we fix degree of the next vertex, *try* to satisfy this vertex, and store the result again. We repeat this procedure trying to satisfy a new vertex at each round until there is no configuration to consider.

Moreover, we improve the running time of the algorithm dramatically by checking (at each step) the necessary condition stated in Theorem 2.1. After satisfying vertex v_1, we check the condition with $m = 2$ for the two vertices that are satisfied at this point, that is, vertices v_0, v_1. All configurations that fail this test are removed. At the next steps, after satisfying a new vertex v_i, the additional test is checked for $m = 2$ and $m = 3$, and for all sets of satisfied vertices containing vertex v_i we deal with at the current round. The operation of removing isomorphisms, together with checking the additional condition, can decrease the number of configurations by even 90% (at each round) so without this improvement the number 22 would not be eliminated. The total computational requirements can be estimated to be 70 CPU hours.

In order to remove unnecessary configurations we use Brendan McKay's `nauty` software package [6] for computing automorphism groups of graphs and digraphs. We cannot use, however, the package directly since in our situation each pair of vertices uv can be in three different stages, say: $s(uv) = 2$ if there is no edge uv, $s(uv) = 1$ if there is an edge uv, and finally $s(uv) = 0$ indicates that the existence of an edge uv is not determined yet. Moreover, we need to keep the information of which vertices are satisfied (note that this cannot be determined based on the function s; $|\{uv : v \in V \text{ and } s(uv) = 0\}| = 0$ is only a necessary condition for u to be satisfied). To overcome this problem we introduce a bijection from our configuration to a (binomial) graph H on $2|V(G)| + 2$ vertices. Let $V(G) = \{v_1, v_2, \dots, v_n\}$ and let $V(H) = \{u_1, u_2, \dots, u_n\} \cup \{w_1, w_2, \dots, w_n\} \cup \{x, y\}$. Now, we construct H as follows: $u_i u_j \in E(H)$ if and only if $s(v_i v_j) = 1$ (this corresponds to the edges of G), $w_i w_j \in E(H)$ if and only if $s(v_iv_j) = 2$ (non-edges), and $u_iw_i \in E(H)$ for $i \in [n]$. Moreover, $xu_i \in E(H)$ for $i \in [n]$ in order to keep an information which vertices correspond to the original graph (note that x is the only vertex of degree n in H). Finally, $yw_i \in E(H)$ if v_i is satisfied. (An example of this transformation is depicted on Figure 3; vertex v_1 is satisfied, there are edges v_1v_2 and v_2v_4; v_1v_3 and v_1v_4 are non-edges.) It is clear that we can reconstruct the graph G, together with the information of which vertices are satisfied, from H.



Graph G Graph $H = H(G, \{v_1\})$

FIGURE 3. Transformation.

3.4. **Eliminating 23.** Suppose that G has 23 vertices and is 3-e.c.; each vertex has degree between 9 and 13. Note that there is at least one vertex of even degree since the number of vertices of odd degree must be even. Since if G is 3-e.c. so is $\overline{G}$, without loss of generality, we can assume that there is a vertex v_0 of degree 10, $X = N(v_0)$. The main idea is the same as before: we start with embedding a 2-e.c. graph on 10 vertices into X and *try* to satisfy vertices one by one. However, there are several adjustments that have to be made to adopt the approach to this case. Let us mention a few of the most important ones.

In the previous cases, we were able to avoid considering 2-e.c. graphs on more than 10 vertices whereas now we have to take care of the situation where the size of the neighbourhood of v_i (and thus non-neighbourhood as well) is 11. Since the number of 2-e.c. graphs on 11 vertices is large (see Table 1), it is very time-consuming to find all distributions of edges so that the neighbourhood and the non-neighbourhood of v_i induce 2-e.c. graphs. In order to overcome this problem, we take a vertex $w \in N(v_i)$ for which the number of determined incident edges and non-edges in $G[N(v_i)]$ is maximized. By Theorem 2.1, both $N(v_i) \cap N(w)$ and $N(v_i) \cap N^c(w)$ induce 1-e.c. graph, that is, $N(v_i) \setminus \{w\}$ has to be decomposed into two disjoint sets A, B, $4 \leq |A|, |B| \leq 6$, $|A| + |B| = 10$, so that each set induces 1-e.c. graph. There are only 688 different graphs $G[A]$ and $G[B]$ that, together with w and some edges between A and B can yield 2-e.c. graph on 11 vertices. All possible pairs, pre-computed in advance and stored in an appropriate data structure (for instance, a decision tree has been used), allows us to determine possible decompositions efficiently. Finally, it remains to *try* to generate all configurations of edges and non-edges between A and B to get 2-e.c. graph on 11 vertices. Since some pairs ab, $a \in A$, $b \in B$ have already determined value, this can be done efficiently using pre-computed decision tree. A similar approach is used for the non-neighbourhood.

The second improvement we would like to mention is the following. At each round of the process, we verify the 3-e.c. condition in order to check that a given partial configuration has a chance to yield the 3-e.c. graph. In other words, for each of $\binom{23}{3}$ triples of vertices and *try* to find 8 vertices joint to them in all possible ways. Of course, a pair of vertices for which the status is not determined yet (that is, $s(uv) = 0$) is treated as an edge ($s(uv) = 1$) or a non-edge ($s(uv) = 2$) depending on the case. This eliminates a large number of configurations that are hopeless anyway.

It took approximately 5 CPU hours to eliminate all degree distributions with at least 2 vertices of degree 9 or 13. In order to check all configurations with exactly one vertex of degree 9 or 13, roughly 136 CPU hours are needed. Finally, it took approximately 15,000 CPU hours to check that there is no 3-e.c. graph on 23 vertices with no vertex of degree 9 or 13. The total computational requirements we estimated to be 15,141 CPU hours.

4. Open problems

Using the fact that in an n-e.c. graph with $n > 1$, the neighbour and non-neighbour sets of each vertex are $(n-1)$-e.c., we have that $m_{ec}(n) \geq 2m_{ec}(n-1)+1$. As $m_{ec}(3) \geq 24$ and by a simple (and so omitted recursion), we derive that

$$m_{ec}(n) \geq \frac{25}{8} \cdot 2^n - 1.$$

On the other hand, using the random graph $G(v, 1/2)$ one has that $m_{ec}(n) = O(n^2 2^n)$. From this it follows that

$$\lim_{n\to\infty} m_{ec}(n)^{1/n} = 2.$$

One of the most important open problems in this area is to determine whether

$$\lim_{n\to\infty} \frac{m_{ec}(n)}{2^n}$$

exists and, if so, to find its value.

The probability space $G(\mathbb{N}, p)$ is defined analogously as the random graph $G(v, p)$, but with vertices the nonnegative integers. Erdős and Rényi [5] proved that with probability 1 a graph in $G(\mathbb{N}, p)$, where $p \in (0, 1)$, is isomorphic to a unique graph, called the *infinite random graph* or *Rado graph*. The Rado graph R is deterministic and has a rich structure; it is the unique isomorphism type of countable graphs that is n-e.c. for all $n > 0$.

In Section 2, we presented the counterexample showing that the conjecture that the condition stated in Theorem 2.1 is sufficient for a graph to be n-e.c. is false. But the following question of the same flavour is still open. (This problem was proposed by Anthony Bonato; see [4], problem 20.) Suppose that a graph G has the property that a graph induced by the neighbourhood (and the non-neighbourhood) of each vertex of G is isomorphic to G. It is clear that R has this property but is there any other graph satisfying this property?

5. Acknowledgement

This work was made possible by the facilities of

- the Shared Hierarchical Academic Research Computing Network SHARCNET, Ontario, Canada (www.sharcnet.ca): 8,082 CPUs, and
- the Atlantic Computational Excellence Network ACEnet, Memorial University of Newfoundland, St. John's, NL, Canada (www.ace-net.ca): 412 CPUs.

The programs used to obtain the result can be downloaded from [8].

References

[1] A. Bonato, The search for n-e.c. graphs, *Contributions to Discrete Mathematics*, in press.
[2] A. Bonato, K. Cameron, On an adjacency property of almost all graphs, *Discrete Mathematics* **231** (2001) 103–119.
[3] L. Caccetta, P. Erdős, K. Vijayan, A property of random graphs, *Ars Combinatoria* **19** (1985) 287–294.
[4] P.J. Cameron, Research problems from the 18th British Combinatorial Conference. The 18th British Combinatorial Conference (Brighton, 2001). *Discrete Mathematics* **266** (2003), no. 1-3, 441–451.
[5] P. Erdős, A. Rényi, Asymmetric graphs, *Acta Mathematica Academiae Scientiarum Hungaricae* **14** (1963) 295–315.
[6] B.D. McKay, `nauty` Users Guide (Version 2.4), `http://cs.anu.edu.au/~bdm/nauty/`.
[7] S. Radziszowski, Small Ramsey Numbers, *Electronics Journal of Combinatorics*, Dynamic Survey DS1, revision #11 (2006), 60pp.
[8] Programs written in C/C++, `http://www.mathstat.dal.ca/~pralat/index.php?page=publications`.

Department of Mathematics, Technical University of Lodz, Łódź, Poland
E-mail address: pgordin@p.lodz.pl

Department of Mathematics and Statistics, Dalhousie University, Halifax, NS, Canada
E-mail address: pralat@mathstat.dal.ca