TRACEABILITY IN SMALL CLAW-FREE GRAPHS

JOHN M. HARRIS AND MICHAEL J. MOSSINGHOFF

ABSTRACT. We prove that a claw-free, 2-connected graph with fewer than 18 vertices is traceable, and we determine all non-traceable, claw-free, 2-connected graphs with exactly 18 vertices and a minimal number of edges. This complements a result of Matthews on Hamiltonian graphs.

1. INTRODUCTION

A graph G that does not contain a particular graph H as an induced subgraph is said to be *H-free*; equivalently, H is said to be a *forbidden subgraph* of G. The most commonly studied forbidden subgraph is the *claw*, the complete bipartite graph $K_{1,3}$. Other commonly studied forbidden subgraphs include the *paw* and the *net*. These three graphs are shown in Figure 1. Surveys of properties of claw-free and similar graphs appear in [2] and [3].

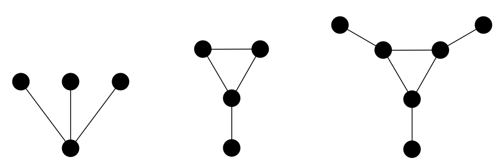


FIGURE 1. The claw, the paw, and the net.

A graph is *traceable* if it contains a spanning path and *Hamiltonian* if it contains a spanning cycle. These properties are often studied in connection with forbidden subgraphs. In 1974, Goodman and Hedetniemi [5] proved that a 2-connected, claw-free, paw-free graph is Hamiltonian. Duffus, Gould, and Jacobson [1] obtained a stronger theorem in 1980, showing that a 2-connected, claw-free, net-free graph is Hamiltonian. They also proved that a connected, claw-free, net-free graph is traceable.

1991 *Mathematics Subject Classification.* Primary: 05C35, 05C38; Secondary: 05C45, 90C35.

Key words and phrases. Forbidden subgraphs, claw-free, traceable, Hamiltonian.

Matthews [6] (see also [7]) proved the following theorem in 1982, and showed that this result is sharp by considering the graph shown in Figure 2. This graph is the line graph of the graph obtained by subdividing a perfect matching in the Petersen graph.

Theorem 1 (Matthews)**.** *If G is a* 3*-connected, claw-free graph of order less than* 20*, then G is Hamiltonian.*

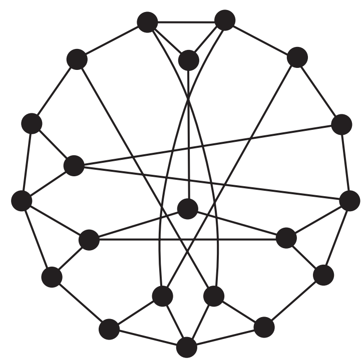
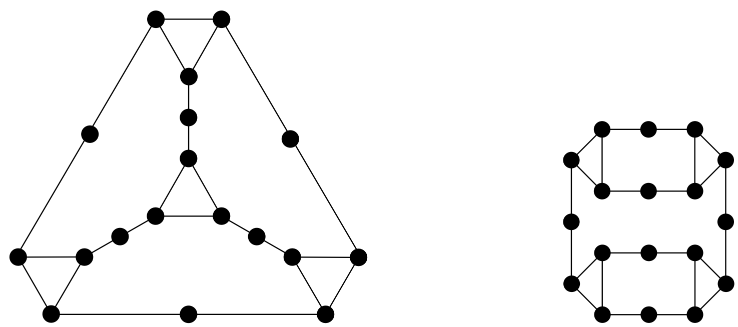


FIGURE 2. The Matthews graph.

Mills [8] studied a similar problem for traceable graphs, assuming one less degree of connectivity, by analogy with the results of Duffus, Gould, and Jacobson. She found that the minimal order n of a 2-connected, claw-free, non-traceable graph satisfies $13 \leq n \leq 18$.

From another point of view, these bounds are not very surprising. In [4] Fronček, Ryjáček, and Skupień used the concept of closure in claw-free graphs (introduced by Ryjáček in [9]) to relate traceability to clique coverings. A claw-free graph is said to be *closed* if the neighborhood of each of its vertices consists of at most two disjoint cliques. Every claw-free graph has a unique closure, and a graph and its closure share several properties, including the maximum length of a cycle and the maximum length of a path within the graph. The *clique covering number* of a graph G, denoted $\theta(G)$, is the minimum number of cliques required to cover the vertices of G. A result in [4] implies that if G is 2-connected, claw-free, closed, and non-traceable, then $\theta(G) \geq 6$. The fact that at least six cliques are required in a covering suggests of course that such a graph cannot have very small order.

The upper bound of 18 on the minimal order of a 2-connected, claw-free, non-traceable graph is attained by the graphs H_1 and H_2 shown in Figure 3; each consists of four nets connected in a particular way. Both graphs are in fact closed, and their clique covering numbers are 9 and 8, respectively.



(a) H_1 (b) H_2

FIGURE 3

Mills conjectured that 18 is in fact the minimal order of a 2-connected, claw-free, non-traceable graph. We prove this conjecture in this paper, and we determine all 2-connected, claw-free, non-traceable graphs with eighteen vertices and a minimal number of edges. More precisely, we prove the following theorem.

Theorem 2. *If G is a 2-connected, claw-free graph of order less than 18, then G is traceable. Further, if G is a 2-connected, claw-free, non-traceable graph with order 18 and size at most 24, then G is isomorphic to H_1 or H_2.*

Section 2 discusses some preliminary results and describes our strategy for proving the theorem. Section 3 describes an algorithm we develop for the proof, and Section 4 summarizes its results.

2. Preliminary Results

In order to prove the first part of Theorem 2, we establish the following proposition.

Proposition 1. *If $n \leq 17$ and G is a 2-connected, claw-free graph with at least n vertices, then there exists a path in G with at least n vertices.*

The proposition is clearly trivial for $n \leq 3$. Our proof of Theorem 2 uses the validity of the proposition for $n-1$ to establish it for n when $n \leq 17$.

Suppose then that the proposition is true for $n-1$, and that G is a 2-connected, claw-free graph having at least n vertices. Then G contains a path P with $n-1$ vertices, labeled consecutively $v_1, v_2, \ldots, v_{n-1}$, as well as a vertex v_0 not on this path. By a theorem of Whitney [10], there exist two disjoint paths connecting v_0 to P at v_a and v_b, and we assume $a < b$. Let P_a denote the path between v_0 and v_a and P_b the path between v_0 and v_b.

The following lemmas from Mills [8] allow us to place some restrictions on G.

Lemma 1. *Using the notation above, if G does not contain a path having n vertices, then $a \geq 3$, $b \leq n-3$, $b-a \geq 4$, and the edges $v_{a-1}v_{a+1}$ and $v_{b-1}v_{b+1}$ are present in G.*

Proof. Let w denote the vertex on P_a adjacent to v_a. Since G is claw-free, one of the edges $v_{a-1}w$, $v_{a+1}w$, or $v_{a-1}v_{a+1}$ must be present in G. The first two cases allow us to splice w into P, yielding a path with n vertices. Thus $v_{a-1}v_{a+1}$ is an edge, and similarly $v_{b-1}v_{b+1}$ is an edge.

Clearly $a \neq 1$, and if $a = 2$ then v_1 is adjacent to v_3 by the previous argument, and we obtain the longer path P_a, v_1, v_3, v_4, $\ldots$, v_{n-1}. Thus $a \geq 3$, and similarly $b \leq n-3$.

Last, it is clear that a longer path exists when $b = a+1$. If $b = a+2$, we obtain the path v_1, $\ldots$, v_{a-1}, v_{a+1}, v_a, P_a, P_b, v_{b+1}, $\ldots$, v_{n-1}, and if $b = a+3$, we have the path v_1, $\ldots$, v_{a-1}, v_{a+1}, v_a, P_a, P_b, v_{b-1}, v_{b+1}, $\ldots$, v_{n-1}. Thus $b - a \geq 4$. □

Lemma 2. *Using the notation above, if G does not contain a path having n vertices, then the vertex v_1 is adjacent to a vertex v_k on P, with $k \geq 3$ and $k \notin \{a, a+1, a+2, b-2, b-1, b, b+1, b+2, n-1\}$. Similarly, the vertex v_{n-1} is adjacent to a vertex v_ℓ with $\ell \leq n-3$ and $\ell \notin \{1, a-2, a-1, a, a+1, a+2, b-2, b-1, b\}$.*

Proof. Since G is 2-connected, every vertex has degree at least 2. The vertex v_1 is not adjacent to any vertex off the path P; otherwise a longer path would exist. Further, we may construct a longer path if we select k from the set above, as shown in the following table.

Edge	**Path**
v_1v_{n-1}	$v_{a+1}, \ldots, v_{n-1}, v_1, \ldots, v_a, P_a$.
v_1v_a	$v_{n-1}, \ldots, v_{a+1}, v_{a-1}, \ldots, v_1, v_a, P_a$.
v_1v_{a+1}	$v_{n-1}, \ldots, v_b, P_b, P_a, v_{a-1}, \ldots, v_1, v_{a+1}, \ldots, v_{b-1}$.
v_1v_{a+2}	$v_{n-1}, \ldots, v_{a+2}, v_1, \ldots, v_{a-1}, v_{a+1}, v_a, P_a$.
v_1v_{b-2}	$v_{n-1}, \ldots, v_{b+1}, v_{b-1}, v_b, P_b, P_a, v_{a+1}, \ldots, v_{b-2}, v_1, \ldots, v_{a-1}$.
v_1v_{b-1}	$v_{n-1}, \ldots, v_b, P_b, P_a, v_{a-1}, \ldots, v_1, v_{b-1}, \ldots, v_{a+1}$.
v_1v_b	$v_{n-1}, \ldots, v_{b+1}, v_{b-1}, \ldots, v_1, v_b, P_b$.
v_1v_{b+1}	$v_{n-1}, \ldots, v_{b+1}, v_1, \ldots, v_b, P_b$.
v_1v_{b+2}	$v_{n-1}, \ldots, v_{b+2}, v_1, \ldots, v_{b-1}, v_{b+1}, v_b, P_b$.

The argument for v_{n-1} is clearly symmetric. □

We remark that Lemmas 1 and 2 yield Proposition 1 immediately for $n \leq 11$. We develop an algorithm to establish the proposition for $12 \leq n \leq 17$.

3. The Algorithm

Any internal vertices on the paths P_a and P_b are unimportant in our method, so we may replace the path P_a by the single edge v_0v_a and P_b by the edge v_0v_b. We develop an algorithm to search for 2-connected, claw-free graphs with n vertices which have a path containing $n-1$ vertices, but no path with n vertices. We call a graph with this last property *nearly traceable.*

Algorithm 1. *Nearly traceable,* 2*-connected, claw-free graphs.*

Input. A positive integer n.
Output. All nearly traceable, 2-connected, claw-free graphs of order n.
Description.

Step 0. Create a graph with n vertices labeled v_0, v_1, …, v_{n-1}, and add edges v_iv_{i+1} for $1 \le i \le n-2$.

Step 1. Connect v_0 to two vertices, v_a and v_b.
We assume that $a < b$, and by symmetry we need only consider cases where v_a is at least as close to v_1 as v_b is to v_{n-1}, so $a+b \le n$. Using Lemma 1, this implies we must test each pair (a, b) satisfying $3 \le a \le \lfloor n/2 \rfloor - 2$ and $a+4 \le b \le n-a$. For each such pair (a, b), we perform the following actions.

- Add the four edges v_0v_a, $v_{a-1}v_{a+1}$, v_0v_b, and $v_{b-1}v_{b+1}$.
- Do Step 2.
- Remove these four edges.

Step 2. Add another edge at v_1.
Since we are looking for 2-connected graphs, we must connect v_1 to another vertex v_k, and by Lemma 2 we may assume that $3 \le k \le n-2$ and $k \notin \{a, a+1, a+2, b-2, b-1, b, b+1, b+2\}$. Further, we do not need to test $k = 3$, for if we add v_1v_3 to the graph, then vertex v_3 is a cut vertex, and we would need to add another edge emanating from either v_1 or v_2. However, v_1 and v_2 play interchangeable roles in this graph, so without loss of generality we may add another edge at v_1. Thus the case $k = 3$ reduces to a case with $k > 3$.
For each k in the range described, we perform the following actions.

- Add edge v_1v_k.
- If the graph is not traceable, continue with Step 3.
- Remove edge v_1v_k.

Step 3. Add an edge to destroy the claw induced by v_1, v_{k-1}, v_k, and v_{k+1}.
There are three possible edges we can add to declaw the graph: $v_{k-1}v_{k+1}$, v_1v_{k-1}, and v_1v_{k+1}. We need only test the first two cases here, since adding v_1v_{k+1} yields the same graph that arises

when $v_1v_{k'}$ and $v_1v_{k'-1}$ are added when $k' = k+1$. Thus, we perform the following steps.

- Add edge v_1v_{k-1}.
- If the graph is not traceable, do Step 4.
- Remove edge v_1v_{k-1} and add edge $v_{k-1}v_{k+1}$.
- If the graph is not traceable, do Step 4.
- Remove edge $v_{k-1}v_{k+1}$.

Step 4. Add another edge at v_{n-1}.
This is analogous to Step 2: for each ℓ with $2 \le \ell \le n-4$ and $\ell \notin \{a-2, a-1, a, a+1, a+2, b-2, b-1, b\}$, do the following.

- Add edge $v_\ell v_{n-1}$.
- If the graph is not traceable, do Step 5.
- Remove edge $v_\ell v_{n-1}$.

Notice that after edge $v_\ell v_{n-1}$ is added, a claw is always induced centered at v_ℓ, unless $\ell = k$ and edge $v_{k-1}v_{k+1}$ was added in Step 3. However, it is easy to see that the graph is traceable in this case, so we may assume that a claw is present at the beginning of Step 5.

Step 5. Add an edge to destroy the claw induced by $v_{\ell-1}$, v_ℓ, $v_{\ell+1}$, and v_{n-1}.
As in Step 3, we perform the following steps.

- Add edge $v_{\ell+1}v_{n-1}$.
- If the graph is not traceable, do Step 6.
- Remove edge $v_{\ell+1}v_{n-1}$ and add edge $v_{\ell-1}v_{\ell+1}$.
- If the graph is not traceable, do Step 6.
- Remove edge $v_{\ell-1}v_{\ell+1}$.

Step 6. Check for cut vertices.
If the graph is 2-connected, then report this graph, since it is also claw-free and non-traceable. Otherwise, select a cut vertex v_c and perform Step 7. Notice that c must satisfy either $k \le c < a$ or $b < c \le \ell$.

Step 7. Circumvent the cut vertex v_c.
We assume here that $c < a$; analogous remarks apply for $c > b$.
In general, we must test every possible edge v_rv_s with $1 \le r < c$ and $c < s < n$, but we can do better in some common situations. If $c = k$, then necessarily the edge v_1v_{k-1} was added in Step 3, so v_i and v_{k-i} have interchangeable roles in the graph for $1 \le i < k$. Also, since each of the vertices $\{v_2, v_3, \ldots, v_{k-2}\}$ is incident only with the edges from the path installed in Step 0, it suffices to check at most two of these vertices as endpoints of the new edge. If $c = 4$ we need only test $r = 1$ and 2, and if $c = k$ with $k > 4$ we need

only test $r = 1$, 2, and 3. Of course, for $r = 1$ we can use Lemma 2 to reduce the number of selections for s in this case.
For each selection of r and s, we perform the following actions.
- Add edge $v_r v_s$.
- If the graph is not traceable, then test for induced claws centered at v_r or v_s.
  - If no claws are present, return to Step 6.
  - If one claw is present, augment the graph with each of the three declawing edges in turn, and test whether each resulting graph is traceable. Repeat Step 6 for any non-traceable cases.
  - If two claws are present, augment the graph with each possible pair of declawing edges, and repeat Step 6 for any non-traceable cases.
- Remove edge $v_r v_s$.

4. Results

We coded this algorithm in C++ and use it to establish Theorem 2. Establishing the first part of the theorem requires approximately thirteen seconds of CPU time on a Sun UltraSPARC computer. At most two iterations of Step 6 are enough to dispatch every case for orders n satisfying $12 \leq n \leq 17$: Table 1 shows the number of graphs augmented by each successive iteration of Step 6 for these graphs. Figure 4 shows the output produced by the program with $n = 14$ when $a = 5$ and $b = 9$, and shows Step 7 handling the single exceptional case.

n	First	Second	Third
12	0		
13	0		
14	1	0	
15	4	0	
16	22	10	0
17	65	60	0

Table 1. Number of graphs augmented by successive iterations of Step 6.

For the second part of the theorem, we run our algorithm with $n = 18$. Our program finds exactly seventeen labeled, claw-free, 2-connected, non-traceable graphs when Step 6 executes the first time: seven of these are isomorphic to H_1; ten to H_2. Step 7 is called on an additional 195 graphs, producing several hundred additional graphs with size greater than 24 to be analyzed by a second iteration of Step 6. We stop the program at this

```
Adding edges {0,5}, {0,9}, {4,6}, {8,10}.
 Left case 1: Adding edge {1,12}...Not traceable.
  Adding edge {1,11}...Traceable: 13,12,11,1,2,3,4,5,0,9,10,8,7,6.
  Adding edge {11,13}...Traceable: 3,2,1,12,13,11,10,8,7,6,4,5,0,9.
 Left case 2: Adding edge {1,4}...Not traceable.
  Adding edge {1,3}...Not traceable.
   Right case 1: Adding edge {2,13}.
    Traceable: 1,3,2,13,12,11,10,8,7,6,4,5,0,9.
   Right case 2: Adding edge {10,13}.
    Not traceable.
     Adding edge {11,13}...Not traceable. Cut vertices: 10 4.
     Enforcing 2-connectedness at vertex 4.
      with edge {1,12}...Traceable: 2,3,1,12,11,13,10,8,7,6,4,5,0,9.
      with edge {2,5}...Traceable: 7,6,4,1,3,2,5,0,9,8,10,11,12,13.
      with edge {2,6}...Traceable: 7,6,2,1,3,4,5,0,9,8,10,11,12,13.
      with edge {2,7}...Traceable: 1,3,2,7,6,4,5,0,9,8,10,11,12,13.
      with edge {2,8}...Traceable: 1,3,2,8,7,6,4,5,0,9,10,11,12,13.
      with edge {2,9}...Traceable: 1,3,2,9,0,5,4,6,7,8,10,11,12,13.
      with edge {2,10}...Traceable: 5,0,9,8,7,6,4,1,3,2,10,11,12,13.
      with edge {2,11}...Traceable: 1,3,2,11,12,13,10,8,7,6,4,5,0,9.
      with edge {2,12}...Traceable: 1,3,2,12,11,13,10,8,7,6,4,5,0,9.
      with edge {2,13}...Traceable: 1,3,2,13,12,11,10,8,7,6,4,5,0,9.
     Adding edge {9,11}...Traceable: 1,2,3,4,5,0,9,11,12,13,10,8,7,6.
  Adding edge {3,5}...Traceable: 13,12,11,10,8,7,6,4,1,2,3,5,0,9.
```

FIGURE 4. Output from $n = 14$ when $a = 5$ and $b = 9$.

point. This last computation requires approximately thirty seconds of CPU time.

References

[1] D. Duffus, R. J. Gould, and M. S. Jacobson, *Forbidden subgraphs and the Hamiltonian theme*, in G. Chartrand et al., eds., *The Theory and Applications of Graphs: Fourth International Conference* (Wiley, New York, 1981), 297–316.

[2] R. Faudree, *Forbidden subgraphs and Hamiltonian properties: A survey*, in G. Chartrand and M. Jacobson, eds., *Surveys in Graph Theory* (San Francisco, 1995), Congr. Numer. **116** (1996), 33–52.

[3] R. Faudree, E. Flandrin, and Z. Ryjáček, *Claw-free graphs — A survey*, Discrete Math. **164** (1997), 87–147.

[4] D. Fronček, Z. Ryjáček, and Z. Skupień, *On traceability and 2-factors in claw-free graphs*, Discuss. Math. Graph Theory **24** (2004), 55–71.

[5] S. Goodman and S. Hedetniemi, *Sufficient conditions for a graph to be Hamiltonian*, J. Combin. Theory Ser. B **16** (1974), 175–180.

[6] M. M. Matthews, *Every 3-connected $K_{1,3}$-free graph with fewer than 20 vertices is Hamiltonian*, Technical Report 82-004, Dept. Comp. Sci., Univ. South Carolina, 1982.

[7] M. M. Matthews and D. P. Sumner, *Hamiltonian results in $K_{1,3}$-free graphs*, J. Graph Theory **8** (1984), 139–146.
[8] J. B. Mills, *Traceability results in $K_{1,3}$-free graphs*, Masters Thesis, Dept. Math. Sci., Appalachian State Univ., 1997.
[9] Z. Ryjáček, *On a closure concept in claw-free graphs*, J. Combin. Theory Ser. B **70** (1997), 217–224.
[10] H. Whitney, *Congruent graphs and the connectivity of graphs*, Amer. J. Math. **54** (1932), 150–168.

Department of Mathematics, Furman University, Greenville, South Carolina 29613
E-mail address: John.Harris@furman.edu

Department of Mathematics, Davidson College, Davidson, North Carolina 28035
E-mail address: mjm@member.ams.org