Recognizing Berge Graphs

Maria Chudnovsky and Paul Seymour [1]
Princeton University, Princeton NJ 08544

November 15, 2002; revised January 30, 2003

[1]Supported by ONR grant N00014-01-1-0608, and NSF grant DMS-0070912.

Abstract

A graph is *Berge* if no induced subgraph of G is an odd cycle of length at least 5 or the complement of one. In this paper we give an algorithm to test if a graph G is Berge, with running time $O(|V(G)|^9)$. This is independent of the recent proof of the strong perfect graph conjecture.

1 Introduction

A graph is *Berge* if no induced subgraph of G is an odd cycle of length at least 5 or the complement of one. It has been an open question whether there is a polynomial algorithm to test whether a graph is Berge (or even whether testing Bergeness belongs to NP). There have been two recent developments that increase the interest of this question.

First, in joint work with Neil Robertson and Robin Thomas, we showed in a recent paper [1] that a graph is Berge if and only if it is perfect. So we can test whether G is perfect by applying an algorithm to test whether G is Berge, and being able to test whether a graph is perfect might have practical applications, as well as being of theoretical interest.

Second, in the same paper we proved a stronger statement, that every Berge graph either lies in one of five basic classes, or admits a useful kind of decomposition. There have been a number of other decomposition theorems of this kind for other kinds of graph properties, and in each case, the decomposition theorem yielded a polynomial time algorithm to test for the property in question (we just test whether the graph has the corresponding structure). It was therefore natural to expect that the same would happen again, that our new decomposition theorem would yield a polynomial time algorithm to recognize Berge graphs.

However, this seemed not to be the case, at least not in any straightforward manner. The decomposition theorem did not work well in an algorithmic context, and there remained serious difficulties that gave us a lot of trouble. What one would really like is a "construction" for Berge graphs — a result saying that a graph is Berge if and only if it can be constructed from some simple set of basic graphs by piecing them together in a well-defined way, such that one can test directly whether a graph can be pieced together in this way. The decomposition theorem does not exactly say that, and so far we have not been able to find anything that one might reasonably call a construction.

Building on earlier work that we had done joint with Robertson and Thomas, we did eventually find a polynomial time algorithm to test Bergeness, based on a "cleaning" method. Basically, given a graph G, we generate polynomially many subgraphs of it, so that if G has an odd hole, then one of these subgraphs has an easily detectable odd hole; and then we examine all these subgraphs separately. (This is an algorithm, but evidently it is far from a construction.)

Our first version of this used the decomposition theorem in several places, and then we started trying to simplify it. To our great surprise, we found that in each place where the decomposition theorem was used, we could eliminate it. Overcoming the "serious difficulties" mentioned above was the heart of the whole problem, and the decomposition theorem was only a crutch that could be cast away. Consequently, we came to a polynomial time algorithm to test Bergeness that does not use any decomposition theorems at all — it tests the graph directly for odd holes and antiholes, rather than testing for the structure enforced by their absence. It could easily have been discovered before the perfect graph conjecture was proved.

Here is an outline of the algorithm. It makes use of "cleaning", a technique first used by Conforti and Rao [3] to recognize linear balanced matrices. With input a graph G, we would like to decide either that G is not Berge, or that G contains no odd hole. (To test Bergeness, we just run this algorithm on G and then again on the complement of G.) If there is an odd hole in G, then there is a shortest one, say C; and C has nicer properties than a general odd hole, that make it more easily detectable. Say a vertex of the remainder of G is *major* for C if its set of neighbours in C do not lie in a 2-edge path of C. The key observation is that, if there are no major vertices for C (that is, if C is "clean"), then there is a polynomial algorithm that will discover an odd hole in G. That is the main content of this paper. (Call this algorithm the "clean shortest hole detector".)

The second half of the algorithm consists of reducing the problem of testing Bergeness to the problem we already solved, that of testing for the presence of a clean shortest odd hole. Let M be the set of all major vertices for a shortest odd hole C. If we knew the set M, we could delete it and run the clean shortest hole detector on what remains of G. Or it would be enough if we knew a set including M and disjoint from C, because again we could delete this set, with the same effect. This is an idea that Conforti (with others) has used in several earlier papers, on similar problems — although we don't know C, and there are at first sight exponentially many possibilities for C, it may be possible to enumerate polynomially many sets $X \subseteq V(G)$, so that if C exists then one of the sets X is guaranteed to include M and be disjoint from C. (If so, we apply the clean shortest hole detector to $G \setminus X$ for each set X in turn.) Enumerating these subsets X is called "cleaning" a shortest odd hole. How can we do that?

We did find a way, but our cleaning step was still very complicated, and at this stage there came another major simplification. G. Cornuéjols, X. Liu and K. Vušković were working independently from us on the same question, and they came very close to solving it; they found their own version of the clean shortest odd hole detector (based on decomposition methods — they present it in [4], in this issue), and they made a great improvement in our cleaning step. The cleaning step itself can be broken into two parts, let us say parts A and B. (Part A is showing that an anticonnected set of major vertices has an edge of common neighbours in the shortest odd hole, and part B is making use of part A.) We had a decent way to do part A, and a long and complex way (about 20 pages) to do part B, which we thought was much harder. Cornuéjols, Liu and Vušković came to the opposite conclusion — they found a 1-page method to do part B, and were having difficulty with part A. It seemed best to combine the methods and make a joint paper on cleaning. So the description of our algorithm to test Bergeness has been split between two papers, this one and [2] (also in this issue). The two together give an algorithm to test Bergeness, with running time $O(|V(G)|^9)$, as we shall see.

Incidentally, there is another interesting open question in this area — is there a polynomial algorithm to test if G contains an odd hole? The half of the algorithm we are going to describe in this paper (the clean shortest hole detector) would work equally well for that question — in other words, it does not terminate on discovering an odd antihole. However, there are places in [2] where we stumble over an odd antihole and stop, declaring the graph non-Berge, and we are currently unable to eliminate that; so the question of testing just for odd holes remains open.

2 Testing for pyramids

Let us be more precise. All graphs in this paper are finite, and simple. The vertex- and edge-sets of a graph G are denoted by $V(G), E(G)$. A subset $X \subseteq V(G)$ is *connected* if the subgraph of G induced on X is connected. If $X \subseteq V(G)$, a *component* of X means a maximal nonnull connected subset of X. A *hole* in G is an induced cycle in G of length ≥ 4, and an *antihole* is an induced subgraph of G whose complement is a hole. A *path* in G is an induced subgraph that is connected, with at least one vertex, no cycle, and no vertex of degree > 2. The *ends* of a path are defined as usual. We denote the set of internal vertices of a path P (that is, its *interior*) by P^*. The *length* of a path or hole is the number of edges in it. If $u, v \in V(G)$, in the same component, we denote the length of the shortest path in G between them by $d_G(u, v)$.

A *triangle* in G means a set of three pairwise adjacent vertices. A *pyramid* in a graph G is an induced subgraph formed by the union of a triangle $\{b_1, b_2, b_3\}$, a fourth vertex a, and three paths P_1, P_2, P_3, satisfying:

- for $i = 1, 2, 3$, P_i is between a and b_i
- for $1 \le i < j \le 3$, a is the only vertex in both P_i, P_j, and $b_i b_j$ is the only edge of G between $V(P_i) \setminus \{a\}$ and $V(P_j) \setminus \{a\}$
- a is adjacent to at most one of b_1, b_2, b_3.

So every pyramid is a subdivision of K_4. Note that the pyramid is determined by a knowledge of P_1, P_2, P_3, and we call it the pyramid *formed* by P_1, P_2, P_3. If there exist a, b_1, b_2, b_3 etc as above, we say that a can be *linked onto* the triangle $\{b_1, b_2, b_3\}$, *via* the paths P_1, P_2, P_3. It is easy to see that any graph containing a pyramid contains an odd hole. Our objective in this section is to give a polynomial algorithm to test whether G contains a pyramid.

If K is a pyramid, formed by three paths P_1, P_2, P_3 linking a onto b_1, b_2, b_3 respectively, we say its *frame* is the 10-tuple

$$a, b_1, b_2, b_3, s_1, s_2, s_3, m_1, m_2, m_3,$$

where

- for $i = 1, 2, 3$, s_i is the neighbour of a in P_i
- for $i = 1, 2, 3$, $m_i \in V(P_i)$ satisfies $d_{P_i}(a, m_i) - d_{P_i}(m_i, b_i) \in \{0, 1\}$.

A pyramid K in G is *optimal* if there is no pyramid K' with $|V(K')| < |V(K)|$. If there is a pyramid in G, then there is an optimal one, and optimal pyramids have some special structure that helps us detect them.

2.1 *Let K be an optimal pyramid, with frame $a, b_1, b_2, b_3, s_1, s_2, s_3, m_1, m_2, m_3$. Let S_1, T_1 be the subpaths of P_1 from m_1 to s_1, b_1 respectively. Let F be the set of all vertices nonadjacent to each of s_2, s_3, b_2, b_3.*

1. *Let Q be a path between s_1 and m_1 with interior in F, and with minimum length over all such paths. Then a-s_1-Q-m_1-T_1-b_1 is a path (say P_1'), and P_1', P_2, P_3 form an optimal pyramid.*
2. *Let Q be a path between m_1 and b_1 with interior in F, and with minimum length over all such paths. Then a-s_1-S_1-m_1-Q-b_1 is a path (say P_1'), and P_1', P_2, P_3 form an optimal pyramid.*

Analogous statements holds for P_2, P_3.

Proof. Note first that from the choice of m_1, we have

$$|E(S_1)| \le |E(T_1)| \le |E(S_1)| + 1.$$

Let us prove the first statement. If $s_1 = m_1$ or s_1, m_1 are adjacent, then the claim holds trivially, so we assume that s_1, m_1 are distinct and nonadjacent. Hence S_1 has length ≥ 2, and therefore T_1 has length ≥ 2, and in particular $m_1 \ne b_1$. From the choice of Q, it follows that $|E(Q)| \le |E(S_1)|$. Let Q have vertices q_1-$\cdots$-q_n, where $q_1 = s_1$ and $q_n = m_1$. Then $n \ge 3$.

Suppose first that none of $q_2, \dots, q_{n-1}$ belong to or have neighbours in $P_2 \cup P_3$. Then there is a path P_1' between a, b_1 with interior in $V(Q \cup T_1)$, of length at most $1 + |E(Q)| + E(T_1)|$. Hence P_1', P_2, P_3 form a pyramid, and from the optimality of K it follows that P_1' has length at least that of P_1. So

$$|E(P_1')| \geq 1 + |E(S_1)| + |E(T_1)| \geq 1 + |E(Q)| + |E(T_1)| \geq |E(P_1')|$$

and therefore equality holds throughout, and in particular, a-s_1-Q-m_1-T_1-b_1 is the path P_1', and the pyramid formed by P_1', P_2, P_3 is optimal. Thus in this case the claim holds.

We may therefore assume that for some k with $2 \leq k \leq n-1$, q_k belongs to or has neighbours in $P_2 \cup P_3$. Choose such a value of k, maximum. We claim that none of $q_k, \dots, q_n$ belongs to $P_2 \cup P_3$; for if $k < n-1$ this follows from the maximality of k, and if $k = n-1$ it follows since $q_n = m_1$ has no neighbour in $P_2 \cup P_3$. From the choice of Q, none of $q_k, \dots, q_{n-1}$ is adjacent to any of b_2, b_3.

Suppose that q_k has neighbours in both P_2^* and P_3^*. Then q_k can be linked onto $\{b_1, b_2, b_3\}$ via a path from q_k to b_1 with interior in $\{q_{k+1}, \dots, q_n\} \cup V(T_1)$, and paths from q_k to b_i with interior in Q_i^* for $i = 2, 3$. Since the first path is strictly shorter than P_1 (since $k \geq 2$ and $|E(Q)| \leq |E(S_1)|$), and the other two are at most as long as P_2, P_3 respectively, this contradicts the optimality of K.

So we may assume that q_k has neighbours in P_2, and none in P_3^*. Let X be the subpath of P_2 between a and the neighbour of q_k that is closest to a in P_2 (say x), and let Y be the minimal subpath of P_2 between b_2 and the neighbour of q_k that is closest to b_2 (say y). Suppose first that $x = y$. Then x can be linked onto $\{b_1, b_2, b_3\}$, via a path from x to b_1 with interior in $\{q_k, \dots, q_n\} \cup V(T_1)$, x-Y-b_2, and x-X-a-P_3-b_3, contrary to the optimality of K (since $k \geq 3$ and $|E(Q)| \leq |E(S_1)|$). So x, y are distinct. If they are nonadjacent, then q_k can be linked onto $\{b_1, b_2, b_3\}$, via a path from q_k to b_1 with interior in $\{q_{k+1}, \dots, q_n\} \cup V(T_1)$, q_k-y-Y-b_2, and q_k-x-X-a-P_3-b_3, again contrary to the optimality of K. So x, y are adjacent. Since s_2 has no neighbour in the interior of Q, it follows that $x \neq a$, and so a can be linked onto $\{q_k, x, y\}$, via a path from a to q_k with interior in $V(S_1) \cup \{q_{k+1}, \dots, q_n\}$, a-X-x and a-P_3-b_3-b_2-Y-y. Since the sum of the lengths of the second two paths equals the sum of the lengths of P_2, P_3, and the first path has length

$$\leq 1 + |E(S_1)| + n - k \leq 2|E(S_1)| \leq |E(S_1)| + |E(T_1)| < |E(P_1)|,$$

this contradicts the optimality of K. This proves the first statement of 2.1.

Now we prove the second statement. If $m_1 = b_1$ or m_1, b_1 are adjacent, the claim holds trivially, so we assume m_1, b_1 are distinct and nonadjacent. Hence T_1 has length ≥ 2, and therefore S_1 has length ≥ 1. From the choice of Q, $|E(Q)| \leq |E(T_1)|$. Let Q have vertices q_1-$\cdots$-q_n, where $q_1 = m_1$ and $q_n = b_1$; then $n \geq 3$, since m_1, b_1 are distinct and nonadjacent. Suppose that none of $q_2, \dots, q_{n-1}$ belong to or have neighbours in $P_2 \cup P_3$. Then from the optimality of K, it follows that P_1 is the shortest path between a, b_1 with interior in $V(P_1) \cup \{q_2, \dots, q_{n-1}\}$; and so Q, T_1 have the same length, and there are no edges between $\{q_2, \dots, q_{n-1}\}$ and $V(S_1 \setminus m_1)$, and therefore the claim holds.

So we may assume that q_k belongs to or has a neighbour in $P_2 \cup P_3$, for some k with $2 \leq k \leq n-1$. Choose k minimum. It follows that none of $q_1, \dots, q_k$ belong to $P_2 \cup P_3$. Let R be a path from q_k to b_1 with interior in $\{q_1, \dots, q_k\} \cup V(T_1)$. We claim that $|E(R)| \leq |E(P_1)| - 2$. This is clear if q_k is adjacent to b_1, since P_1 has length ≥ 4. If q_k is not adjacent to b_1 then

$$k - 1 \leq n - 3 \leq |E(T_1)| - 2 \leq |E(S_1)| - 1,$$

and so

$$|E(R)| \leq k - 1 + |E(T_1)| < |E(S_1)| + |E(T_1)| - 3 = |E(P_1)| - 2.$$

So in either case, $|E(R)| \leq |E(P_1)| - 2$ as claimed.

From the choice of Q, q_k is nonadjacent to b_2, b_3. Suppose first that q_k has neighbours in both P_2^* and P_3^*. Then q_k can be linked onto $\{b_1, b_2, b_3\}$ via R and paths from q_k to b_i with interior in Q_i^* for $i = 2, 3$. Since R is strictly shorter than P_1, and the other two are at most as long as P_2, P_3 respectively, this contradicts the optimality of K.

So we may assume that q_k has neighbours in P_1 and none in P_2^*. Let X be the subpath of P_2 between a and the neighbour of q_k that is closest to a in P_2 (say x), and let Y be the minimal subpath of P_2 between b_2 and the neighbour of q_k that is closest to b_2 (say y). Suppose first that $x = y$. Then x can be linked onto $\{b_1, b_2, b_3\}$, via x-q_k-R-b_1, x-Y-b_2, and x-X-a-P_3-b_3, contrary to the optimality of K (since $|E(R)| \leq |E(P_1)| - 2$). So x, y are distinct. If they are nonadjacent, then q_k can be linked onto $\{b_1, b_2, b_3\}$, via q_k-R-b_1, q_k-y-Y-b_2, and q_k-x-X-a-P_3-b_3, again contrary to the optimality of K. So x, y are adjacent. But then a can be linked onto $\{q_k, x, y\}$, via a path from a to q_k with interior in $\{q_1, \ldots, q_{k-1}\} \cup V(S_1)$, a-X-x and a-P_3-b_3-b_2-Y-y. Since the first path is strictly shorter than P_1, and the sum of the lengths of the second two paths equals the sum of the lengths of P_2, P_3, again this contradicts the optimality of K. This proves the second statement of 2.1. ∎

We use the previous lemma to prove the following.

2.2 *There is an algorithm with the following specifications:*

- **Input** *A graph G.*
- **Output** *Either it finds a pyramid (and hence an odd hole) in G, or it determines that G contains no pyramid.*
- **Running time** $O(|V(G)|^9)$.

Proof. Here is an algorithm. Enumerate all 6-tuples $b_1, b_2, b_3, s_1, s_2, s_3$ that satisfy the following conditions:

- for $1 \leq i < j \leq 3$, $\{b_i, s_i\}$ is disjoint from $\{b_j, s_j\}$, and $b_i b_j$ is the unique edge between them
- there is a vertex a adjacent to all of s_1, s_2, s_3 and to at most one of b_1, b_2, b_3, such that for $1 \leq i \leq 3$, if a is adjacent to b_i then $s_i = b_i$ (let us call such a vertex a an *apex* for the 6-tuple).

(We can find all such 6-tuples in time $O(|V(G)|^7)$.)

For each such choice of $b_1, b_2, b_3, s_1, s_2, s_3$ we do the following. Let $M = V(G) \setminus \{b_1, b_2, b_3, s_1, s_2, s_3\}$. For each $m \in M$, find a shortest path $S_1(m)$ between s_1, m such that s_2, s_3, b_2, b_3 have no neighbours in its interior, if such a path exists. Find a shortest path $T_1(m)$ between m, b_1 such that s_2, s_3, b_2, b_3 have no neighbours in its interior, if such a path exists. Find $S_2(m), T_2(m), S_3(m), T_3(m)$ similarly. (To find these paths, for a given 6-tuple $b_1, b_2, b_3, s_1, s_2, s_3$, but for all m, takes time $O(|V(G)|^2)$.)

Next, for each $m \in M \cup \{b_1\}\}$ let $P_1(m)$ be defined as follows. If $s_1 = b_1$ let $P_1(b_1)$ be the one-vertex path with vertex b_1, and let $P_1(m)$ be undefined for all other vertices for each $m \in M$. Now assume that $s_1 \neq b_1$. Then $P_1(m)$ is undefined; and for each $m \in M$, test whether all the following are true:

- m is nonadjacent to b_2, b_3, s_2, s_3
- $S_1(m), T_1(m)$ both exist

- $V(S_1(m) \cap T_1(m)) = \{m\}$
- there are no edges between $V(S_1(m) \setminus m)$ and $V(T_1(m) \setminus m)$.

If so, then s_1-$S_1(m)$-m_1-$T_1(m)$-b_1 is a path; call it $P_1(m)$ (and otherwise $P_1(m)$ is undefined). Define $P_2(m), P_3(m)$ similarly. (Finding $P_1(m), P_2(m), P_3(m)$ for all m takes time $O(|V(G)|^3)$.)

Our goal is now to check for each triple m_1, m_2, m_3 whether the three paths $P_i(m_i)$ $(i = 1, 2, 3)$ form a pyramid, for some suitable choice of a vertex a. But with care we can significantly reduce the running time, so let us do it carefully.

If $1 \leq i < j \leq 3$, we say that (m_i, m_j) is a *good* (i, j)-pair if $m_i \in M \cup \{b_i\}$, $m_j \in M \cup \{b_j\}$, $P_i(m_i), P_j(m_j)$ both exist, and the sets $V(P_i(m_i)), V(P_j(m_j))$ are disjoint and there are no edges between them. Next we want to find the list of all good $(1, 2)$-pairs. (And we need to do it in time $O(|V(G)|^3)$, so the obvious method is not fast enough.) For each $m_1 \in M \cup \{b_1\}$, we find the set of all m_2 such that (m_1, m_2) is good, in two stages as follows. If $P_1(m_1)$ does not exists, there are no such good pairs. If it exists, colour black the vertices of M that either belong $P_1(m_1)$ or have a neighbour in $P_1(m_1)$, and colour all other vertices white. (This takes time $O(|V(G)|^2)$.) Then for each $m_2 \in M \cup \{b_2\}$, test whether $P_2(m_2)$ exists and contains no black vertices. (For each m_2 this takes linear time, so doing it for all m_2 takes takes time $O(|V(G)|^2)$.) Repeating this for all m_1, we compute the set of all good $(1, 2)$-pairs (in time $O(|V(G)|^3)$). Repeat to find the good $(1, 3)$-pairs and $(2, 3)$-pairs. Now we examine all triples m_1, m_2, m_3 such that $m_i \in M_i \cup \{b_i\}$ for $i = 1, 2, 3$ and test whether (m_i, m_j) is a good (i, j) pair for $1 \leq i < j \leq 3$. (For each triple this takes constant time, so altogether it again takes time $O(|V(G)|^3)$.) If we find a triple such that all three pairs are good, we output that G contains a pyramid and stop.

After examining all choices of $b_1, b_2, b_3, s_1, s_2, s_3$, output that G contains no pyramid. (Each choice of $b_1, b_2, b_3, s_1, s_2, s_3$ takes total time $O(|V(G)|^3)$ to process and since there are $O(|V(G)|^6)$ such choices, the total running time is $O(|V(G)|^9)$.)

Now we need to prove that the algorithm works correctly. Suppose first that it outputs that G contains a pyramid. Therefore for some choice of $b_1, b_2, b_3, s_1, s_2, s_3$, we know the following:

- for $1 \leq i < j \leq 3$, $\{b_i, s_i\}$ is disjoint from $\{b_j, s_j\}$, and $b_i b_j$ is the unique edges between them, and in particular, $\{b_1, b_2, b_3\}$ is a triangle
- there is a vertex a adjacent to all of s_1, s_2, s_3 and to at most one of b_1, b_2, b_3, such that for $1 \leq i \leq 3$, if a is adjacent to b_i then $s_i = b_i$,
- with notation as before, for $i = 1, 2, 3$ there exists $m_i \in M \cup \{b_i\}$, such that (m_i, m_j) is a good (i, j) pair for $1 \leq i < j \leq 3$.

Then for $i = 1, 2, 3$, $P_i(m_i)$ is a path between s_i and b_i, these three paths are vertex-disjoint, and the only edges between them join two of $\{b_1, b_2, b_3\}$. Moreover there is a vertex a, adjacent to at most one of b_1, b_2, b_3, and with a neighbour in each $P_i(m_i)$. Hence a can be linked onto the triangle $\{b_1, b_2, b_3\}$, via paths with interior in $V(P_i(m_i))(i = 1, 2, 3)$, and therefore G contains a pyramid. So in this case the output of the algorithm was correct.

For the converse, suppose that G in fact does contain a pyramid, and let an optimal pyramid in G be formed by P_1, P_2, P_3, with frame

$$a, b_1, b_2, b_3, s_1, s_2, s_3, m_1, m_2, m_3.$$

Let us examine what the algorithm does when it inspects the 6-tuple $b_1, b_2, b_3, s_1, s_2, s_3$. Since a is an apex for this 6-tuple, the algorithm will proceed to search for paths $S_i(m_i), T_i(m_i)$ for $1 \leq i \leq 3$, and since such paths exist, it will find them. By six applications of 2.1, it follows that the union of these six paths (together with the vertex a, the edges as_i, and the edges b_ib_j) is an optimal pyramid; and so the algorithm will detect this pyramid, and output correctly that G contains a pyramid. This proves 2.2. ▮

Since this is the slowest part (or one of them) of the algorithm to test Bergeness, it would be nice to make it faster, but at the moment we don't see how. Here are three encouraging observations, each of which looks at first sight as if it brings the running time down to $O(|V(G)|^8)$:

- Let P_1, P_2, P_3 be the optimal pyramid, and let P_1 be the shortest of the three paths, with second vertex s_1. Then any minimum length path between s_1 and b_1 containing no neighbours of b_2, b_3 can be used in place of $P_1 \setminus s_1$; in other words we don't need to "guess" the middle vertex m_1 before we can find the path
- for any one of the three paths, say P_3, if we have figured out P_1, P_2 correctly, we don't need the best possible P_3; to be sure that G contains a pyramid, it is enough to detect any path from a to b_3 that does not pass through vertices with neighbours in P_1, P_2
- we don't really need to find a pyramid; it is enough to find an odd hole. And if there is an optimal pyramid, there is an odd hole formed by the union of two of the paths $P_i(m_i)$ ($i = 1, 2, 3$). So it is enough to examine each possible pair of paths, rather than trying to examine triples of paths.

But none of these is enough, as far as we can see; the running time remains $O(|V(G)|^9)$.

3 Finding jewels

Before we turn to detecting clean shortest odd holes, there is another configuration we need to eliminate. We say a sequence $v_1, \ldots, v_5, P$ is a *jewel* in G if $v_1, \ldots, v_5$ are distinct vertices, $v_1v_2, v_2v_3, v_3v_4, v_4v_5, v_5v_1$ are edges, v_1v_3, v_2v_4, v_1v_4 are nonedges, and P is a path of G between v_1, v_4 such that v_2, v_3, v_5 have no neighbours in P^*.

3.1 *There is an algorithm with the following specifications:*

- **Input** *A graph G.*
- **Output** *Decides whether there is a jewel in G.*
- **Running time** $O(|V(G)|^6)$.

Proof. The obvious implementation (enumerate all choices of $v_1, \ldots, v_5$ and check them) has running time $O(|V(G)|^7)$, but we can gain a little bit as follows. Enumerate all 3-tuples v_2, v_3, v_5 of distinct vertices such that v_2v_3 is an edge. For each choice of v_2, v_3, v_5, find the set F of all vertices nonadjacent to each of v_2, v_3, v_5, and find all its components. Find the set X_1 of all vertices adjacent to v_2, v_5 and not to v_3, and for each $v_1 \in X_1$ and each component of F' of F, record whether v has a neighbour in F'. Do the same for the set X_2 of all vertices adjacent to v_3, v_5 and not to v_2. Then test if there exist $v_1 \in X_1$, $v_2 \in X_2$, and a component F' of F, so that v_1, v_2 are nonadjacent and both have neighbours in F'. If so then output that G contains a jewel. If after examining all choices we still have not found a jewel, then none exists; output that fact. This proves 3.1. ∎

We observe also that:

3.2 *If there is a jewel in G, then there is an odd hole in G.*

Proof. Let $v_1, \dots, v_5, P$ be a jewel. If P has odd length, then v_1-P-v_4-v_5-v_1 is an odd hole, and otherwise v_1-P-v_4-v_3-v_2-v_1 is an odd hole. This proves 3.2. ∎

4 The clean shortest odd hole detector

Let C be a shortest odd hole in G. If $x \in V(G)$, we say x is *major* (for C) if the set of its neighbours in C is not contained in a 2-edge path of C (and therefore $x \notin V(C)$). We say C is *clean* if no vertex is major for C. In this section we give a polynomial algorithm that, with input a graph G containing no jewels and pyramids, either finds an odd hole in G, or concludes that there is no clean shortest odd hole in G.

We recall that $d_G(u, v)$ is the length of the shortest path in G between u, v.

4.1 *Let G be a graph containing no jewel or pyramid, and let C be a shortest odd hole in G. Assume that C is clean. Let $u, v \in V(C)$ be distinct and nonadjacent; let L_1, L_2 be the two subpaths of C joining u, v, where $|E(L_1)| \le |E(L_2)|$; and let P be any path in G between u, v with length $d_G(u, v)$. Then*

- *P, L_1 have the same length, and so $d_G(u, v) = d_C(u, v)$*
- *P^* is disjoint from L_2^*, and there are no edges between them, and so u-P-v-L_2-u us a hole C' say*
- *C' is a shortest odd hole in G, and it is clean.*

Proof. We proceed by induction on the length of P. Let P have vertices u-p_1-$\cdots$-p_k-v, where $k \ge 1$, so its length is $k + 1$. For fixed k we proceed by induction on $|E(L_2)|$. Assign C an orientation, clockwise say, and for any two distinct vertices x, y in C, let $C(x, y)$ be the clockwise path in C from x to y, when it exists (that is, unless y immediately precedes x in the clockwise order). We may assume that $L_1 = C(u, v)$. Let C have vertices c_1-$\cdots$-c_{2n+1} in clockwise order, where $c_1 = u$ and $c_m = v$; and therefore $m \le n + 1$. From the hypothesis,

$$k + 1 = d_G(u, v) \le d_C(u, v) = m - 1 \le n.$$

(1) *The sets $P^*, C(v, u)^*$ are disjoint, and there are no edges between $\{p_2, \dots, p_{k-1}\}$ and $C(v, u)^*$.*

For otherwise, for some j with $m + 1 \le j \le 2n + 1$, there exist paths P_1, P_2 from c_j to u, v respectively, with interior in P^*, both strictly shorter than P. Now

$$|E(P_1)| + |E(P_2)| \le k + 3 \le m + 1 \le 2n + 3 - m < (2n + 3 - j) + (j + 1 - m),$$

and so either P_1 has length at most that of $C(c_j, u)$, or P_2 has length at most that of $C(v, c_j)$. From the symmetry between u, v we may assume the first. From the inductive hypothesis, it follows that

$$d_G(c_j, u) = d_C(c_j, u) = min(2n + 2 - j, j - 1).$$

But

$$d_G(c_j, u) \le |E(P_1)| < |E(P)| \le m - 1 \le j - 1,$$

so $d_G(c_j, u) = 2n + 2 - j$. Since P_1 has length at most $2n + 2 - j$, it follows that P_1 is a shortest path in G between c_j, u, and therefore, from the inductive hypothesis, u-$C(u, c_j)$-c_j-P_1-u is a clean shortest odd hole in G, say C'. Orient C' so that the orientations of C, C' agree on the common subpath $C(u, v)$. Let u' be the neighbour of c_j in P_2; then there is a subpath P' of P between u', v, of length $k + 2 - |E(P_1)| = k + j - 2n$. Since P' is strictly shorter than P, it follows from the inductive hypothesis applied to C', P' that one of the paths $C'(u', v), C'(v, u')$ has length at most $k + j - 2n$, and in particular is strictly shorter than P. Since $C'(u', v)$ includes $C(u, v)$ and therefore is not strictly shorter than P, it follows that $C'(v, u')$ has length at most $k + j - 2n$. Its length is $j - m + 1$, and so $k + m \ge 2n + 1$. But $k + 2, m \le n + 1$, a contradiction. This proves (1).

(2) $k \ge 2$.

For assume that $k = 1$. Since p_1 is not major, it follows that $m = 3$, and the only neighbours of p_1 in C are c_1, c_3 and possibly c_2. In particular, c_1-p_1-c_3-$\cdots$-c_{2n+1}-c_1 is a hole C'. Since it has the same length as C, we deduce that C' is a shortest odd hole. It remains to prove that C' is clean. For suppose there is a vertex w that is major for C'. Since w is not major for C, it follows that $C' \ne C$, and so $p_1 \notin V(C)$; and for the same reason, w is adjacent to p_1. The neighbours of w in C do not all lie in the path c_2-c_3-c_4, since w is major for C'; and similarly they do not all lie in c_{2n+1}-c_1-c_2, and not all in c_2-c_2-c_4. Since they do all lie in some 2-edge path of C, we may assume from the symmetry that w is nonadjacent to both c_1, c_2. Choose i, j with $3 \le i \le j \le 2n+1$, minimum and maximum respectively so that w is adjacent to c_i, c_j. Hence $j \le i+2$, and $j \ge 5$. The hole w-c_j-$\cdots$-c_{2n+1}-c_1-p_1-w is shorter than C and therefore even, and so j is odd. If $i > 3$ then similarly i is even, and since $j - i \le 2$ it follows that $j = i + 1$; but then the paths p_1-w, p_1-c_3-$\cdots$-c_i, p_1-c_1-c_{2n+1}-$\cdots$-c_{i+1} form a pyramid in G, a contradiction. So $i = 3$, and therefore $j = 5$. But then c_1, c_2, c_3, w, p_1 and the path w-c_5-$\cdots$-c_{2n+1}-c_1 form a jewel, a contradiction. This proves (2).

From (2) it follows that $m \ge 4$, and therefore $n \ge 3$.

(3) *Either c_1 is the only neighbour of p_1 in C, or c_1, c_2 are the only neighbours of p_1 in C, or $m = n + 1$ and c_1, c_{2n+1} are the only neighbours of p_1 in C; and in particular, $p_1 \notin V(C)$. The analogous statement also holds for p_k.*

For suppose first that p_1 has two nonadjacent neighbours $x, y \in V(C)$. Since p_1 is not major, we may assume that $C(x, y)$ has length 2 and contains all neighbours of p_1 in C (and, if $p_1 \in V(C)$, it is the middle vertex of $C(x, y)$). In particular, p_1-x-$C(x, y)$-y-p_1 is a hole C' of the same length as C, and hence is a shortest odd hole, and it is clean, by (2) and the inductive hypothesis. Since $d_G(p_1, v) = k$, it follows from the inductive hypothesis that $d_{C'}(p_1, v) = k$. But $d_{C'}(p_1, v) = d_C(p_1, v) = m - 1 \ge k + 1$, a contradiction. This proves that p_1 does not have two nonadjacent neighbours in C, and in particular, $p_1 \notin V(C)$. Since p_1 is adjacent to c_1, we may assume it is also adjacent to c_{2n+1}, for otherwise the claim holds. Suppose that $m \le n$. Then from the second inductive hypothesis (on $|E(L_2)|$) applied to the path c_{2n+1}-p_1-$\cdots$-p_k-c_m, it follows that

$$d_C(c_{2n+1}, c_m) = d_G(c_{2n+1}, c_m) \le k+1.$$

But $d_C(c_{2n+1}, c_m) = m \ge k+2$, a contradiction. Hence $m = n+1$. This proves (3).

(4) *If $P, C(u,v)$ have the same length then there are no edges between P^* and $C(v,u)^*$.*

For suppose there are edges between P^* and $C(v,u)^*$. From (1) and (3), we may assume that p_1 is adjacent to c_{2n+1} and $m = n+1$. Let P' be the path c_{2n+1}-p_1-$\cdots$-p_k-v. If $d_G(c_{2n+1}, v) < k+1$, then from the inductive hypothesis, $d_C(c_{2n+1}, v) < k+1 \le n$, a contradiction since $m = n+1$. So $d_G(c_{2n+1}, v) \ge k+2$, and therefore P' is a shortest path in G between c_{2n+1}, v. Hence there is symmetry between c_1, c_{2n+1}, and from (1) applied under this symmetry we deduce that P^* is disjoint from $C(u,v)^*$, and there are no edges between $\{p_2, \dots, p_{k-1}\}$ and $C(u,v)^*$. Consequently there are no edges between P^* and C except for p_1c_1, p_1c_{2n+1} and possibly edges incident with p_k. Since G contains no pyramid, c_m cannot be linked onto the triangle $\{p_1, c_1, c_{2n+1}\}$, and therefore p_k has at least two neighbours in C; and so (by (3)) it has exactly two, and they are adjacent, and from the symmetry between c_1, c_{2n+1} we may assume that the second neighbour of p_k is p_{m-1}. If $k+m$ is even then c_1-$C(c_1, c_{m-1})$-p_k-$\cdots$-p_1-c_1 is an odd hole of length $k+m-1 < 2n+1$, while if $k+m$ is odd then c_m-$C(c_m, c_{2n+1})$-c_{2n+1}-p_1-$\cdots$-p_k-c_m is an odd hole of length $2n-m+k+2 < 2n+1$, again a contradiction. This proves that there are no edges between P^* and $C(v,u)^*$, and therefore the theorem holds. This proves (4).

(5) *If $P, C(u,v)$ have the same length then the theorem holds.*

For by (4), c_1-p_1-$\cdots$-p_k-c_m-c_{m+1}-$\cdots$-c_{2n+1}-c_1 is a hole, C' say; and it is a shortest odd hole, since $P, C(u,v)$ have the same length. It remains to prove that C' is clean. Suppose not, and let $w \in V(G)$ be major for C'. Since it is not major for C, it is adjacent to at least one of $p_1, \dots, p_k$. Since P is a shortest path between u, v, the neighbours of w in P lie in a 2-edge subpath of P; and since w is major for C', it follows that w is adjacent to at least one of $c_{m+1}, \dots, c_{2n+1}$.

Suppose that w has two nonadjacent neighbours in the path $C(v,u)$. Since w is not major for C, we may assume that the neighbours of w in C are c_i, c_{i+2}, and possibly c_{i+1} where $m \le i \le 2n-1$. By (2) and the inductive hypothesis, the hole obtained from C by replacing c_{i+1} by w is clean; but there are edges between w and $p_1, \dots, p_k$, contradicting (4) applied to this hole. So w does not have two nonadjacent neighbours in $C(v,u)$.

Suppose that w is adjacent to none of $c_{m+2}, \dots, c_{2n}$. Then we may assume w is adjacent to c_{2n+1}. Since it is not major for C, it has no neighbours in C except c_{2n+1} and possibly $c_1, c_2, c_{2n}, c_{2n-1}$; and it is not adjacent to c_{2n} since $2n \ge m+2$ (because $n \ge 3$), and it is not adjacent to c_{2n-1} since it does not have two nonadjacent neighbours in $C(v,u)$. Choose i with $1 \le i \le k$ maximum such that w is adjacent to p_i. Since w is major for C' it follows that $i > 1$; and since $c_{2n+1}, c_1, p_1, p_2, w$ and the path p_2-$\cdots$-p_k-c_m-$\cdots$-c_{2n+1} do not form a jewel, it follows that $i > 2$. Hence $d_G(c_m, c_{2n+1}) \le k$, and so from the first inductive hypothesis, $d_G(c_m, c_{2n+1}) = d_C(c_m, c_{2n+1})$. But

$$d_C(c_m, c_{2n+1}) = min(m, 2n+1-m) > k,$$

a contradiction. So w is adjacent to one of $c_{m+2},\dots,c_{2n}$. Since it is not major for C, it is nonadjacent to all of $c_2,\dots,c_{m-1}$. Since it does not have two nonadjacent neighbours in $C(v,u)$, it is nonadjacent to both c_1,c_m.

Choose i,j with $1\le i\le j\le k$, minimum and maximum respectively so that w is adjacent to p_i,p_j. Choose s,t with $m\le s\le t\le 2n+1$, minimum and maximum respectively so that w is adjacent to c_s,c_t. Hence $s>m$ and $t\le 2n$. Since P is a shortest path between u,v it follows that $j-i\le 2$. Since w does not have two nonadjacent neighbours in $C(v,u)$, it follows that $t-s\le 1$. Since

$$w\text{-}c_t\text{-}\cdots\text{-}c_{2n+1}\text{-}c_1\text{-}p_1\text{-}\cdots\text{-}p_i\text{-}w$$

is a hole of length $\le 2n$, it is even, and so $t+i$ is even; and similarly $s-m+k-j$ is odd. Since $k=m-2$ it follows that $s+j$ is odd. Consequently $(t-s)+(j-i)$ is odd.

Suppose that $i=j$. Then $t-s=1$ since $t-s$ is odd; but then p_i can be linked onto the triangle $\{w,c_s,c_t\}$ via p_i-w and two subpaths of C', contradicting that G contains no pyramid. So $j>i$. Similarly if $s=t$ then $j-i=1$, and c_s can be linked onto $\{w,p_i,p_j\}$, again a contradiction. So $t>s$. Since $t-s\le 1$ it follow that $t=s+1$ and therefore $j=i+2$. But then the path $u\text{-}p_1\text{-}\cdots\text{-}p_i\text{-}w\text{-}p_j\text{-}\cdots\text{-}p_k\text{-}v$ is a path between u,v with the same length as P, and yet there are edges between its interior and $C(v,u)^*$, contrary to (4). This proves (5).

In view of (5) we assume henceforth that $k+2<m$.

(6) *The sets $P^*,C(u,v)^*$ are disjoint, and there are no edges between $\{p_2,\dots,p_{k-1}\}$ and $C(u,v)$.*

The argument is almost identical with that for (1). Suppose the claim is false; then there exists j with $2\le j\le m-1$, and paths P_1,P_2 from c_j to u,v respectively, with interior in P^*, and both strictly shorter than P. Since

$$|E(P_1)|+|E(P_2)|\le k+3\le m<j+(m-j+1)$$

it follows that either P_1 has length $<j$ or P_2 has length $<m-j+1$, and from the symmetry we may assume the first. By the inductive hypothesis it follows that $d_G(u,c_j)=j-1$, and therefore P_1 is a shortest path between u,c_j, and it has length $j-1$ and $u\text{-}P_1\text{-}c_j\text{-}C(c_j,u)\text{-}u$ is a clean shortest odd hole, C' say. Orient C' to agree with the orientation of C on $C(v,u)$. Let u' be the neighbour of c_j in P_1, and let P' be the subpath of P between u',v. Thus P' has length

$$|E(P)|-(|E(P_1)|-1)=(k+1)-(j-2)=k-j+3.$$

From the inductive hypothesis applied to C',P', it follows that $d_{C'}(u',v)=k-j+3$. But $C'(u',v)$ has length $m-j+1>k-j+3$, and $C'(v,u')$ has length strictly longer than P, a contradiction. This proves (6).

Note that c_{2n+1},c_1,c_2 are all different from c_{m-1},c_m,c_{m+1}, since $k\ge 2$. From (1),(3) and (6) it follows that the only edges between P^* and $V(C)$ are p_1c_1, p_kc_m, possibly one edge from p_1 to one of c_2,c_{2n+1}, and possibly one edge from p_k to one of c_{m-1},c_{m+1}. If neither or both of the possible extra edges are present, there is an odd hole shorter than C, a contradiction; while if exactly one of the possible extra edges is present, then G contains a pyramid (induced on $V(C\cup P)$), again a contradiction. This proves 4.1. ∎

We use the previous result to prove the following, the main result of this section.

4.2 *There is an algorithm with the following specifications:*

- **Input** *A graph G containing no pyramid or jewel.*
- **Output** *Determines one of the following:*
  1. *G contains an odd hole*
  2. *there is no clean shortest odd hole in G.*
- **Running time** $O(|V(G)|^4)$.

Proof. Here is an algorithm. For every pair of vertices, u, v, find a shortest path $P(u, v)$ between them, if one exists. For every triple u, v, w, test whether the three paths $P(u, v), P(v, w), P(w, u)$ all exist, and if so whether their union is an odd hole. If we find such a hole, output that fact. If not, when all triples have been examined, output that there is no clean shortest odd hole in G.

To see that this works correctly, certainly if the algorithm outputs statement 1 then that is correct. We must check that if statement 2 is false then the algorithm will output statement 1. So assume that statement 2 is false, that is, there is a shortest odd hole C of G that is clean. Choose vertices $u, v, w \in V(C)$, roughly equally spaced in C; more precisely, such that every component of $C \setminus \{u, v, w\}$ contains at most $n-1$ vertices, where C has length $2n+1$. Since there is a path joining u, v, the algorithm will find a shortest such path $P(u, v)$. We claim that C can be chosen containing $P(u, v)$. For let L_1 be the path of C joining u, v, not passing through w. Then L_1 has length $\leq n$, from the choice of u, v, w, and so from 4.1, $L_1, P(u, v)$ have the same length. If u, v are adjacent then $P(u, v) = L_1$ and therefore C already contains $P(u, v)$; and otherwise let L_2 be the second path between u, v in C. The union of $L_2, P(u, v)$ is a clean shortest odd hole, by 4.1, and so again we may choose C containing $P(u, v)$. By repeating this for the other two pairs from u, v, w, we see that C can be chosen to include any of $P(u, v), P(v, w), P(w, u)$; and indeed all of them simultaneously. So the union of the three paths joining u, v, w chosen by the algorithm is an odd hole, and therefore in this case the algorithm correctly outputs an odd hole. The running time iof the algorithm as described is $O(|V(G)|^5)$, because after selecting u, v, w and the three paths, it takes quadratic time to check whether the three paths make a hole. With a little more care we can bring it down to $O(|V(G)|^4)$, using the black/white colouring trick we used in 2.2. But here the running time is not crucial, so we omit the details. This proves 4.2. ∎

5 An algorithm for Bergeness

If C is an odd hole in G, a subset $X \subseteq V(G)$ is a *cleaner* for C if X contains all vertices that are major for C, and $X \cap V(C)$ is a subset of a 2-edge path of C. In [2] we show the following.

5.1 *There is an algorithm with the following specifications:*

- **Input** *A graph G.*
- **Output** *Either*

1. *$O(|V(G)|^5)$ subsets, such that if C is a shortest odd hole in G, and every major vertex has at least four neighbours in C, then one of the subsets is a cleaner for C, or*
2. *it determines that G is not Berge.*

- **Running time** $(|V(G)|^6)$.

Let us put these pieces together.

5.2 *There is an algorithm with the following specifications:*

- **Input** *A graph G.*
- **Output** *Determines whether G is Berge.*
- **Running time** $O(|V(G)|^9)$.

Proof. Here is a wasteful way to do it. First we use 2.2 and 3.1 to check there is no pyramid or jewel (if we find a pyramid or jewel we output that G is not Berge). Then we run 5.1, and we may assume we obtain the $O(|V(G)|^5)$ subsets (for otherwise we again output that G is not Berge). (Note that the hypothesis of that algorithm that every major vertex has at least four neighbours in C holds for every shortest odd hole C, since G contains no pyramid.) For each subset X in turn, enumerate all $Y \subseteq X$ with $|Y| \leq 3$; and apply 4.2 to each graph $G \setminus (X \setminus Y)$. If G has an odd hole, then one of these subgraphs has a clean shortest odd hole, and we will therefore detect it. If this does not find an odd hole, then we run the whole thing over again on the complement. If we also find no odd hole in the complement, then G is Berge and we output that.

This is simple to state, but the running time is $O(|V(G)|^{12})$, and with more care we can do it more efficiently. So let us do it again.

As before, first we use 2.2 and 3.1 to check there is no pyramid or jewel. (Running 2.2 takes time $O(|V(G)|^9)$; all the remainder will also take time $O(|V(G)|^9)$, as we shall see.) Then we run 5.1, and we may assume we obtain the $O(|V(G)|^5)$ subsets. We will examine each such subset X in turn. For a given X, we do the following.

For every pair $x, y \in V(G)$ of distinct vertices, find a shortest path $P(x, y)$ between x, y with no internal vertex in X, if there is one, and let $r(x, y)$ be its length. For all $y_1 \in V(G) \setminus X$ and all 2-edge paths x_1-x_3-x_2 of $G \setminus y_1$, we check whether all the following are true, where y_2 is the neighbour of y_1 in $P(x_2, y_1)$:

- $r(x_1, y_1)$, $r(x_2, y_1)$ both exist (and therefore so does y_2)
- $r(x_2, y_1) = r(x_1, y_1) + 1 = r(x_1, y_2)$ ($= n$ say)
- $r(x_3, y_1), r(x_3, y_2) \geq n$.

If we find such a choice of x_1, x_2, x_3, y, then we output that there is an odd hole. When we have tried all the subsets X, if we have not found such a choice of x_1, x_2, x_3, y, we repeat the same procedure on the complement of G. If again we find no choice of X, x_1, x_2, x_3, y, we output that G is Berge.

Let us see that the output of this algorithm is correct. First, suppose that for some X there is a choice of $x_1, x_2, x_3, y_1, y_2, n$ as above, satisfying the three conditions. We claim that G contains an odd hole. Let $P(x_1, y_1)$ have vertices p_1-$\cdots$-p_n, and let $P(x_2, y_1)$ have vertices q_1-$\cdots$-q_{n+1}, where $p_1 = x_1, p_n = q_{n+1} = y_1$, $q_1 = x_2$, and $q_n = y_2$.

Since $r(x_3, y_1), r(x_3, y_2) \geq n$, it follows that x_3 does not belong to $P(x_1, y_1)$ and $P(x_2, y_1)$, and has no neighbours in $P(x_1, y_1) \setminus x_1$, and none in $P(x_2, y_1) \setminus x_2$. We claim first that $p_2, \ldots, p_{n-1}$ are all different from $q_2, \ldots, q_n$. For suppose that $p_i = q_j$ say, where $2 \leq i \leq n-1$ and $2 \leq j \leq n$. Then the subpaths of these two paths between p_i, y_1 are both subpaths of shortest paths, and therefore have the same length, that is, $j = i+1$. So p_1-$\cdots$-p_i-q_{j+1}-$\cdots$-q_n contains a path between x_1, y_2 of length $\leq n-2$, contradicting that $r(x_1, y_2) = n$. So $P(x_1, y_1)$ and $P(x_2, y_1)$ have no common vertex except y_1. If there are no edges between $P(x_1, y_1) \setminus y_1$ and $P(x_2, y_1) \setminus y_1$, then the union of these two paths and the path x_1-x_3-x_2 is an odd hole, and so the algorithm has performed correctly. So assume that $p_i q_j$ is an edge where $1 \leq i \leq n-1$ and $1 \leq j \leq n$. There is a path between x_1, y_2 within $\{p_1, \ldots, p_i, q_j, \ldots, q_n\}$, which therefore has length $\leq n - j + i$; and since $r(x_1, y_2) = n$, it follows that $n - j + i \geq n$, that is, $i \geq j$. But also $d(x_2, y_1) \geq n$, and so $j + n - i \geq n$, that is, $j \geq i$. Consequently $i = j$; let us choose i minimum. Then x_3-p_1-$\cdots$-p_i-q_i-$\cdots$-q_1-x_3 is an odd hole, and again the algorithm has answered correctly. So when the algorithm outputs that G is not Berge, it is true that G is not Berge.

Now we must prove the converse. By taking complements we may assume that there is an odd hole in G, and let C be a shortest odd hole. Since one of the sets output by the call to 5.1 was a cleaner for C, we may assume that for some choice of X and x_1, x_2, x_3, y_1 above, all these four vertices lie in C and $d_C(x_2, y) = d_C(x_1, y) + 1$ ($= n$ say), and C is a clean shortest odd hole in H, where $H = G \setminus (X \setminus \{x_1, x_2, x_3\})$. It follows that $d_C(x_1, y_2) = n$, and $d_C(x_3, y_1), d_C(x_3, y_2) \geq n$. We need to show the same statements with d_C replaced by r. By 4.1, for any two vertices $u, v \in V(C)$, $d_H(u, v) = d_C(u, v)$. Since

$$r(x_1, y_1) \leq d_C(x_1, y_1) = n - 1 = d_H(x_1, y_1) \leq r(x_1, y_1)$$

we have equality throughout, and so $r(x_1, y_1) = n - 1$, and similarly $r(x_2, y_1) + 1 = r(x_1, y_2) = n$; and since

$$n = d_C(x_3, y_1) = d_H(x_3, y_1) \leq r(x_3, y_1),$$

it follows that $r(x_3, y_1) \geq n$ and similarly $r(x_3, y_2) \geq n$. Thus in this case the algorithm would output correctly that G is not Berge, and so the output of the algorithm is correct in all cases.

The two calls to 2.2 (in G and in its complement) take time $O(|V(G)|^9)$. For the remainder, the calls to 3.1 and 5.1 take time $O(|V(G)|^6)$, and then we examine $O(|V(G)|^5)$ cleaner candidates. For each, it takes time $O(|V(G)|^3)$ to find all the shortest paths; then we check all quadruples x_1, x_2, x_3, y, and each takes constant time. So the time taken here is $O(|V(G)|^9)$, as claimed. This proves 5.2. ∎

6 Acknowledgements

We would like to express our thanks to Neil Robertson and Robin Thomas. Many of the ideas and methods, both in this paper and in the part that we contributed to [2], grew from joint work with them. In particular, Robin correctly insisted on trying to clean shortest odd holes, despite our resistance; and they did a lot of work with us on finding common neighbours of anticonnected sets of major vertices. Due to physical distance, they were not working with us when we finally figured out the algorithm, but they contributed a great deal to our understanding.

Thanks also to the American Institute of Mathematics for their support. They hosted a conference on perfect graphs devoted in large part to decomposition-based cleaning methods, just a week before we found the algorithm, and that was a great stimulus to our work.

References

[1] M. Chudnovsky, N. Robertson, P. Seymour and R. Thomas, "The strong perfect graph theorem", manuscript, June 2002.

[2] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour and K. Vušković, "Cleaning for Bergeness", manuscript, November 2002.

[3] M. Conforti and M.R. Rao, "Structural properties and decomposition of linear balanced matrices", *Math. Programming* 55 (1992), 129-168.

[4] G. Cornuéjols, X. Liu and K. Vušković, in preparation.