Finding and counting given length cycles *

Noga Alon † *Raphael Yuster* † *Uri Zwick* †

February 28, 2012

Abstract

We present an assortment of methods for finding and counting simple cycles of a given length in directed and undirected graphs. Most of the bounds obtained depend solely on the number of edges in the graph in question, and not on the number of vertices. The bounds obtained improve upon various previously known results.

1 Introduction

The main contribution of this paper is a collection of new bounds on the complexity of finding simple cycles of length exactly k, where $k \geq 3$ is a fixed integer, in a directed or an undirected graph $G = (V, E)$. These bounds are of the form $O(E^{\alpha_k})$ or of the form $O(E^{\beta_k} \cdot d(G)^{\gamma_k})$, where $d(G)$ is the *degeneracy* of the graph (see below). The bounds improve upon previously known bounds when the graph in question is relatively sparse or relatively degenerate.

We let C_k stand for a simple cycle of length k. When considering directed graphs, a C_k is assumed to be directed. We show that a C_k in a directed or undirected graph $G = (V, E)$, if one exists, can be found in $O(E^{2-\frac{2}{k}})$ time, if k is even, and in $O(E^{2-\frac{2}{k+1}})$ time, if k is odd. For finding triangles (C_3's), we get the slightly better bound of $O(E^{\frac{2\omega}{\omega+1}}) = O(E^{1.41})$, where $\omega < 2.376$ is the exponent of matrix multiplication.

Even cycles in undirected graphs can be found even faster. A C_{4k-2} in an undirected graph $G = (V, E)$, if one exists, can be found in $O(E^{2-\frac{1}{2k}(1+\frac{1}{k})})$ time. A C_{4k}, if one exists, can be found in $O(E^{2-(\frac{1}{k}-\frac{1}{2k+1})})$ time. In particular, we can find an undirected C_4 in $O(E^{4/3})$ time and an undirected C_6 in $O(E^{13/8})$ time.

The *degeneracy* $d(G)$ of an undirected graph $G = (V, E)$ is the largest minimal degree among the minimal degrees of all the subgraphs G' of G (see Bollobás [Bol78], p. 222). The degeneracy $d(G)$ of a graph G is linearly related to the *arboricity* $a(G)$ of the graph, i.e., $a(G) = \Theta(d(G))$, where $a(G)$ is the minimal number of forests needed to cover all the edges of G. The degeneracy of a directed graph $G = (V, E)$ is defined to be the degeneracy of the undirected version of G. The degeneracy

*Work supported in part by THE BASIC RESEARCH FOUNDATION administrated by THE ISRAEL ACADEMY OF SCIENCES AND HUMANITIES.

†School of Mathematical Sciences, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, ISRAEL.

cycle	complexity		cycle	complexity	
C_3	$E^{1.41}$	$E\cdot d(G)$	C_7	$E^{1.75}$	$E^{3/2}\cdot d(G)$
C_4	$E^{1.5}$	$E\cdot d(G)$	C_8	$E^{1.75}$	$E^{3/2}\cdot d(G)$
C_5	$E^{1.67}$	$E\cdot d(G)^2$	C_9	$E^{1.8}$	$E^{3/2}\cdot d(G)^{3/2}$
C_6	$E^{1.67}$	$E^{3/2}\cdot d(G)^{1/2}$	C_{10}	$E^{1.8}$	$E^{5/3}\cdot d(G)^{2/3}$

Table 1: Finding small cycles in directed graphs – some of the new results

cycle	complexity	cycle	complexity
C_4	$E^{1.34}$	C_8	$E^{1.7}$
C_6	$E^{1.63}$	C_{10}	$E^{1.78}$

Table 2: Finding small cycles in undirected graphs – some of the new results

of a graph is an important parameter of the graph that appears in many combinatorial results. It is easy to see that for any graph $G=(V,E)$ we have $d(G)\le 2E^{1/2}$. For graphs with relatively low degeneracy we can improve upon the previously stated results. A C_{4k} in a directed or undirected graph $G=(V,E)$ that contains one can be found in $O(E^{2-\frac{1}{k}}\cdot d(G))$ time. A C_{4k+1}, if one exists, can be found in $O(E^{2-\frac{1}{k}}\cdot d(G)^{1+\frac{1}{k}})$ time. Similar results are obtained for finding C_{4k-2}'s and C_{4k-1}'s. In particular, C_3's and C_4's can be found in $O(E\cdot d(G))$ time and C_5's in $O(E\cdot d(G)^2)$ time. Some of the results mentioned are summarized in Tables 1 and 2.

As any planar graph has a vertex whose degree is at most 5, the degeneracy of any planar graph is at most 5. As a consequence of the above bounds we get, in particular, that C_3's, C_4's and C_5's in planar graphs can be found in $O(V)$ time. This in fact holds not only for planar graphs but for any non-trivial *minor-closed* family of graphs.

Another contribution of this paper is an $O(V^\omega)$ algorithm for *counting* the number of C_k's, for $k\le 7$, in a graph $G=(V,E)$.

2 Comparison with previous works

Monien [Mon85] obtained, for any fixed $k\ge 3$, an $O(VE)$ algorithm for finding C_k's in a directed or undirected graph $G=(V,E)$. In a previous work [AYZ94] we showed, using the *color-coding* method, that a C_k, for any fixed $k\ge 3$, if one exists, can also be found in $O(V^\omega)$ expected time or in $O(V^\omega \log V)$ worst-case time, where $\omega<2.376$ is the exponent of matrix multiplication.

Our new $O(E^{2-\frac{2}{k}})$ algorithm is better than both the $O(VE)$ and the $O(V^\omega)$ algorithms when the input graph $G=(V,E)$ is sufficiently sparse. It is interesting to note that for $k\le 6$, Monien's $O(VE)$ bound is superseded by either the $O(V^\omega)$ algorithm, when the graph is dense, or by the $O(E^{2-1/\lceil\frac{k}{2}\rceil})$ algorithm, when the graph is sparse. For every $k\ge 7$, each one of the four bounds (including the bound that involves the degeneracy) beats the others on an appropriate family of graphs.

In a previous work [YZ94] we have also showed that cycles of an *even* length in *undirected* graphs can be found even faster. Namely, for any even $k \geq 4$, if an undirected graph $G = (V, E)$ contains a C_k then such a C_k can be found in $O(V^2)$ time. Our $O(E^{2-\frac{1}{2k}(1+\frac{1}{k})})$ bound for C_{4k-2} and $O(E^{2-(\frac{1}{k}-\frac{1}{2k+1})})$ bound for C_{4k} are again better when the graph is sparse enough.

Itai and Rodeh [IR78] showed that a *triangle* (a C_3) in a graph $G = (V, E)$ that contains one can be found in $O(V^\omega)$ or $O(E^{3/2})$ time. We improve their second result and show that the same can be done, in directed or undirected graphs, in $O(E^{\frac{2\omega}{\omega+1}}) = O(E^{1.41})$ time.

Chiba and Nishizeki [CN85] showed that triangles (C_3's) and quadrilaterals (C_4's) in graphs that contain them can be found in $O(E \cdot d(G))$ time. As $d(G) = O(E^{1/2})$ for any graph G, this extends the result of Itai and Rodeh. We extend the result of Chiba and Nishizeki and show that C_{4k-1}'s and C_{4k}'s can be found in $O(E^{2-\frac{1}{k}} \cdot d(G))$ time. We also show that C_{4k+1}'s can be found in $O(E^{2-\frac{1}{k}} \cdot d(G)^{1+\frac{1}{k}})$ time. This gives, in particular, an $O(E \cdot d(G)^2)$ algorithm for finding pentagons (C_5's). Our results apply to both directed and undirected graphs.

Itai and Rodeh [IR78] and also Papadimitriou and Yannakakis [PY81] showed that C_3's in planar graphs can be found in $O(V)$ time. Chiba and Nishizeki [CN85] showed that C_3's as well as C_4's in planar graphs can be found in $O(V)$ time. Richards [Ric86] showed that C_5's and C_6's in planar graphs can be found in $O(V \log V)$ time. We improve upon the result of Richards and show that C_5's in planar graphs can be found in $O(V)$ time. In a previous work [AYZ94] we showed, using color-coding, that for any $k \geq 3$, a C_k in a planar graph, if one exists, can be found in either $O(V)$ *expected* time or $O(V \log V)$ worst case time.

Finally, the fact that the number of triangles in a graph can be counted in $O(V^\omega)$ time is trivial. In [AYZ94] we showed, using color-coding, that for any $k \geq 3$, a C_k, if one exists, can be found in either $O(V^\omega)$ *expected* time or in $O(V^\omega \log V)$ worst case time. Here we show that for any $k \leq 7$ the number of C_k's in a graph can be counted in $O(V^\omega)$ time. The counting method used here yields, in particular, a way of finding C_k's for $k \leq 7$, in $O(V^\omega)$ worst case time.

3 Finding cycles in sparse graphs

Monien [Mon85] obtained his $O(VE)$ algorithm by the use of *representative collections*. Such collections are also used by our algorithms. In the sequel, a p-set is a set of size p.

Definition 3.1 ([Mon85]) *Let $\mathcal{F}$ be a collection of p-sets. A sub-collection $\hat{\mathcal{F}} \subseteq \mathcal{F}$ is q-representative for $\mathcal{F}$, if for every q-set B, there exists a set $A \in \mathcal{F}$ such that $A \cap B = \emptyset$ if and only if there exists a set $A \in \hat{\mathcal{F}}$ with this property.*

It follows from a combinatorial lemma of Bollobás [Bol65] that any collection $\mathcal{F}$ of p-sets, no matter how large, has a q-representative sub-collection of size at most $\binom{p+q}{p}$. Monien [Mon85] describes an $O(pq \cdot \sum_{i=0}^{q} p^i \cdot |\mathcal{F}|)$ time algorithm for finding a q-representative sub-collection of $\mathcal{F}$ whose size is at most $\sum_{i=0}^{q} p^i$. Relying on Monien's result we obtain the following lemma:

Lemma 3.2 *Let $\mathcal{F}$ be a collection of p-sets and let $\mathcal{G}$ be a collection of q-sets. Consider p and q to be fixed. In $O(|\mathcal{F}| + |\mathcal{G}|)$ time, we can either find two sets $A \in \mathcal{F}$ and $B \in \mathcal{G}$ such that $A \cap B = \emptyset$ or decide that no two such sets exist.*

Proof : We use Monien's algorithm to find a q-representative sub-collection $\hat{\mathcal{F}}$ of $\mathcal{F}$ whose size is at most p^q and a p-representative sub-collection $\hat{\mathcal{G}}$ of $\mathcal{G}$ whose size is at most q^p. This takes only $O(|\mathcal{F}| + |\mathcal{G}|)$ time (as p and q are constants).

It is easy to see that if there exist $A \in \mathcal{F}$ and $B \in \mathcal{G}$ such that $A \cap B = \emptyset$, then there also exist $A' \in \hat{\mathcal{F}}$ and $B' \in \hat{\mathcal{G}}$ such that $A' \cap B' = \emptyset$. To see this note that if $A \cap B = \emptyset$ then by the definition of q-representatives, there must exist a set $A' \in \hat{\mathcal{F}}$ such that $A' \cap B = \emptyset$ and then, there must exist a set $B' \in \hat{\mathcal{G}}$ such that $A' \cap B' = \emptyset$ as required.

After finding the representative collections $\hat{\mathcal{F}}$ and $\hat{\mathcal{G}}$ it is therefore enough to check whether they contain two disjoint sets. This can be easily done in constant time (as p and q are constants). □

We also need the following lemma obtained by Monien [Mon85].

Lemma 3.3 ([Mon85]) *Let $G = (V, E)$ be a directed or undirected graph, let $v \in V$ and let $k \geq 3$. A C_k that passes through v, if one exists, can be found in $O(E)$ time.*

We are finally able to present our improved algorithm.

Theorem 3.4 *Deciding whether a directed or undirected graph $G = (V, E)$ contains simple cycles of length exactly $2k - 1$ and of length exactly $2k$, and finding such cycles if it does, can be done in $O(E^{2-\frac{1}{k}})$ time.*

Proof : We describe an $O(E^{2-\frac{1}{k}})$ time algorithm for finding a C_{2k} in a directed graph $G = (V, E)$. The details of all the other cases are similar. Let $\Delta = E^{\frac{1}{k}}$. A vertex in G whose degree is at least Δ is said to be of *high degree*. The graph $G = (V, E)$ contains at most $2E/\Delta = O(E^{1-\frac{1}{k}})$ high degree vertices. We check, using Monien's algorithm (Lemma 3.3), whether any of these high degree vertices lies on a simple cycle of length $2k$. For each vertex this costs $O(E)$ operations and the total cost is $O(E^2/\Delta) = O(E^{2-\frac{1}{k}})$. If one of these vertices does lie on a cycle of length $2k$ we are done. Otherwise, we remove all the high degree vertices and all the edges adjacent to them from G and obtain a subgraph G' that contains a C_{2k} if and only if G does. The maximum degree of G' is at most $\Delta = E^{\frac{1}{k}}$ and there are therefore at most $E{\cdot}\Delta^{k-1} = E^{2-\frac{1}{k}}$ simple directed paths of length k in G'. We can find all these simple path in $O(E^{2-\frac{1}{k}})$ time. We divide these paths into groups according to their endpoints. This can be done using radix sort in $O(E^{2-\frac{1}{k}})$ time and space. We get a list of all the pairs of vertices connected by simple directed paths of length exactly k. For each such pair u, v, we get a collection $\mathcal{F}_{u,v}$ of $k-1$-sets. Each $k-1$-set in $\mathcal{F}_{u,v}$ corresponds to the $k-1$ intermediate vertices that appear on simple directed paths of length k from u to v. For each pair u, v that appears on the list, we check whether there exist two directed paths of length k, one from u to v and the other from v to u, that meet only at their endpoints. Such two paths exist if there exist $A \in \mathcal{F}_{u,v}$ and $B \in F_{v,u}$ such that $A \cap B = \emptyset$. This can be checked, as shown in Lemma 3.2, in $O(|\mathcal{F}_{u,v}| + |\mathcal{F}_{v,u}|)$ time. As the sum of the sizes of all these collections is $O(E^{2-\frac{1}{k}})$, the total complexity is again $O(E^{2-\frac{1}{k}})$. This completes the proof. □

In the case of triangles, we can get a better result by using fast matrix multiplication.

Theorem 3.5 *Deciding whether a directed or an undirected graph $G = (V, E)$ contains a triangle, and finding one if it does, can be done is $O(E^{\frac{2\omega}{\omega+1}}) = O(E^{1.41})$ time.*

Proof : Let $\Delta = E^{\frac{\omega-1}{\omega+1}}$. A vertex is said to be of *high degree* if its degree is more than Δ and of *low degree* otherwise. Consider all directed paths of length two in G whose intermediate vertex is of low degree. There are at most $E\cdot\Delta$ such paths and they can be found in $O(E\cdot\Delta)$ time. For each such path, check whether its endpoints are connected by an edge in the appropriate direction. If no triangle is found in this way, then any triangle in G must be composed of three high degree vertices. As there are at most $2E/\Delta$ high degree vertices, we can check whether there exists such a triangle using matrix multiplication in $O((E/\Delta)^\omega)$ time. The total complexity of the algorithm is therefore

$$O(E\cdot\Delta + (E/\Delta)^\omega) = O(E^{\frac{2\omega}{\omega+1}}).$$

This completes the proof. □

We have not been able to utilize matrix multiplication to improve upon the result of Theorem 3.4 for $k \geq 4$. This constitutes an interesting open problem.

4 Finding cycles in graphs with low degeneracy

An undirected graph $G = (V, E)$ is *d-degenerate* (see Bollobás [Bol78], p. 222) if there exists an acyclic orientation of it in which $d_{out}(v) \leq d$ for every $v \in V$. The smallest d for which G is d-degenerate is called the *degeneracy* or the *max-min degree* of G and is denoted by $d(G)$. It can be easily seen (see again [Bol78]) that $d(G)$ is the maximum of the minimum degrees taken over all the subgraphs of G. The degeneracy $d(G)$ of a graph G is linearly related to the *arboricity* $a(G)$ of the graph, i.e., $a(G) = \Theta(d(G))$, where $a(G)$ is the minimal number of forests needed to cover all the edges of G. The degeneracy of a directed graph $G = (V, E)$ is defined to be the degeneracy of the undirected version of G. It is easy to see that the degeneracy of any planar graph is at most 5. Clearly, if G is d-degenerate then $|E| \leq d\cdot|V|$. The following simple lemma, whose proof is omitted, is part of the folklore (see, e.g., [MB83]).

Lemma 4.1 *Let $G = (V, E)$ be a connected undirected graph $G = (V, E)$. An acyclic orientation of G such that for every $v \in V$ we have $d_{out}(v) \leq d(G)$ can be found in $O(E)$ time.*

Theorem 4.2 *Let $G = (V, E)$ be a directed or an undirected graph.*

(i) Deciding whether G contains a simple cycle of length exactly $4k-2$, and finding such a cycle if it does, can be done in $O(E^{2-\frac{1}{k}}\cdot d(G)^{1-\frac{1}{k}})$ time.

(ii) Deciding whether G contains simple cycles of length exactly $4k-1$ and of length exactly $4k$, and finding such cycles if it does, can be done in $O(E^{2-\frac{1}{k}}\cdot d(G))$ time.

(iii) Deciding whether G contains a simple cycle of length exactly $4k+1$, and finding such a cycle if it does, can be done in $O(E^{2-\frac{1}{k}}\cdot d(G)^{1+\frac{1}{k}})$ time.

Proof : We show how to find a C_{4k+1} in a directed graph $G = (V, E)$, if one exists, in $O(E^{2-\frac{1}{k}} \cdot d(G)^{1+\frac{1}{k}})$ time. The proofs of the other claims are easier. If $d(G) \geq E^{\frac{1}{2k+1}}$, we can use the algorithm of Theorem 3.4 whose complexity is $O(E^{2-\frac{1}{2k+1}}) \leq O(E^{2-\frac{1}{k}}\cdot d(G)^{1+\frac{1}{k}})$. Assume therefore that $d(G) \leq E^{\frac{1}{2k+1}}$.

Let $\Delta = E^{\frac{1}{k}}/d(G)^{1+\frac{1}{k}}$. As $d(G) \le E^{\frac{1}{2k+1}}$, we have that $d(G) \le \Delta$. A vertex is said to be of *high degree* if its degree is more than Δ and of *low degree* otherwise. As in the proof of Theorem 3.4, we can check in $O(E^2/\Delta)$ time whether any of the high degree vertices lies on a C_{4k+1}. If none of them lies on a C_{4k+1}, we can remove all the high degree vertices along with the edges adjacent to them from G and obtain a graph whose maximal degree is at most Δ. The degeneracy of a graph can only decrease when vertices and edges are deleted and $d(G)$ is therefore an upper bound on the degeneracy of the graph obtained.

Suppose therefore that G is a graph with maximal degree Δ and degeneracy $d(G)$. To find a C_{4k+1} in G, it is enough to find all directed simple paths of length $2k$ and $2k+1$ in G and then check, using the algorithm described in the proof of Lemma 3.2, whether there exist a path of length $2k$ and a path of length $2k+1$ that meet only at their endpoints.

In $O(E)$ time we can get an acyclically oriented version G' of G in which the out-degree of each vertex is at most $d(G') \le d(G)$. The orientations of the edges in G and G' may be completely different.

The number of paths, not necessarily directed, of length $2k+1$ in G, is at most

$$2 \cdot 2E \cdot \sum_{i=0}^{k} \binom{2k}{i} \Delta^i d(G)^{2k-i} = O(E\Delta^k d(G)^k).$$

To see this, consider the orientations, in G', of the edges on a $2k+1$-path in G. In at least one direction, at most k of the edges are counter directed. The number of paths of length $2k+1$ in which exactly i among the last $2k$ edges are counter directed is at most $2E \cdot \binom{2k}{i} \Delta^i d(G)^{2k-i}$. The binomial coefficient $\binom{2k}{i}$ stands for the possible choices for the position of the counter directed edges in the path. Similarly, the number of paths of length $2k$ in G is $O(E\Delta^k d(G)^{k-1})$.

We can lower the number of paths of length $2k+1$ and $2k$ we have to consider by utilizing the fact that a C_{4k+1} can be broken into two paths of length $2k+1$ and $2k$ in many different ways. In particular, let C be a directed C_{4k+1} in G and consider the orientations of its edges in G'. As $4k+1$ is odd and as G' is acyclic, C must contain three consecutive edges e_{2k}, e_{2k+1} and e_{2k+2}, the first two of which have the same orientation while the third one has an opposite orientation. It is therefore enough to consider all $2k+1$-paths that start with at least two backward oriented edges and all $2k$-paths that start with at least one backward oriented edge. The orientations referred to here are in G'.

The number of paths of length $2k+1$ in G whose first two edges are backward oriented in G' is $O(E\Delta^{k-1}d(G)^{k+1})$. To see this, note that any such path is composed of a directed path $\{e_{2k}, e_{2k+1}\}$ of length two, attached to an arbitrarily oriented path $\{e_1, \ldots, e_{2k-1}\}$ of length $2k-1$. The number of paths of length $2k-1$ is, as shown earlier, at most $O(E\Delta^{k-1}d(G)^{k-1})$ and the number of directed path of length two with a specified starting point is at most $d(G)^2$. Similarly, there are at most $O(E\Delta^{k-1}d(G)^k)$ $2k$-paths that start with a backward oriented edge.

It should be clear from the above discussion that all the required $2k+1$- and $2k$-paths, whose total number is $O(E\Delta^{k-1}d(G)^{k+1})$, can be found in $O(E\Delta^{k-1}d(G)^{k+1})$ time. Paths which are not properly directed, in G, are thrown away. Properly directed paths are sorted, using radix sort, according to their endpoints. Using Lemma 3.2 we then check whether there exist a directed $2k+1$-path and a directed $2k$-path that close a directed simple cycle. All these operations can again be performed in $O(E\Delta^{k-1}d(G)^{k+1})$ time.

Recalling that $\Delta = E^{\frac{1}{k}}/d(G)^{1+\frac{1}{k}}$, we get that the over all complexity of the algorithm is

$$O(\frac{E^2}{\Delta} + E \cdot \Delta^{k-1} d(G)^{k+1}) = O(E^{2-\frac{1}{k}} d(G)^{1+\frac{1}{k}}) \quad .$$

This completes the proof of the Theorem. □

As an immediate corollary we get:

Corollary 4.3 *If a directed or undirected planar graph $G = (V, E)$ contains a pentagon (a C_5), then such a pentagon can be found in $O(V)$ worst-case time.*

By combining the ideas of this section, the $O(E^{\frac{2\omega}{\omega+1}})$ algorithm of Theorem 3.5 and the color-coding method [AYZ94] we can also obtain the following result.

Theorem 4.4 *Let $G = (V, E)$ be a directed or undirected graph. A C_6 in G, if one exists, can be found in either $O((E \cdot d(G))^{\frac{2\omega}{\omega+1}}) = O((E \cdot d(G))^{1.41})$ expected time or $O((E \cdot d(G))^{\frac{2\omega}{\omega+1}} \cdot \log V) = O((E \cdot d(G))^{1.41} \cdot \log V)$ worst case time.*

Proof : We show how to find a C_6 in an undirected graph $G = (V, E)$, if one exists, in $O((Ed(G))^{\frac{2\omega}{\omega+1}})$ expected time, or $O((Ed(G))^{\frac{2\omega}{\omega+1}} \log V)$ worst case time. The proof of the directed case is similar.

In $O(E)$ time we can get an acyclically oriented version G' of G in which the out-degree of each vertex is at most $d(G)$. There are exactly six possible non-isomorphic orientations of a C_6 of G in G' as shown in Figure 1. We refer to these orientations as $A_1, \ldots, A_6$. Our algorithm outputs, for each $i = 1, \ldots 6$, whether G' contains an A_i, thereby having an algorithm for detecting a C_6 in G.

We show how to detect an A_1 in G', if it exists. The other cases are easier. As in [AYZ94], we color the vertices of G' randomly using 6 colors (i.e. every vertex receives a number between 1 and 6, all numbers equally likely). Let $c(v)$ denote the color of vertex v. Let A be a specific copy of an A_1 in G'. We say that A is *well-colored* if its vertices are consecutively colored by 1 through 6, and the color 1 is assigned to one of the three vertices having only outgoing edges in A. (By "consecutively colored" we mean that each $v \in A$ with $c(v) < 6$ has a neighbor $u \in A$ with $c(u) = c(v) + 1$). The probability that A is well-colored is $\frac{1}{6^6} 6 = \frac{1}{6^5}$. We now show how to detect a well-colored copy of an A_1 deterministically, if it exists. Create a new undirected graph $G^* = (V^*, E^*)$ defined as follows:

$$V^* = \{v \in G' \ : \ c(v) \in \{2, 4, 6\}\}$$

$$\begin{aligned} E^* \quad = \quad & \{(u, v) \ : \ c(u) = 2, \ c(v) = 6 \ \ \exists\, w \in G' \ \ c(w) = 1, \ (w, u), (w, v) \in G'\} \\ \cup \quad & \{(u, v) \ : \ c(u) = 2, \ c(v) = 4 \ \ \exists\, w \in G' \ \ c(w) = 3, \ (w, u), (w, v) \in G'\} \\ \cup \quad & \{(u, v) \ : \ c(u) = 4, \ c(v) = 6 \ \ \exists\, w \in G' \ \ c(w) = 5, \ (w, u), (w, v) \in G'\}. \end{aligned}$$

Clearly, $V^* \leq V$. To create G^*, we examine each edge $(w, u) \in G'$ with $c(w)$ odd. Suppose $c(w) = 1$ and $c(u) = 2$. We create edges in G^* between u and all vertices v such that $(w, v) \in G'$ and $c(v) = 6$. There are at most $d(G) - 1$ such vertices. We therefore have $E^* < Ed(G)$ and G^* can be constructed

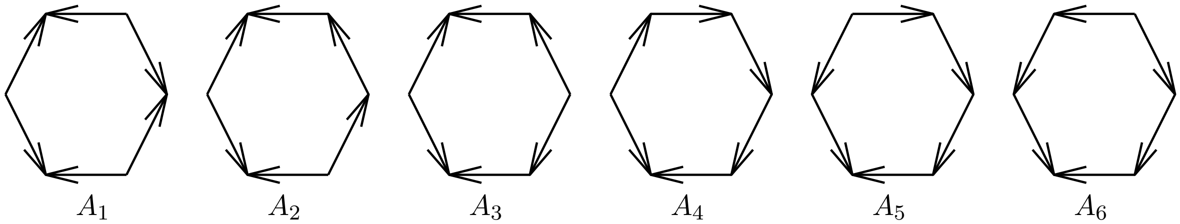

Figure 1: The possible orientations of C_6 in G'.

in $O(Ed(G))$ time from G'. Clearly, there exists an undirected triangle in G^* iff there exists a well-colored A_1 in G'. We can detect such a triangle in G^* in $O((E^*)^{\frac{2\omega}{\omega+1}}) = O((Ed(G))^{\frac{2\omega}{\omega+1}})$ time using the algorithm of Theorem 3.5. Notice that we have shown how to detect an A_1 in G', if it exists, in expected time $O(6^5(Ed(G))^{\frac{2\omega}{\omega+1}}) = O((Ed(G))^{\frac{2\omega}{\omega+1}})$. As shown in [AYZ94], such a coloring scheme can be derandomized at the price of an $O(\log V)$ factor. $\square$

5 Finding cycles in sparse undirected graphs

To obtain the results of this section we rely on the following combinatorial lemma of Bondy and Simonovits [BS74].

Lemma 5.1 ([BS74]) *Let $G = (V, E)$ be an undirected graph. If $|E| \geq 100k{\cdot}|V|^{1+\frac{1}{k}}$ then G contains a $C_{2\ell}$ for every integer $\ell \in [k, n^{1/k}]$.*

By combining the algorithm described in the proof of Theorem 4.2 with an algorithm given in [YZ94] we obtain the following theorem.

Theorem 5.2 *Let $G = (V, E)$ be an undirected graph.*

(i) A C_{4k-2} in G, if one exists, can be found in $O(E^{2-\frac{1}{2k}(1+\frac{1}{k})})$ time.

(ii) A C_{4k} in G, if one exists, can be found in $O(E^{2-(\frac{1}{k}-\frac{1}{2k+1})})$ time.

Proof : We prove the second claim. The proof of the first claim is similar. Let $d = 200k{\cdot}E^{\frac{1}{2k+1}}$. If $d(G) \geq d$ then, by the definition of degeneracy, there is a subgraph $G' = (V', E')$ of G in which the minimal degree is at least d. Such a subgraph can be easily found in $O(E)$ time (see, e.g., [MB83]). Clearly $E' \geq dV'/2 \geq 100k \cdot V' \cdot E'^{\frac{1}{2k+1}}$ and therefore $E' \geq (100k \cdot V')^{1+\frac{1}{2k}} \geq 100k \cdot (V')^{1+\frac{1}{2k}}$. By Lemma 5.1 we get that G' contains a C_{4k} and such a C_{4k} can be found in $O(V'^2) = O(E^{2-\frac{2}{2k+1}})$ time using the algorithm given in [YZ94]. If, on the other hand, $d(G) \leq d$, then a C_{4k} in G, if one exists, can be found in $O(E^{2-\frac{1}{k}} \cdot d) = O(E^{2-(\frac{1}{k}-\frac{1}{2k+1})})$ time using the algorithm of Theorem 4.2. It is easy to check that $E^{2-\frac{2}{2k+1}} \leq E^{2-(\frac{1}{k}-\frac{1}{2k+1})}$ with equality holding only if $k = 1$. In both cases, the complexity is therefore $O(E^{2-(\frac{1}{k}-\frac{1}{2k+1})})$ as required. $\square$

6 Counting small cycles

A graph $H = (V_H, E_H)$ is called *k-cyclic* for $k \geq 3$, if it contains a sequence of vertices $(v_1, \ldots v_k, v_{k+1})$ where $(v_i, v_{i+1}) \in E_H$ for $i = 1 \ldots k$, $v_{k+1} = v_1$ and $\cup_{i=1}^{k}\{(v_i, v_{i+1})\} = E_H$. The number of such sequences is denoted by $c_k(H)$. For example, an undirected triangle is k-cyclic for all $k \geq 3$ except for $k = 4$. Clearly, $c_3(C_3) = 6$. It is also not difficult to see that $c_5(C_3) = 30$. In fact, $c_k(C_k) = 2k$ and for a directed C_k we have $c_k(C_k) = k$.

Let $A = A_G$ be the adjacency matrix of a graph $G = (V, E)$. Assume for simplicity that $V = \{1, \ldots, n\}$. Denote by $a_{ij}^{(k)} = (A^k)_{ij}$ the elements of the k-th power of A. The trace $tr(A^k)$ of A^k, which is the sum of the entries along the diagonal of A^k, gives us the number of closed walks of length k in G. For a graph H, let $n_G(H)$ be the number of subgraphs of G isomorphic to H. Clearly,

$$tr(A^k) = \sum_H n_G(H) c_k(H).$$

If $c_k(H) \neq 0$ then H is connected and has at most k edges. Also, H cannot be a tree on $k+1$ vertices since each bridge must appear twice in a sequence. Hence, $|V_H| \leq k$. In fact, $|V_H| < k$ unless $H = C_k$. We therefore obtain, for an undirected graph $G = (V, E)$:

$$n_G(C_k) = \frac{1}{2k}[tr(A^k) - \sum_{|V_H|<k} n_G(H) c_k(H)]. \tag{1}$$

(For a directed graph, replace the $\frac{1}{2k}$ factor by $\frac{1}{k}$.) It follows that, for fixed k, if we can compute $n_G(H)$ in $O(f(n))$ time for all k-cyclic graphs H with less than k vertices, and we can also compute $tr(A^k)$ in $O(f(n))$ time, then $n_G(C_k)$ can also be computed in $O(f(n))$ time.

We will show that for all k-cyclic graphs H, where $3 \leq k \leq 7$, we can compute $n_G(H)$ in $O(n^\omega)$ time, and hence we obtain:

Theorem 6.1 *The number of C_k's, for $3 \leq k \leq 7$ in a directed or undirected graph $G = (V, E)$ can be found in $O(V^\omega)$ time.*

Proof : We prove the Theorem for the undirected case. The proof of the directed case is simpler. Clearly, $tr(A^k)$ can be computed in $O(V^\omega)$ time using fast matrix multiplication. It remains to show how to compute $n_G(H)$ for all k-cyclic graphs H, where $3 \leq k \leq 7$ in $O(V^\omega)$ time, where H is not a C_k. Figure 2 lists all these graphs, ordered by their number of vertices. We have named these graphs $H_1, \ldots, H_{15}$ as reference. The only 3-cyclic graph is C_3. The 4-cyclic graphs are H_1, H_2 and C_4. The 5-cyclic graphs are C_3, H_5 and C_5. The 6-cyclic graphs are $H_1, H_2, C_3, H_3, H_4, C_4, H_6, H_9, H_{11}$ and C_6. The 7-cyclic graphs are $C_3, H_5, H_6, C_5, H_7, H_8, H_{10}, H_{12}, H_{13}, H_{14}, H_{15}$ and C_7. The following list shows how to obtain $n_G(H_i)$. In all cases, the formulae reference at most $O(V^2)$ values of $a_{ij}^{(p)}$ for some $1 \leq p \leq k$ and hence can be computed in $O(V^\omega)$ time. We put $d_i = a_{ii}^{(2)}$ to denote the degree of vertex i.

1.
$$n_G(C_3) = \frac{1}{6} tr(A^3).$$

2.
$$n_G(H_1) = |E| = \sum_{1 \le i < j \le n} a_{ij}^{(1)}.$$

3.
$$n_G(H_2) = \sum_{i=1}^{n} \binom{d_i}{2}.$$

4.
$$n_G(C_4) = \frac{1}{8}[tr(A^4) - 4n_G(H_2) - 2n_G(H_1)].$$

5.
$$n_G(H_3) = \sum_{(i,j) \in E} (d_i - 1)(d_j - 1) - 3n_G(C_3).$$

6.
$$n_G(H_4) = \sum_{i=1}^{n} \binom{d_i}{3}.$$

7.
$$n_G(H_5) = \frac{1}{2}[\sum_{i=1}^{n} a_{ii}^{(3)}(d_i - 2)].$$

Note that $a_{ii}^{(3)}$ is twice the number of triangles that pass through vertex i.

8.
$$n_G(C_5) = \frac{1}{10}[tr(A^5) - 10n_G(H_5) - 30n_G(C_3)].$$

9.
$$n_G(H_6) = \sum_{(i,j) \in E} \binom{a_{ij}^{(2)}}{2}.$$

Note that $a_{ij}^{(2)}$ is the number of common neighbors of i and j.

10.
$$n_G(H_7) = \frac{1}{2}[\sum_{i=1}^{n} a_{ii}^{(3)} \binom{d_i - 2}{2}].$$

11.
$$n_G(H_8) = \sum_{(i,j) \in E} a_{ij}^{(2)}(d_i - 2)(d_j - 2) - 2n_G(H_6).$$

Note that we must subtract $2n_G(H_6)$ to avoid the case when the two degree-one vertices of H_8 are, actually, the same vertex.

12.
$$n_G(H_9) = \sum_{i=1}^{n} (d_i - 2) \sum_{j \ne i} \binom{a_{ij}^{(2)}}{2}.$$

Note that $\sum_{j \ne i} \binom{a_{ij}^{(2)}}{2}$ is exactly the number of quadrilaterals in which i participates.

13.
$$n_G(H_{10}) = \sum_{i=1}^{n}(\frac{1}{2}a_{ii}^{(3)})(\sum_{j\neq i} a_{ij}^{(2)}) - 6n_G(C_3) - 2n_G(H_5) - 4n_G(H_6).$$

Note that $(\frac{1}{2}a_{ii}^{(3)})(\sum_{j\neq i} a_{ij}^{(2)})$ is simply the number of triangles through i times the number of paths of length two that begin with i. However, we must only count such a triangle and such a path if they are disjoint, so we must subtract appropriate occurrences of C_3, H_5 and H_6.

14.
$$n_G(H_{11}) = \sum_{i=1}^{n} \binom{\frac{1}{2}a_{ii}^{(3)}}{2} - 2n_G(H_6).$$

15. Since we have already shown how to compute $n_G(H)$ for all 6-cyclic graphs, H, that are not C_6, we can use equation (1) to compute $n_G(C_6)$.

16.
$$n_G(H_{12}) = \sum_{(i,j)\in E} a_{ij}^{(2)} \cdot a_{ij}^{(3)} - 9n_G(C_3) - 2n_G(H_5) - 4n_G(H_6).$$

Here we count the number of triangles through (i, j) and multiply each triangle by the number of walks of length 3 between i and j. Since these walks need not be simple, or may intersect the triangle, we may be actually counting C_3, H_5 or H_6. Therefore, we subtract the appropriate values.

17.
$$n_G(H_{13}) = \sum_{(i,j)\in E} \binom{a_{ij}^{(2)}}{3}.$$

18.
$$n_G(H_{14}) = \sum_{i=1}^{n}(d_i - 2) \cdot B_i - 2n_G(H_{12})$$

where B_i is the number of C_5 passing through i. The expression for B_i is:

$$B_i = \frac{1}{2}[a_{ii}^{(5)} - 5a_{ii}^{(3)} - 2a_{ii}^{(3)}(d_i - 2) - 2\sum_{(i,j)\in E} a_{ij}^{(2)}(d_j - 2) - 2\sum_{(i,j)\in E}(\frac{1}{2}a_{jj}^{(3)} - a_{ij}^{(2)})].$$

19.
$$n_G(H_{15}) = \sum_{i=1}^{n}(\frac{1}{2}a_{ii}^{(3)})(\sum_{j\neq i} \binom{a_{ij}^{(2)}}{2}) - 6n_G(H_6) - 2n_G(H_{12}) - 6n_G(H_{13}).$$

□

Similar formulae can be obtained, of course, for the number of octagons (C_8's) and even larger cycles. To compute the number of octagons, however, we have to compute first the number of K_4's in the graph, since a K_4 is 8-cyclic. We do not know how to do this is less than $O(V^{\omega+1})$ time.

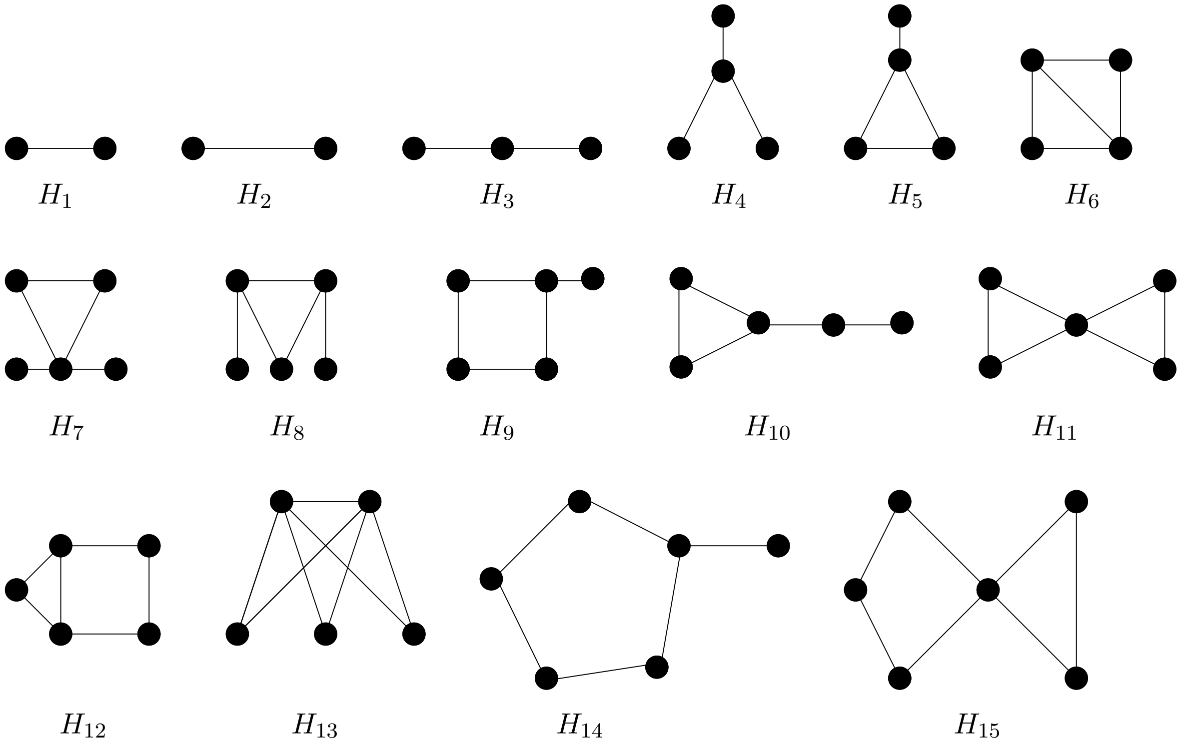

Figure 2: The k-cyclic graphs for $3 \leq k \leq 7$.

References

[AYZ94] N. Alon, R. Yuster, and U. Zwick. Color-coding: a new method for finding simple paths, cycles and other small subgraphs within large graphs. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing, Montréal, Canada*, 1994. To appear.

[Bol65] B. Bollobás. On generalized graphs. *Acta Math. Acad. Sci. Hungar.*, 16:447–452, 1965.

[Bol78] B. Bollobás. *Extremal graph theory.* Academic Press, 1978.

[BS74] J.A. Bondy and M. Simonovits. Cycles of even length in graphs. *Journal of Combinatorial Theory, Series B*, 16:97–105, 1974.

[CN85] N. Chiba and L. Nishizeki. Arboricity and subgraph listing algorithms. *SIAM Journal on Computing*, 14:210–223, 1985.

[IR78] A. Itai and M. Rodeh. Finding a minimum circuit in a graph. *SIAM Journal on Computing*, 7:413–423, 1978.

[MB83] D.W. Matula and L.L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *Journal of the ACM*, 30:417–427, 1983.

[Mon85] B. Monien. How to find long paths efficiently. *Annals of Discrete Mathematics*, 25:239–254, 1985.

[PY81] C.H. Papadimitriou and M. Yannakakis. The clique problem for planar graphs. *Information Processing Letters*, 13:131–133, 1981.

[Ric86] D. Richards. Finding short cycles in a planar graph using separators. *Journal of Algorithms*, 7:382–394, 1986.

[YZ94] R. Yuster and U. Zwick. Finding even cycles even faster. In *Proceedings of the 21st International Colloquium on Automata, Languages and Programming, Jerusalem, Israel*, 1994. To appear.