

René Schott and G. Stacey Staples

Abstract

We show how a number of combinatorial problems, such as determining the number of cycles in graphs, can be recast using a graded commutative algebra constructed within a real Grassmann exterior algebra. The computational complexity of this approach is then measured by considering operations at the basis blade level of the algebra. In particular, the worst-case time complexity of counting arbitrary length cycles in simple n -vertex graphs via nilpotent adjacency matrix methods is shown to be $\mathcal{O}(n^{\alpha+1}2^n)$, where $\alpha \leq 3$ is the exponent representing the complexity of matrix multiplication. The storage complexity of the nilpotent adjacency matrix approach is $\mathcal{O}(n^22^n)$. A probabilistic model is used to describe a class of graphs in which the average-case time complexity of cycle enumeration is $\mathcal{O}(n^3(1+q)^n)$ for fixed $0 < q < 1$. For reference, experimental results detailing computation times (in seconds) are compared with algorithms based on the approaches of Bax and Tarjan.

12.1 Introduction

The complexity of a number of NP-class combinatorial problems can be solved using only a polynomial number of multivector operations in a 2^n -dimensional algebra generated by n mutually commuting null-squares, as we have shown in [5]. In particular, by defining a “nilpotent adjacency matrix” associated with a finite graph

R. Schott

IECN and LORIA, Nancy Université, Université Henri Poincaré, BP 239, 54506
Vandoeuvre-lès-Nancy, France
e-mail: schott@loria.fr

G.S. Staples (✉)

Department of Mathematics and Statistics, Southern Illinois University Edwardsville,
Edwardsville, IL 62026-1653, USA
e-mail: sstaple@siue.edu

on n vertices, the problem of enumerating cycles of length k requires $\mathcal{O}(n^\alpha \log k)$ multivector operations in the algebra, where $\alpha \leq 3$ denotes the exponent associated with matrix multiplication. While $\alpha < 3$ for ordinary matrix multiplication (see [2]), such algorithmic speedups do not necessarily hold for matrices whose elements are multivectors.

In this chapter, the computational complexity of enumerating cycles of arbitrary length in graphs is studied in greater detail by counting algebraic operations at the basis blade level of the algebra as opposed to the multivector level. For practical reference, the theoretical complexity of this approach is compared to that of the algorithms of Bax and Tarjan. Moreover, experimental comparisons using MATHEMATICA illustrate practical advantages of the nilpotent adjacency matrix approach, particularly in the case of sparse graphs.

The algebra used in the construction can be regarded as a commutative subalgebra of the exterior algebra $\bigwedge \mathbb{R}^{2n}$. Its generators commute and square to zero so that linear combinations of the generators are nilpotent. This nilpotent nature is the key to combinatorial applications.

All MATHEMATICA examples were computed on a 2.4 GHz MacBook Pro with 4 GB of 667 MHz DDR2 SDRAM running MATHEMATICA 6 for MAC OS X with the *Combinatorica* package. Cycle enumeration is accomplished using the nilpotent adjacency matrix approach, Bax's approach, and the HamiltonianCycle procedure found in the MATHEMATICA package *Combinatorica*. Time plots comparing the three approaches are included. MATHEMATICA code used to generate examples can be found online through the second-named author's web page, <http://www.siu.edu/~sstaple>.

12.2 Essential Background

A graph $G = (V, E)$ is a collection of vertices V and a set E of unordered pairs of vertices called *edges*. Two vertices $v_i, v_j \in V$ are said to be *adjacent* if there exists an edge $e_{ij} = \{v_i, v_j\} \in E$. In this case, the vertices v_i and v_j are said to be *incident* with e_{ij} .

A k -walk $\{v_0, \dots, v_k\}$ in a graph G is a sequence of vertices in G with *initial vertex* v_0 and *terminal vertex* v_k such that there exists an edge $(v_j, v_{j+1}) \in E$ for each $0 \leq j \leq k - 1$. A k -walk contains k edges. A k -*path* is a k -walk in which no vertex appears more than once. A *closed k -walk* is a k -walk whose initial vertex is also its terminal vertex. A k -*cycle* is a closed k -path with $v_0 = v_k$.

The *diameter* of a graph G is defined to be the length of the longest path in G . The *girth* and *circumference* of a graph are defined to be the lengths of the graph's shortest and longest cycles, respectively. If the graph has no cycles, its girth and circumference are defined to be ∞ .

If $G = (V, E)$ is a graph, A is its adjacency matrix, and $S \subseteq V$, define the modified adjacency matrix A_S by

$$[A_S]_{ij} = \begin{cases} A_{ij} & \text{if } i \in S \text{ and } j \in S, \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 12.1 (Bax) *Each main diagonal element of*

$$\sum_{S \subseteq V} (-1)^{|V|-|S|} (A_S)^{|V|}$$

contains the number of Hamiltonian cycles in G if $|V| > 0$.

Bax’s approach to cycle enumeration uses powers of a graph’s adjacency matrix with the principle of inclusion–exclusion to count all Hamiltonian cycles in $\mathcal{O}(2^n \text{poly}(n))$ time and $\text{poly}(n)$ storage [1]. Enumerating only those cycles of length k is accomplished by applying Bax’s algorithm to all k -vertex subgraphs. For fixed k , this is $\mathcal{O}(\text{poly}(n))$ since $\binom{n}{k} \leq n^k$ for all $n \geq k$. For k increasing with n , the complexity remains $\mathcal{O}(2^n \text{poly}(n))$, which can be verified by a rearrangement of Bax’s formula (cf. Exercise 12.1).

Tarjan’s algorithm (based on pruning with look-ahead) enumerates all cycles in a graph on n vertices with time complexity $\mathcal{O}((n + |E|)(C + 1))$ when applied to a graph with C cycles [7]. The storage complexity is $\mathcal{O}(n + |E| + S)$, where S is the sum of the lengths of all cycles. Note that the number of cycles on a k -vertex subgraph is potentially of order $k!$ while the number of subgraphs supporting such cycles is of order $\binom{n}{k}$.

A convenient and practical Tarjan-type implementation is the HamiltonianCycle procedure found in the MATHEMATICA package *Combinatorica*. The algorithm uses backtracking and look-ahead to enumerate all Hamiltonian cycles in a graph on n vertices. The implementation utilized for the examples in this paper enumerates cycles of length k in an n -vertex graph G by applying HamiltonianCycle to all k -vertex subgraphs of G . Implementations of this Tarjan-like approach are referred to henceforth as “CombiTarjan.” Tarjan’s algorithm actually lists cycles, which can result in $\mathcal{O}(n!)$ space complexity.

12.3 Technical Considerations

Fix positive integer n and let $\gamma = \{\mathbf{e}_i : 1 \leq i \leq 2n\}$ be an orthonormal basis for \mathbb{R}^{2n} . Note that the pair (γ, \wedge) generates a nonabelian semigroup A of order 2^{2n} . Let

$$Z = \{\mathbf{e}_{2i-1} \wedge \mathbf{e}_{2i} : 1 \leq i \leq n\} \subset A, \tag{12.1}$$

and note that the pair (Z, \wedge) generates an abelian subsemigroup $\mathbb{R}\Psi_n$ of A . It should be clear that all elements of Ψ_n square to zero.

Definition 12.1 Let $\mathbb{R}\Psi_n$ denote the real abelian semigroup algebra of Ψ_n . For convenience, the generators of Ψ_n are rewritten as $\zeta_i = \mathbf{e}_{2i-1} \wedge \mathbf{e}_{2i}$, and henceforth the wedge operator is implicit.

Remark 12.1 The algebra $\mathbb{R}\Psi_n$ is isomorphic to the n -particle zeon algebra $\mathcal{C}\ell_n^{\text{nil}}$ appearing in the earlier work [5].

It is evident that the dimension of $\mathbb{R}\Psi_n$ is 2^n and that an arbitrary element $u \in \mathbb{R}\Psi_n$ can be expanded as

$$u = \sum_{I \in 2^{[n]}} u_I \zeta_I, \tag{12.2}$$

where $I \in 2^{[n]}$ is a subset of $[n] = \{1, 2, \dots, n\}$ used as a multiindex, $u_I \in \mathbb{R}$, and $\zeta_I = \prod_{i \in I} \zeta_i$.

A canonical basis element ζ_I is referred to as a *blade*. The number of elements in the multiindex I is referred to as the *grade* of the blade ζ_I .

The *scalar sum* evaluation of an element $u \in \mathbb{R}\Psi_n$, denoted $\langle\langle u \rangle\rangle$, is defined by

$$\langle\langle u \rangle\rangle = \left\langle\left\langle \sum_{I \in 2^{[n]}} u_I \zeta_I \right\rangle\right\rangle = \sum_{I \in 2^{[n]}} u_I. \tag{12.3}$$

Definition 12.2 Let G be a graph on n vertices, either simple or directed with no multiple edges, and let $\{\zeta_i\}$, $1 \leq i \leq n$, denote the nilpotent generators of $\mathbb{R}\Psi_n$. Define the *nilpotent adjacency matrix* associated with G by

$$\mathcal{A}_{ij} = \begin{cases} \zeta_j & \text{if } (v_i, v_j) \in E(G), \\ 0 & \text{otherwise.} \end{cases} \tag{12.4}$$

Recalling Dirac notation, the i th row of \mathcal{A} will be conveniently denoted by $\langle v_i | \mathcal{A}$, while the j th column will be denoted by $\mathcal{A} | v_j \rangle$.

A graph-theoretic interpretation of the nilpotent adjacency matrix can be stated thusly: $\langle v_i | \mathcal{A} | v_j \rangle = \zeta_j$ if and only if one can reach v_j from v_i in one step. Moreover, that “step” algebraically corresponds to multiplication by the null-square generator ζ_j . Extending by induction, nonzero terms of $\langle v_i | \mathcal{A}^k | v_j \rangle$ correspond to k -step walks from v_i to v_j in which each walk is “accomplished” in the algebra by computing a product of null-square generators. The null-square property then naturally “sieves out” walks on distinct generators, i.e., *self-avoiding* walks. This is all made precise in the next theorem.

Theorem 12.2 Let \mathcal{A} be the nilpotent adjacency matrix of an n -vertex graph G . For any $k > 1$ and $1 \leq i, j \leq n$,

$$\langle v_i | \mathcal{A}^k | v_j \rangle = \sum_{\substack{(w_1, \dots, w_k) \in V^k \\ (w_k = v_j) \wedge (m \neq \ell \Rightarrow w_m \neq w_\ell)}} \zeta_{\{w_1, \dots, w_k\}} = \sum_{\substack{I \subseteq V \\ |I|=k}} \omega_I \zeta_I, \tag{12.5}$$

where ω_I denotes the number of k -step walks from v_i to v_j visiting each vertex in I exactly once when initial vertex $v_i \notin I$, and revisiting v_i exactly once when $v_i \in I$. In particular, for any $k \geq 3$ and $1 \leq i \leq n$,

$$\langle v_i | \mathcal{A}^k | v_i \rangle = \sum_{\substack{I \subseteq V \\ |I|=k}} \omega_I \zeta_I, \tag{12.6}$$

where ω_I denotes the number of k -cycles on vertex set I based at $v_i \in I$.

Proof Because the generators of $\mathbb{R}\Psi_n$ square to zero, a straightforward inductive argument shows that the nonzero terms of $\langle v_i | \mathcal{A}^k | v_j \rangle$ are multivectors corresponding to two types of k -walks from v_i to v_j : self-avoiding walks (i.e., walks with no repeated vertices) and walks in which v_i is repeated exactly once at some step but are otherwise self-avoiding. Walks of the second type are zeroed in the k th step when the walk is closed. Hence, terms of $\langle v_i | \mathcal{A}^k | v_i \rangle$ represent the collection of k -cycles based at v_i . \square

In light of this theorem, the name “nilpotent adjacency matrix” is justified by the following corollary.

Corollary 12.1 *Let \mathcal{A} be the nilpotent adjacency matrix of a simple graph on n vertices. For any positive integer $k \leq n$, the entries of \mathcal{A}^k are homogeneous elements of grade k in $\mathbb{R}\Psi_n$. Moreover, $\mathcal{A}^k = \mathbf{0}$ for all $k > n$.*

Another immediate corollary is that

$$\langle\langle \text{tr}(\mathcal{A}^k) \rangle\rangle = k | \{k\text{-cycles in } G\} |, \tag{12.7}$$

since each k -cycle appears with k choices of base point along the main diagonal of \mathcal{A}^k .

In earlier work (see [5, 6]), the authors defined complexity in terms of the number of multivector operations in $\mathcal{C}l_n^{\text{nil}}$, or “*Cl*ops”, required. In contrast, this chapter work considers complexity at the level of the basis blades of $\mathbb{R}\Psi_n \cong \mathcal{C}l_n^{\text{nil}}$.

Definition 12.3 A *blade operation* in $\mathbb{R}\Psi_n$ is defined as computing the sum or product of two basis blades. In particular, for multiindices I and J , each of the following computations is regarded as a blade operation:

$$(a\zeta_I)(b\zeta_J) = \begin{cases} 0 & \text{if } I \cap J \neq \emptyset, \\ (ab)\zeta_{I \cup J} & \text{otherwise;} \end{cases} \tag{12.8}$$

$$a\zeta_I + b\zeta_J = \begin{cases} (a + b)\zeta_I & \text{if } I = J, \\ a\zeta_I + b\zeta_J & \text{otherwise.} \end{cases} \tag{12.9}$$

Recalling the correlation between subsets of $[n]$ and bit strings of length n , each basis blade ζ_I is uniquely associated with a binary string \underline{I} . The cost of a basis blade multiplication in $\mathbb{R}\Psi_n$ is then equal to that of computing first the bitwise AND and then the bitwise OR of two n -bit words, which is known to be $\mathcal{O}(n)$. Summing a pair of basis blades is similarly $\mathcal{O}(n)$.

Given arbitrary elements $u, v \in \mathbb{R}\Psi_n$ and letting ν_u and ν_v denote the respective numbers of nonzero coefficients in the canonical expansions of u and v , the number of blade multiplications required to compute uv is then $\nu_u\nu_v$. The number of blade additions is similarly $\mathcal{O}(\nu_u\nu_v)$. Taking the costs of the blade operations themselves into consideration, the complexity of expanding the product uv is thus seen to be $\mathcal{O}(n\nu_u\nu_v)$.

The MATHEMATICA implementation of $\mathbb{R}\Psi_n$ used in the examples contained herein is based on subset operations rather than binary representations of subsets and bit operations. The additional overhead is offset by the relatively low dimensions of the examples.

12.4 Theoretical Complexity

Lemma 12.1 *Enumerating cycles in a simple graph on n vertices using nilpotent adjacency matrix methods has storage complexity $\mathcal{O}(n^22^n)$.*

Proof The nilpotent matrix method requires construction of $n \times n$ matrices whose entries are elements of a 2^n -dimensional algebra; i.e., in the worst case, $\mathcal{O}(2^n)$ coefficients must be associated with each matrix entry. Consequently, the space complexity is $\mathcal{O}(n^22^n)$. \square

Theorem 12.3 *The worst-case time complexity of enumerating cycles of arbitrary length in a graph on n vertices using the nilpotent adjacency matrix method is $\mathcal{O}(n^{\alpha+1}2^n)$.*

Proof In light of Theorem 12.2, for any $k \leq n$, computing $\mathcal{A}^k = \mathcal{A}^{k-1}A$ requires computing

$$\langle v_i | \mathcal{A}^k | v_j \rangle = \sum_{\ell=1}^n \langle v_i | \mathcal{A}^{k-1} | v_\ell \rangle \langle v_\ell | \mathcal{A} | v_j \rangle \quad (12.10)$$

for all $1 \leq i, j \leq n$. Entries of \mathcal{A}^{k-1} are homogeneous grade- $(k-1)$ elements of \mathcal{L}_n . Moreover, terms in the canonical expansion of $\langle v_i | \mathcal{A}^{k-1} | v_\ell \rangle$ must be indexed by subsets containing v_ℓ , while in all cases, $\langle v_\ell | \mathcal{A} | v_j \rangle$ is either 0 or ζ_{v_j} .

Thus, the maximum number of blade multiplications performed in computing the product $\langle v_i | \mathcal{A}^{k-1} | v_\ell \rangle \langle v_\ell | \mathcal{A} | v_j \rangle$ is $\binom{n-1}{k-2}$ for each $1 \leq \ell \leq n$.

Computing the product $\mathcal{A}^{k-1}\mathcal{A}$ then requires at most $n^\alpha \binom{n-1}{k-2}$ blade multiplications. Applying this result recursively, computing \mathcal{A}^k requires

$$n^\alpha \sum_{\ell=2}^k \binom{n-1}{\ell-2} < n^\alpha 2^{n-1} \quad (12.11)$$

blade multiplications. Since each blade multiplication is of complexity $\mathcal{O}(n)$, the result follows. \square

Note that an immediate consequence of the theorem is that the worst-case complexity of computing the girth and circumference of a graph on n vertices is also $\mathcal{O}(n^{\alpha+1}2^n)$.

Recalling that the diameter of a graph is defined as the length of the graph's longest *path*, another corollary is obtained.

Corollary 12.2 *The worst-case time complexity of computing the diameter of a graph on n vertices using the nilpotent adjacency matrix method is $\mathcal{O}(n^{\alpha+1}2^n)$.*

Proof Letting Δ be the $n \times n$ diagonal matrix whose main diagonal entries are $\Delta_{ii} = \xi_i$, it can be shown that the off-diagonal entries of $\Delta \mathcal{A}^k$ are homogeneous grade- $(k+1)$ multivectors corresponding to k -paths in the graph associated with \mathcal{A} . The effect of left multiplication by Δ is to account for the initial vertex of any walk. Hence, the diameter of the graph is given by the largest positive integer k for which $\Delta \mathcal{A}^k$ is not the zero matrix. \square

Note that the complexity of computing \mathcal{A}^k may vary depending on various methods of computing powers. The *iterated method* requires $k-1$ matrix products to compute

$$\mathcal{A}^k := \begin{cases} \mathcal{A} & \text{if } k = 1, \\ \mathcal{A}^{k-1} \mathcal{A} & \text{otherwise.} \end{cases} \quad (12.12)$$

Given the binary representation of positive integer k , the *successive squares method* requires $\lfloor \log_2 k \rfloor$ matrix products and matrix sums to compute. In particular, letting \underline{k} be a set of nonnegative integers such that $k = \sum_{\ell \in \underline{k}} 2^\ell$, then

$$\mathcal{A}^k = \sum_{\ell \in \underline{k}} \mathcal{A}^{2^\ell}. \quad (12.13)$$

While the successive squares method is generally more efficient than the iterated method, the application to nilpotent adjacency matrices is not straightforward. All discussion is henceforth restricted to the iterated method.

Example 12.1 Computation times in seconds are given for enumerating $\lfloor n/2 \rfloor$ -cycles in randomly generated n -vertex graphs in Fig. 12.1. The experimental results illustrate practical advantages of the nilpotent adjacency matrix approach.

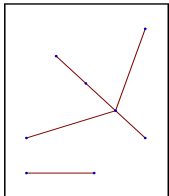
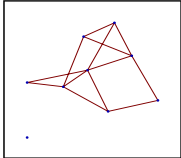
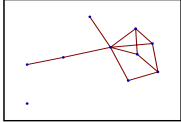
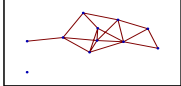
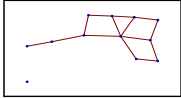
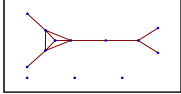
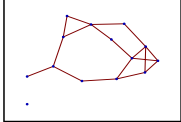
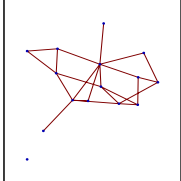
n	p	Zeon Time	Graph	Bax Time	CombiTarjan Time	cycle size	$\# \{k\text{-cycles}\}$
8	0.25	0.003565		0.022864	0.041043	4	0
9	0.25	0.011676		0.040318	0.142099	4	10
10	0.25	0.026255		0.157398	0.376437	5	8
11	0.25	0.074991		0.298306	0.857093	5	40
12	0.25	0.070187		0.907379	1.183048	6	2
13	0.25	0.063806		1.725030	1.602526	6	0
14	0.25	0.252071		6.338007	5.981672	7	6
15	0.25	0.803990		12.365624	10.606065	7	50

Fig. 12.1 Times (in seconds) required to enumerate $\lfloor n/2 \rfloor$ -cycles in randomly generated n -vertex graphs having equiprobable edges ($p = 0.25$)

12.4.1 Average-Case Complexity in “Suitably Sparse” Graphs

Discussion now turns to the average-case complexity of cycle enumeration in homogeneous random graphs. A *homogeneous random graph* $G = \mathcal{G}_{n,p}$ is a graph on n vertices with independent equiprobable edges of probability p . That is, each pair of vertices in the graph has equal probability p of being adjacent.

The next example illustrates the role of graph sparseness in the algorithmic comparisons. For fixed values of n and k , the time required to count k -cycles in an n -vertex graph depends on graph density for the CombiTarjan and nilpotent adjacency matrix methods, but is essentially constant for Bax’s algorithm.

Example 12.2 Mean run times over 20 trials of counting $\lfloor n/2 \rfloor$ -cycles in simple graphs are compared in Fig. 12.2 (top). In Fig. 12.2 (bottom), the number of vertices is fixed at 10, and edge existence probability varies from $p = 0.1$ to $p = 0.5$.

The next theorem describes a class of random graphs for which the nilpotent adjacency matrix method is more efficient than $\mathcal{O}(2^n \text{poly}(n))$.

Theorem 12.4 *Let $q \in (0, 1)$ be fixed. Let n be a positive integer, and let $3 \leq k \leq n$. Let $\mathcal{G}_{n,p}$ be a homogeneous random graph on n vertices with independent equiprobable edges of probability $p \leq \frac{q}{k-1}$. Then, the average-case complexity of enumerating cycles of length less than or equal to k in $\mathcal{G}_{n,p}$ using the nilpotent adjacency matrix method is $\mathcal{O}(n^3(1+q)^n)$.*

Proof As in the proof of Theorem 12.3, the result is obtained by considering the number of nonzero coefficients in \mathcal{A}^k . To consider the average-case complexity, we consider expected numbers of nonzero coefficients according to the probability model indicated. In particular, the average number of blade multiplications performed in computing

$$\langle v_i | \mathcal{A}^k | v_j \rangle = \sum_{\ell=1}^n \langle v_i | \mathcal{A}^{k-1} | v_\ell \rangle \langle v_\ell | \mathcal{A} | v_j \rangle$$

is the product of the expected numbers of nonzero coefficients in the canonical expansions of $\langle v_i | \mathcal{A}^{k-1} | v_\ell \rangle$ and $\langle v_\ell | \mathcal{A} | v_j \rangle$.

Let $G = \mathcal{G}_{n,p}$ be a homogeneous random graph on n vertices with independent equiprobable edges of nonzero probability $p \leq \frac{q}{k-1}$ for fixed $q \in (0, 1)$.

Claim Let $n \geq 3$, and let $2 \leq k \leq n$. For any $1 \leq i, j \leq n$, the expected number of nonzero coefficients in the canonical expansion of $\langle v_i | \mathcal{A}^k | v_j \rangle$ satisfies the following inequality:

$$\mathbb{E}(\#\{\text{nonzero coefficients}\}) \leq q^{k-1} \binom{n-1}{k-1}. \tag{12.14}$$

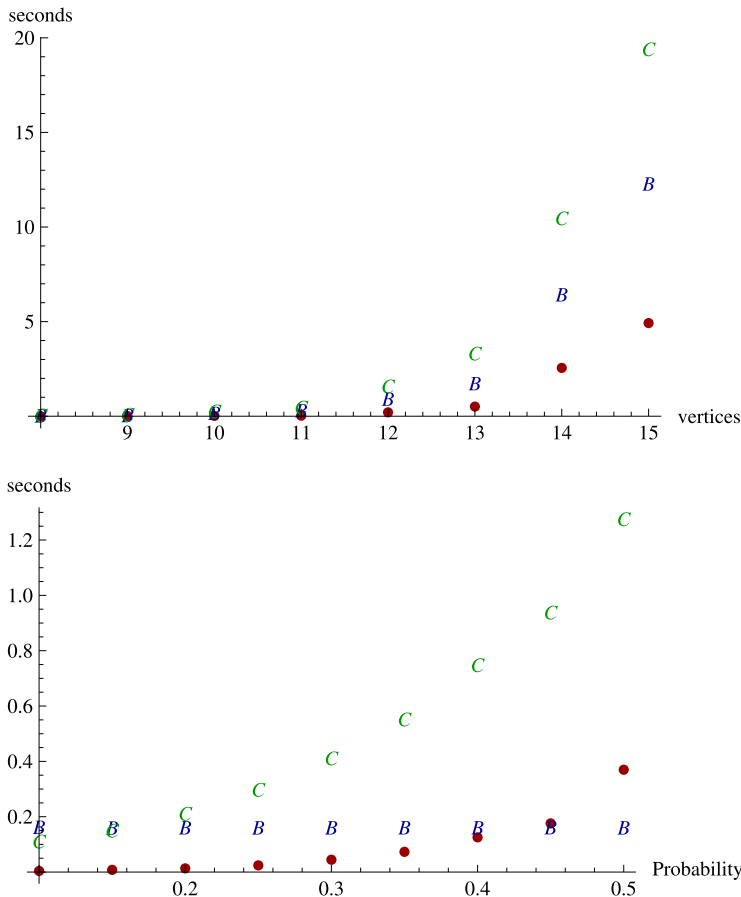


Fig. 12.2 *Top:* Mean run times over 20 trials enumerating $[n/2]$ -cycles in randomly generated n -vertex graphs having equiprobable edges ($p = 0.25$). *Bottom:* 20-Run mean run times of counting 5-cycles in 10-vertex graphs with edge probabilities running from $p = 0.1$ to $p = 0.5$. Plot-markers: B—Bax, C—CombiTarjan, ●— $\mathbb{R}\Psi_n$

Proof of Claim By Theorem 12.2, the expected number of nonzero coefficients in the canonical expansion of $\langle v_i | \mathcal{A}^k | v_j \rangle$ is equal to the expected number of k -vertex subsets $I \subseteq V$ such that there exists a k -step walk from v_i to $v_j \in I$ visiting each vertex of I exactly once when $v_i \notin I$ and revisiting v_i exactly once when $v_i \in I$.

The special case $k = 2$ is treated first. The expected number of nonzero coefficients in the canonical expansion of $\langle v_i | \mathcal{A}^2 | v_j \rangle$ is equal to the expected degree of v_i when $i = j$ and equal to the expected number of two step walks on distinct vertices $v_i \rightarrow v_\ell \rightarrow v_j$ when $i \neq j$; i.e.,

$$\mathbb{E}(\#\{\text{nonzero coefficients}\}) = \begin{cases} p(n-1) & i = j, \\ p^2(n-2) & \text{otherwise.} \end{cases} \tag{12.15}$$

The desired inequality for $k = 2$ is then established by observing that $p < q$, whence

$$p^2(n-2) < p(n-1) < q(n-1) = q^{k-1}(n-1). \quad (12.16)$$

Turning now to the more general case $2 < k \leq n$, the expected number of vertex sets I on which k -walks $v_i \rightarrow v_j$ exist with no repeated vertices except possibly v_i at an intermediate step is determined by partitioning the collection of these walks into two classes: (i) walks on k edges and (ii) walks on $k-1$ edges (in which case, vertex v_i is revisited on the second step).

Unless otherwise indicated, k -walks will refer only to those walks $w : v_i \rightarrow v_j$ with no revisited vertex except possibly v_i exactly once at an intermediate step.

Note that the total number of k -walks $w : v_i \rightarrow v_j$ in K_n revisiting no vertex except possibly v_i at an intermediate step is given by

$$W = (k-1)! \binom{n-1}{k-1} \quad (12.17)$$

since these walks are specified by ordered k -tuples of vertices with v_j in the k th position. Hence, $k-1$ intermediate vertices visited are chosen from $V \setminus \{v_j\}$ with $(k-1)!$ possible permutations.

Denote as Class I those walks on k independent equiprobable edges. Class I walks either revisit no vertices or may revisit v_i at some step other than the second step. Let W_1 denote the total number of these walks in K_n . Denote as Class II those walks on $k-1$ independent equiprobable edges. Class II walks revisit vertex v_i at the second step. Let W_2 denote the number of Class II walks in K_n .

Note first that $W = W_1 + W_2$. Given a homogeneous random graph $G = \mathcal{G}_{n,p}$, it is now evident that

$$\begin{aligned} & \mathbb{E}(\#\{k\text{-walks } w : v_i \rightarrow v_j \text{ in } G\}) \\ &= \mathbb{E}(\#\{\text{Class I } k\text{-walks } w : v_i \rightarrow v_j \text{ in } G\}) \\ & \quad + \mathbb{E}(\#\{\text{Class II } k\text{-walks } w : v_i \rightarrow v_j \text{ in } G\}). \end{aligned} \quad (12.18)$$

When a collection of k -walks $v_i \rightarrow v_j$ exists on k independent equiprobable edges,

$$\begin{aligned} \mathbb{E}(\#\{\text{Class I } k\text{-walks } w : v_i \rightarrow v_j\}) &= \sum_{\text{Class I } k\text{-walks } w : v_i \rightarrow v_j} \mathbb{P}(w \text{ exists}) \\ &= p^k W_1. \end{aligned} \quad (12.19)$$

Similarly,

$$\mathbb{E}(\#\{\text{Class II } k\text{-walks } w : v_i \rightarrow v_j\}) = p^{k-1} W_2. \quad (12.20)$$

Together, (12.18), (12.19), and (12.20) give

$$\begin{aligned} \mathbb{E}(\#\{k\text{-walks } w : v_i \rightarrow v_j \text{ in } G\}) &= p^k W_1 + p^{k-1} W_2 \\ &\leq p^{k-1} (W_1 + W_2) = p^{k-1} W. \end{aligned} \tag{12.21}$$

The expected number of vertex subsets supporting these walks therefore satisfies the following inequality:

$$\mathbb{E}(\#\{I : \exists k\text{-walk } w : v_i \rightarrow v_j\}) \leq p^{k-1} W. \tag{12.22}$$

With the assumption $p \leq \frac{q}{k-1}$ for fixed $q > 0$ and substitution of W from (12.18), one thereby obtains

$$\begin{aligned} \mathbb{E}(\#\{I : \exists k\text{-walk } w : v_i \rightarrow v_j\}) &\leq \frac{q^{k-1}}{(k-1)^{k-1}} W \leq \frac{q^{k-1}}{(k-1)!} W \\ &= q^{k-1} \binom{n-1}{k-1}. \end{aligned} \tag{12.23}$$

This completes the proof of the claim. □

By considering the expected number of nonzero coefficients in the canonical expansion of $\langle v_i | \mathcal{A}^{k-1} | v_j \rangle$ for $3 \leq k \leq n$, it is now evident that the expected number of blade multiplications performed in computing $\langle v_i | \mathcal{A}^{k-1} \mathcal{A} | v_j \rangle$ is bounded above by

$$\begin{aligned} \sum_{\ell=1}^n q^{k-2} \binom{n-1}{k-2} p &= n p q^{k-2} \binom{n-1}{k-2} \leq q^{k-1} \frac{n}{k-1} \binom{n-1}{k-2} \\ &= q^{k-1} \binom{n}{k-1}. \end{aligned} \tag{12.24}$$

Hence, the expected number of blade multiplications performed in computing the matrix product $\mathcal{A}^k = \mathcal{A}^{k-1} \mathcal{A}$ is bounded above by $n^2 q^{k-1} \binom{n}{k-1}$. Applying this result recursively, the average number of blade multiplications required to compute \mathcal{A}^k is found to be bounded above by $n^2 \sum_{\ell=1}^{k-1} q^\ell \binom{n}{\ell}$.

Observing that

$$\sum_{\ell=1}^{k-1} q^\ell \binom{n}{\ell} \leq \sum_{\ell=0}^n q^\ell \binom{n}{\ell} = (1+q)^n, \tag{12.25}$$

cycle enumeration is of average-case complexity $\mathcal{O}(n^2(1+q)^n)$ in terms of blade operations required. Recalling the $\mathcal{O}(n)$ complexity of blade operations thus completes the proof. □

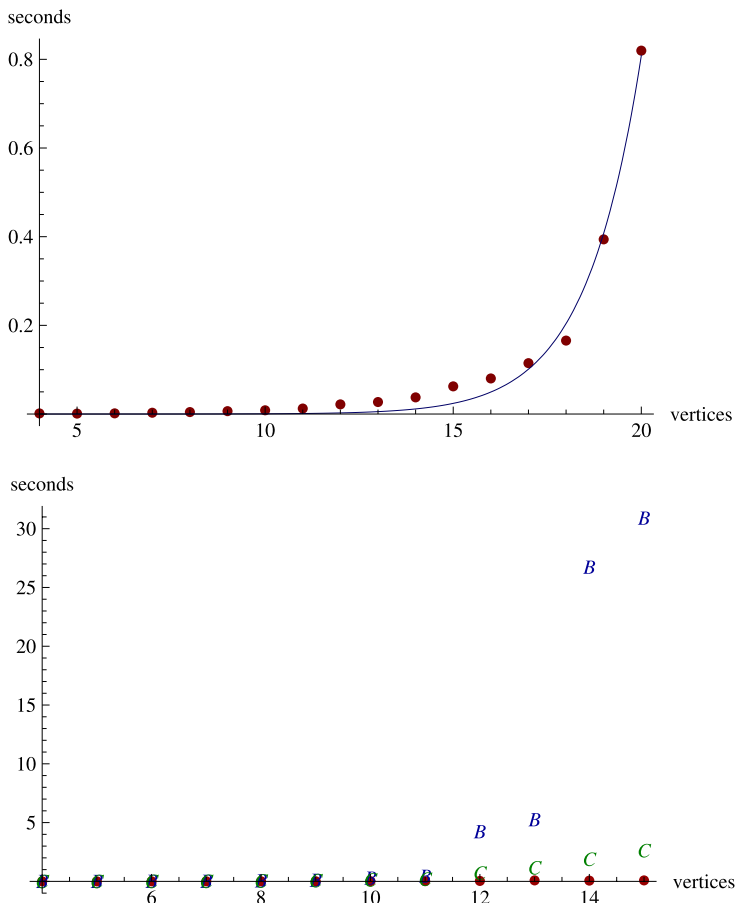


Fig. 12.3 *Top:* Counting cycles of random length $k \in \{3, \dots, \max(\{3, n/2\})\}$ in n -vertex graphs with edge probability $p = q/(k - 1)$. The continuous curve is $y = cn^3(1 + q)^n$, where $q = 0.7$ and $c = 2.4862 \cdot 10^{-9}$, obtained by least squares method. *Bottom:* Counting cycles of length $k \in \{3, \dots, \max(3, \lfloor n/2 \rfloor)\}$ in n -vertex graphs with edge probability $p = q/(k - 1)$, where $q = 0.7$. Plotmarkers: *B*—Bax, *C*—CombiTarjan, \bullet — $\mathbb{R}\Psi_n$

Example 12.3 Average computation times (over 200 trials) of enumerating cycles of length $k \in \{3, \dots, \max(\{3, n/2\})\}$ in homogeneous random graphs satisfying the conditions of Theorem 12.4 with constant $q = 0.7$ are depicted in Fig. 12.3 (top). Also plotted is the curve $y = cn^3(1 + q)^n$ with $c = 2.4862 \cdot 10^{-9}$ determined by least squares.

Example 12.4 A comparison of average computation times (over 20 trials) for the three methods of enumerating cycles of length $k \in \{3, \dots, \max(\{3, n/2\})\}$ in homogeneous random graphs satisfying the conditions of Theorem 12.4 with constant $q = 0.7$ are depicted in Fig. 12.3 (bottom).

As the next theorem shows, the fixed cycle length case is very well-behaved in terms of complexity.

Theorem 12.5 *For fixed $k \in \mathbb{N}$, the worst-case complexity of enumerating k -cycles in an n -vertex graph by the nilpotent adjacency matrix method is $\mathcal{O}(n^{\alpha+k-1})$.*

Proof The case $k = 3$ is clear from the special case in the proof of Theorem 12.4. When $k > 3$, the maximum number of nonzero coefficients in the canonical expansion of $\langle v_i | \mathcal{A}^{k-1} | v_j \rangle$ is $\binom{n-1}{k-2}$. Asymptotically, $\binom{n-1}{k-2} \approx \frac{(n-1)^{k-2}}{(k-2)!} = \mathcal{O}(n^k)$. Hence, computing \mathcal{A}^k requires computing at most

$$n^\alpha \sum_{\ell=0}^{k-2} \binom{n-1}{\ell} = \mathcal{O}(n^\alpha n^{k-2}) = \mathcal{O}(n^{k+(\alpha-2)}) \tag{12.26}$$

blade products. □

12.5 Implementation Notes

In general, coding the geometric product (or any noncommutative operation) in MATHEMATICA is most reliably done using one of MATHEMATICA's undefined symbols. However, since the product in $\mathbb{R}\Psi_n$ is commutative, a much more efficient implementation is possible. The nilpotent adjacency matrix approach was implemented herein by overloading the Times operator of MATHEMATICA to facilitate multiplication of blades from $\mathbb{R}\Psi_n$. Figure 12.4 details MATHEMATICA code for implementing the multiplication in $\mathbb{R}\Psi_n$.

Once the $\mathbb{R}\Psi_n$ multiplication is implemented, the algorithm for counting cycles is very straightforward. The corresponding code is seen in Fig. 12.5.

Bax's algorithm is implemented using the formula obtained in Exercise 12.1. Counting all k -cycles in a graph G having adjacency matrix A is accomplished by computing the quantity

$$k \# \{k\text{-cycles in } G\} = \text{Tr} \left(\sum_{\substack{S \subseteq V \\ |S| \leq k, S \neq \emptyset}} \binom{n-|S|}{k-|S|} (-1)^{k-|S|} (A_S)^k \right). \tag{12.27}$$

The corresponding code is seen in Fig. 12.6.

The CombiTarjan method was implemented by first extracting all k -vertex subgraphs and summing recovered numbers of Hamiltonian cycles on them. The code appears in Fig. 12.7.

The comparisons seen in Fig. 12.1, Fig. 12.3, and Fig. 12.2 were generated as follows: For any given trial, a random simple graph is first generated by constructing a random symmetric binary matrix. The corresponding nilpotent adjacency matrix is then constructed.

```
(* Define Rψ multiplication *)

Unprotect[ψ];
ψ = Symbol["ψ"];

Unprotect[Times];
ψ_a ψ_b := If[Length[a ∩ b] > 0, 0, ψ_a ψ_b];
Unprotect[Power];
(x_ /; ! FreeQ[x, ψ_a])^n_Integer :=
Module[{y, f}, y = Expand[x];
Switch[EvenQ[n], True, If[n == 0, Return[1], Composition[Expand][Distribute[f[x, y]] /. f → Times]^(n/2)],
False, If[n == 1, Return[x], Composition[Expand][Distribute[f[y, y]] /. f → Times]^(n-1/2) x]];

Protect[Power];
Protect[Times];
Unprotect[Expand];
Expand[x_ /; ! FreeQ[x, ψ]] := DeleteCases[Distribute[x, Plus, Times], 0. ψ_];
Protect[Expand];
```

Fig. 12.4 MATHEMATICA code defining $\mathbb{R}\psi_n$ multiplication by overloading the Times operator

```
In[111]:= (* Count k-cycles in simple graph with nilpotent adjacency matrix A *)
NAMCycles[A_, k_] := (Simplify[Tr[Expand[MatrixPower[A, k]]] / Tr[A]] /. {ψ_ → 1};
```

Fig. 12.5 MATHEMATICA code for counting k -cycles via nilpotent adjacency matrix method

```
(* Count k cycles using approach of E.T. Bax *)
BaxKCycleCount[A_, k_] := Module[{list, Kset, verts, nulltable, Ksub, ell},
list = Table[i, {i, 1, Length[A]}];
verts = Length[A];
Kset = Subsets[list, {1, k}];
nulltable = Table[0, {i, 1, verts}, {j, 1, verts}];
Return[Sum[(-1)^(k-Length[Kset[[ell]]]) Binomial[n - Length[Kset[[ell]]], k - Length[Kset[[ell]]]]
MatrixPower[(A ReplacePart[nulltable, CartesianProduct[Kset[[ell]], Kset[[ell]] → 1]), k],
{ell, 1, Length[Kset]}]];]
```

Fig. 12.6 MATHEMATICA code for counting k -cycles via algorithm of Bax

```
(* Count k-cycles in graph using Combinatorica's HamiltonianCycle algorithm *)
KCycleCount[A_, k_] := Module[{list, Kset},
list = Table[i, {i, 1, Length[A]}];
Kset = Subsets[list, {k}];
Return[Sum[Length[HamiltonianCycle[FromAdjacencyMatrix[A[[Kset[[ell]], Kset[[ell]]]]], All]],
{ell, 1, Length[Kset]}]]]
```

Fig. 12.7 MATHEMATICA implementation of “CombiTarjan” approach

The MATHEMATICA system cache was cleared before counting by each method. The system time was stored in a variable, the appropriate method was called, and the subsequent system time was stored in another variable. Relevant data were then appended to a table. Test points were incorporated after each method to ensure that all three methods were returning the same results.

12.6 Conclusion

Leslie Valiant first defined the complexity class $\sharp P$ when dealing with the complexity of counting solutions to NP decision problems [8]. The problem of deciding whether or not a graph contains a Hamiltonian cycle is one such decision problem [3].

Generally, the problem of counting cycles in finite graphs is known to be $\sharp P$ -complete, and no polynomial-time algorithm is known to exist for solving this problem. Algorithms with $\mathcal{O}(2^n \text{poly}(n))$ time complexity and $\mathcal{O}(\text{poly}(n))$ space complexity are known to exist (see [1, 4]).

The time complexity of Tarjan's algorithm is proportional to the number of cycles contained in the graph. Unlike Bax's algorithm, which simply counts cycles, Tarjan's algorithm actually lists the cycles. As a result, the storage and time complexity of Tarjan's algorithm are potentially factorial rather than exponential.

The nilpotent adjacency matrix method, like Bax's algorithm, counts cycles without listing them, providing an advantage over Tarjan's approach in terms of storage-complexity (assuming the counting of cycles is all that is required). Unlike Bax's algorithm, which has time complexity $\mathcal{O}(2^n \text{poly}(n))$ in all cases, the average-case complexity is significantly less when dealing with "suitably-sparse" graphs. The storage complexity of the nilpotent adjacency matrix approach lies between Bax and Tarjan, since it is proportional to the number of vertex subsets supporting cycles in the graph.

As illustrated by the experimental results, the nilpotent adjacency matrix method often has computational advantages over other classical algorithms. Much work remains to be done in characterizing theoretical complexity on various families of graphs.

12.7 Exercises

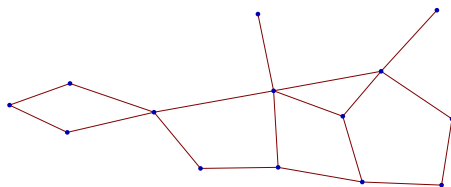
12.1 Let A be the adjacency matrix of a simple graph G on n vertices as seen in Theorem 12.1. Show that for any vertex v_i in G and positive integer $k \leq n$,

$$\sharp\{k\text{-cycles at } v_i\} = \left[\sum_{\substack{S \subseteq V \\ |S| \leq k, S \neq \emptyset}} \binom{n - |S|}{k - |S|} (-1)^{k - |S|} (A_S)^k \right]_{ii}. \quad (12.28)$$

12.2 For positive integer n , let $Z = \{\mathbf{e}_{2i-1} \wedge \mathbf{e}_{2i} : 1 \leq i \leq n\}$ as in (12.1). Prove that the pair (Z, \wedge) forms an abelian semigroup.

12.3 Let $u \in \mathbb{R}\Psi_n$ be an element of the form $u = a_1 \zeta_1 + a_2 \zeta_2$, where a_1 and a_2 are nonzero scalars. Show that $u^2 \neq 0$ and $u^3 = 0$.

Fig. 12.8 A simple graph on 14 vertices



12.4 Suppose that n and k are positive integers satisfying $1 < k \leq n$ and $u \in \mathbb{R}\Psi_n$ is an element of the form $u = \sum_{i=1}^k a_i \zeta_i$ where $a_i \neq 0$ for $i = 1, \dots, k$. Prove the following:

- i. $u^k = k! a_1 \cdots a_k \zeta_{\{1, \dots, k\}}$
- ii. $u^m = 0$ for all $m > k$.

12.5 Construct a nilpotent adjacency matrix \mathcal{A} for the graph appearing in Fig. 12.8 and count the 7-cycles contained therein by computing \mathcal{A}^7 . Verify your result by hand as well as by applying the Bax and CombiTarjan algorithms.

References

1. Bax, E.: Algorithms to count paths and cycles. *Inf. Process. Lett.* **52**, 249–252 (1994)
2. Coppersmith, D., Winograd, S.: Matrix multiplication via arithmetic progressions. *J. Symb. Comput.* **9**, 251–280 (1990)
3. Karp, R.M.: Reducibility among combinatorial problems. In: *Complexity of Computer Computations*, pp. 85–103. Plenum, New York (1972)
4. Karp, R.: Dynamic programming meets the principle of inclusion and exclusion. *Oper. Res. Lett.* **1**, 49–51 (1982)
5. Schott, R., Staples, G.S.: Computational complexity reductions using Clifford algebras. In: Bayro-Corrochano, E., Scheuermann, G. (eds.) *Geometric Algebra Computing for Engineering and Computer Science*, pp. 431–453. Springer, Berlin (2010)
6. Schott, R., Staples, G.S.: Reductions in computational complexity using Clifford algebras. *Adv. Appl. Clifford Algebras* **20**, 121–140 (2010)
7. Tarjan, R.: Enumeration of the elementary circuits of a directed graph. *SIAM J. Comput.* **2**, 211–216 (1973)
8. Valiant, L.: The complexity of computing the permanent. *Theor. Comput. Sci.* **8**, 189–201 (1979)