An evolutionary algorithm for finding an optimal basis of a subspace

Murray R. Bremner

Department of Mathematics and Statistics
University of Saskatchewan
106 Wiggins Road, Room 142
Saskatoon, SK, S7N 5E6, Canada

bremner@math.usask.ca
math.usask.ca/~bremner

10 March 2006

Abstract

This paper presents an evolutionary algorithm for finding a basis of the nullspace of a matrix over the rational numbers which is optimal in the sense that the basis vectors have integral components with no common factors and the absolute values of the components are as small as possible. The algorithm employs a novel variation operator in which an existing basis is combined with one or more randomly generated bases and then filtered by a greedy algorithm to produce a better candidate basis. The paper studies the effectiveness of this algorithm on three examples: (1) a random matrix of size 5×10: for this matrix the algorithm is compared with an exhaustive search; (2) a random matrix of size 10×20: for this matrix the algorithm is tested with population sizes from 1 to 10; (3) a matrix of size 120×90 arising from a computational study of polynomial identities for nonassociative algebras. The better bases located with the algorithm presented here permit the automatic discovery of new algebraic identities with simple statements. This simplification is critical to permitting researchers in abstract algebra to access the intuition embedded in automatically discovered identities.

1 Introduction

We shall thus see that a large amount of hereditary modification is at least possible; and, what is equally or more important, we shall see how great is the power of man in accumulating by his selection successive slight variations. (Charles Darwin [10])

The whole purpose of science is to find meaningful simplicity in the midst of disorderly complexity. (Herbert Simon [20])

1.1 General description

Any subspace of a finite dimensional vector space over a field can be realized as the nullspace of a matrix. The row canonical form of the matrix uniquely determines a basis for the nullspace, since an ordering is implicitly specified on the columns. Over the rational numbers a basis is defined to be optimal if the components of each basis vector are relatively prime integers and if for each basis vector both the nonzero components and the number of nonzero components are as small as possible. This paper presents an evolutionary algorithm for finding such a basis. This algorithm employs a novel variation operator in which an existing basis (the old basis) is combined with one or more randomly generated bases (the new bases) and then filtered by a greedy algorithm to select another basis which is closer to optimal. To produce a new basis for the nullspace, the algorithm randomly permutes the columns of the matrix, recomputes the row canonical form, and derives the corresponding canonical basis for the nullspace. It then creates a list of vectors by merging the old basis with the new basis (or bases), sorts the vectors according to the values of an objective function, and extracts a better basis from the list. This algorithm is compared with exhaustive search (over all permutations of the columns) on a small random matrix (5×10). The behavior of the algorithm for population sizes from 1 to 10 is then analyzed on a larger random matrix (10×20) for which an exhaustive search is not practical. The algorithm is then applied to simplify the known computer generated polynomial identities satisfied by a trilinear nonassociative operation.

1.2 Precise formulation

Consider the following general problem in computational linear algebra:

> Let V be an n-dimensional vector space over a field $\mathbb{F}$. Let U be a nonzero subspace of V. Assume that we have a precise definition of the statement X *is better than* Y for vectors $X, Y \in V$, and a corresponding definition of $\mathcal{B}$ *is better than* $\mathcal{C}$ for bases $\mathcal{B}$, $\mathcal{C}$ of U. Find the best basis for U.

Any subspace U can be represented as the nullspace of a matrix with respect to some given ordered basis of V. (Define a scalar product on V by declaring the given ordered basis to be orthonormal, and then let A be any matrix whose row space is the orthogonal complement of U.) So we can reformulate the problem as follows:

> Let A be an $m \times n$ matrix over the field $\mathbb{F}$. Let $N(A) \subseteq \mathbb{F}^n$ be the nullspace of A and suppose that $N(A) \neq \{0\}$. Assume that we have a precise definition of a partial order $X \leq Y$ for vectors $X, Y \in \mathbb{F}^n$, and a corresponding partial order $\mathcal{B} \leq \mathcal{C}$ on bases of $N(A)$. Find the best basis for $N(A)$.

This is a global optimization problem, and so we can use an evolutionary algorithm to search for solutions.

1.3 Motivation

This problem arose in the computational study of polynomial identities for nonassociative algebras. For surveys of nonassociative algebra, see Kuzmin and Shestakov [16] and Bremner,

Murakami and Shestakov [7]; for more detailed expositions, see Schafer [19] and Zhevlakov, Slinko, Shestakov and Shirshov [21].

The varieties of nonassociative algebras which have received the most attention are those defined by the following identities:

commutativity:	$ab - ba = 0$
anticommutativity:	$ab + ba = 0$
associativity:	$(ab)c - a(bc) = 0$
Jacobi identity:	$[[ab]c] + [[bc]a] + [[ca]b] = 0$
right alternativity:	$(ab)c + (ac)b - a(bc) - a(cb) = 0$
Jordan identity:	$((a \circ a) \circ b) \circ a - (a \circ a) \circ (b \circ a) = 0$
Malcev identity:	$[[[ab]c]d] + [[[bc]d]a] + [[[cd]a]b] + [[[da]b]c] - [[ac][bd]] = 0$

These identities have low degree (≤ 4), very small coefficients (± 1), and very few terms (≤ 5). Computational searches for polynomial identities of degree ≥ 5 satisfied by nonassociative systems often produce results with many large and apparently random coefficients: see Bremner and Hentzel [3, 4, 5, 6], Bremner and Peresi [8, 9], and Tables 20 and 21 in the present paper. In such cases we usually expect that there are simpler identities equivalent to the known identities, but these good identities can be very difficult to find. The present paper provides an effective algorithm for finding these good identities.

2 Preliminaries

Definition 1. *Let $\mathbb{Q}$ be the field of rational numbers. We say that $X = (x_1, \ldots, x_n) \in \mathbb{Q}^n$ is a **good vector** if its components are relatively prime integers:*

$$x_i \in \mathbb{Z} \; (i = 1, \ldots, n) \quad \textit{and} \quad \gcd(x_1, \ldots, x_n) = 1.$$

Lemma 2. *For any nonzero vector $X \in \mathbb{Q}^n$ there is a unique positive rational number r for which rX is a good vector.*

Proof. Clear denominators and cancel common factors. (See Algorithm 10 below.) □

Definition 3. *If $X, Y \in \mathbb{Q}^n$ are good vectors, then we say that X **is better than** Y and write $X \leq Y$ if and only if*

- *either the max norm of X is smaller than the max norm of Y:*

$$\|X\|_\infty < \|Y\|_\infty \quad \textit{where} \quad \|X\|_\infty = \max\{|x_1|, \ldots, |x_n|\};$$

- *or (if the max norms are equal) X has fewer nonzero components than Y:*

$$n(X) < n(Y) \quad \textit{where} \quad n(X) = |\{\, i \mid x_i \neq 0 \,\}|.$$

With this definition, any two vectors $X, Y \in \mathbb{Q}^n$ are comparable (take their unique good scalar multiples from Lemma 2 and compare them using Definition 3), but two distinct vectors can be equally good.

We can give a precise definition for the objective function inducing this partial order on vectors as follows.

Definition 4. *Let $\mathbb{N}$ denote the set of nonnegative integers. We define an **objective function** $v\colon \mathbb{Q}^n \to \mathbb{N}^2$ by*

$$v(X) = \big(\|rX\|_\infty,\, n(rX) \big),$$

*where rX is the unique good scalar multiple of X from Lemma 2. We define a **partial order** on $\mathbb{N}^2$ by*

$$(a_1, b_1) \le (a_2, b_2) \quad \text{if and only if} \quad a_1 \le a_2, \text{ or } a_1 = a_2 \text{ and } b_1 \le b_2.$$

We can now restate the definition of one vector being better than another.

Lemma 5. *The condition that X is better than Y for vectors $X, Y \in \mathbb{Q}^n$ is equivalent to the inequality $v(X) \le v(Y)$.* □

The partial order on vectors induces a partial order on bases as follows.

Definition 6. *Let $U \subseteq \mathbb{Q}^n$ be a subspace of dimension $d \ge 1$, and let*

$$\mathcal{B}\colon X^{(1)} \le \cdots \le X^{(d)} \quad \text{and} \quad \mathcal{C}\colon Y^{(1)} \le \cdots \le Y^{(d)},$$

*be two ordered bases of U. We say that $\mathcal{B}$ **is better than** $\mathcal{C}$, and write $\mathcal{B} \le \mathcal{C}$, if and only if $X^{(d)} \le Y^{(d)}$; that is, the worst vector in $\mathcal{B}$ is no worse than the worst vector in $\mathcal{C}$. If we define the **max norm on a basis** $\mathcal{B}$ in terms of the max norm on vectors by*

$$\|\mathcal{B}\|_\infty = \max\{\|X^{(1)}\|_\infty, \ldots, \|X^{(d)}\|_\infty\},$$

then $\mathcal{B} \le \mathcal{C}$ if and only if $\|\mathcal{B}\|_\infty \le \|\mathcal{C}\|_\infty$.

This gives a precise meaning to the concept of a better basis. We can now restate our problem as follows: We want to find an ordered basis for $N(A)$ which is optimal in the sense that each vector in the basis is good, the vectors in the basis are sorted by decreasing goodness, and the basis is as good as possible in the partial order on bases induced by the order on vectors.

3 The evolutionary algorithm

The classic text on evolutionary algorithms is Goldberg [13]. A contemporary introduction to evolutionary computation is Ashlock [1]. A collection of survey articles on related strategies for optimization is Glover and Kochenberger [12].

Before discussing the evolutionary aspects of the algorithm, we review two algorithms from linear algebra which produce a good basis of the nullspace $N(A)$ of a matrix A.

Algorithm 7. Nullspace basis. This algorithm produces the canonical basis for the nullspace of a matrix.

Input: An $m \times n$ matrix A over the field $\mathbb{F}$.

1. Compute the row canonical form $\mathrm{rcf}(A)$, also known as the reduced row-echelon form, or the Gauss-Jordan form. See Press, Teukolsky, Vetterling and Flannery [17] (Chapter 2) for the standard algorithms, and von zur Gathen and Gerhard [11] (Chapter 12) for more recent developments.

2. Let
$$F = \{f_1, \ldots, f_d\} \subseteq \{1, \ldots, n\},$$
be the indices of the columns which do not contain the leading 1 of any nonzero row of $\mathrm{rcf}(A)$. These columns represent free variables in the solution of the homogeneous system $AX = O$. Then $d = |F| = \dim N(A)$.

3. Obtain a basis for $N(A)$ from $\mathrm{rcf}(A)$ as follows. For $1 \leq k \leq d$ define
$$Z^{(k)} = \left(z_1^{(k)}, \ldots, z_n^{(k)}\right) \in \mathbb{F}^n,$$
by the formulas:
$$\begin{aligned}
z_j^{(k)} &= 1 && \text{if } j \in F \text{ and } j = f_k, \\
z_j^{(k)} &= 0 && \text{if } j \in F \text{ and } j \neq f_k, \\
z_j^{(k)} &= 0 && \text{if } j \notin F \text{ and } f_k < j, \\
z_j^{(k)} &= -\mathrm{rcf}(A)_{i,\, f_k} && \text{if } j \notin F \text{ and } j < f_k \text{ and row } i \text{ has its leading 1 in column } j.
\end{aligned}$$
That is, to find $Z^{(k)}$ we set the free variables equal to the components of the k-th standard basis vector E_k in $\mathbb{F}^d$ (the vector E_k has 1 in position k and 0 in the other positions) and then solve for the other variables using $\mathrm{rcf}(A)$ as the coefficient matrix.

Output: The vectors $Z^{(1)}$, ..., $Z^{(d)}$.

Proposition 8. *The vectors $Z^{(1)}$, ..., $Z^{(d)}$ form a basis for $N(A)$.*

Proof. The algorithm constructs an isomorphism of vector spaces from $\mathbb{F}^d$ to $N(A)$ for which the image of the k-th standard basis vector E_k in $\mathbb{F}^d$ is the k-th basis vector $Z^{(k)}$ of $N(A)$. □

The following theorem implies that this basis is uniquely determined by the implicit order on the columns of the matrix. Our evolutionary algorithm will exploit this fact by repeatedly generating a random permutation of the columns of the matrix in order to produce a different basis of the nullspace.

Theorem 9. *Let m and n be positive integers and let $\mathbb{F}$ be a field. Suppose U is a subspace of $\mathbb{F}^n$ with $\dim U \leq m$. Then there is a unique $m \times n$ matrix over $\mathbb{F}$ in row canonical form which has U as its row space.*

Proof. Hoffman and Kunze [15] (Section 2.5, Theorem 11). □

Algorithm 10. Good vector. This algorithm produces the unique nonzero good scalar multiple rX ($r > 0$) of the input vector X over the field $\mathbb{Q}$ of rational numbers.

Input: A vector $X = (x_1, \ldots, x_n) \in \mathbb{Q}^n$.

1. Write $x_i = \dfrac{a_i}{b_i}$ uniquely such that $a_i, b_i \in \mathbb{Z}$, $b_i > 0$ and $\gcd(a_i, b_i) = 1$.
2. Compute $\ell = \operatorname{lcm}(b_1, \ldots, b_n)$: the least common multiple of the denominators of X.
3. Set $X' = (\ell x_1, \ldots, \ell x_n) \in \mathbb{Z}^n$.
4. Compute $g = \gcd(\ell x_1, \ldots, \ell x_n)$: the greatest common divisor of the components of X'.
5. Set $X'' = \left(\dfrac{\ell x_1}{g}, \ldots, \dfrac{\ell x_n}{g}\right) \in \mathbb{Z}^n$; then $X'' = rX$ where $r = \dfrac{\ell}{g} > 0$.

Output: The good vector $rX \in \mathbb{Z}^n$.

Proposition 11. *For any nonzero vector $X \in \mathbb{Q}^n$ the vector rX produced by Algorithm 10 is the unique nonzero good scalar multiple with $r > 0$.*

Proof. Suppose that some other good vector $Y \neq X''$ is a scalar multiple of X. Then Y is a scalar multiple of X'', so we can write $Y = (a/b)X''$ where $a, b \in \mathbb{Z}$ have $\gcd(a, b) = 1$. Since Y is an integral vector and the components of X'' are relatively prime integers, it follows that $b = 1$. Then $Y = aX''$, and since the components of Y are relatively prime integers, it follows that $a = \pm 1$. Hence $Y = \pm X''$. □

Algorithm 12. Random permutation. This algorithm uses a pseudorandom number generator to produce a permutation π of the columns of the matrix. We assume that we have a procedure `random`(n) which produces uniformly distributed pseudorandom integers $1 \leq x \leq n$.

Input: A positive integer n.

1. Initialize the lists (ordered sets) $X = [1, \ldots, n]$ and $\pi = [\,]$.
2. For i from 1 to n do:
   - Set $j =$ `random`$(n - i + 1)$.
   - Append $X[j]$ (the j-th element of X) to π.
   - Remove $X[j]$ from X.

Output: The permutation π of $1, \ldots, n$.

Algorithm 13. Better vector. This algorithm (a Boolean function) determines whether one vector is better than another.

Input: Two good vectors X, Y of length n.

1. Compute $X_{\max} = \max\{|x_1|, \ldots, |x_n|\}$ and $Y_{\max} = \max\{|y_1|, \ldots, |y_n|\}$.
2. If $X_{\max} < Y_{\max}$ then set `result := true`.
3. If $X_{\max} > Y_{\max}$ then set `result := false`.
4. If $X_{\max} = Y_{\max}$ then compute $X_{\text{nonzero}} = |\{\ i \mid x_i \neq 0\ \}|$ and $Y_{\text{nonzero}} = |\{\ i \mid y_i \neq 0\ \}|$.
5. If $X_{\text{nonzero}} \leq Y_{\text{nonzero}}$ then set `result := true`, otherwise set `result := false`.
6. Return `result`.

Output: `True` if X is better than (or at least not worse than) Y, and `false` otherwise.

Algorithm 14. Merge and sort. This algorithm merges the old basis and the p new bases of the nullspace into one multiset of vectors which is then sorted according to the comparison function of Algorithm 13. Here p is a positive integer which represents the population size in the evolutionary algorithm.

Input: Ordered bases $X^{(k,1)}, \ldots, X^{(k,d)}$ for $1 \leq k \leq p+1$ of the d-dimensional nullspace of the $m \times n$ matrix A over the field $\mathbb{Q}$.

1. Combine the $p+1$ bases into an ordered list of $(p+1)d$ vectors

$$Y^{(1)}, \ldots, Y^{((p+1)d)}.$$

   Here $Y^{(\ell)} = X^{(k,i)}$ where ℓ is given by the formula $\ell = (k-1)d + (i-1)$. In other words, we divide ℓ by d to get the quotient $k-1$ and the remainder $i-1$.

2. Sort the ordered list $Y^{(1)}, \ldots, Y^{((p+1)d)}$ using any convenient sorting algorithm together with the comparison function of Algorithm 13 to obtain a sorted list of vectors

$$Z^{(1)}, \ldots, Z^{((p+1)d)}.$$

Output: An ordered list $Z^{(1)}, \ldots, Z^{((p+1)d)}$ of good vectors in which $Z^{(i)}$ is better than (or at least not worse than) $Z^{(j)}$ whenever $1 \leq i < j \leq (p+1)d$.

Algorithm 15. Better basis. This algorithm selects a better basis for the nullspace from the $p+1$ known bases. We are given an ordered set of vectors

$$X^{(1)} \leq X^{(2)} \leq \cdots \leq X^{((p+1)d)},$$

which is obtained from one old basis and p new bases of the nullspace by Algorithm 14. We then extract the best basis from this sorted list. This is the greedy algorithm which we use to filter the old and new bases to produce a better basis. This algorithm uses a standard result from linear algebra for finding a subset of a given set of vectors which forms a basis for the subspace spanned by the vectors.

Input: An ordered list $X^{(1)}, \ldots, X^{((p+1)d)}$ of good vectors in which $1 \leq i < j \leq (p+1)d$ implies that $X^{(i)}$ is better than (or at least not worse than) $X^{(j)}$. We regard $X^{(j)}$ as a column vector of size $n \times 1$.

1. Construct the $n \times (p+1)d$ matrix T which has $X^{(j)}$ as column j for $1 \leq j \leq (p+1)d$; that is, $T_{ij} = X_i^{(j)}$ for $1 \leq i \leq n$.

2. Compute the row canonical form $\mathrm{rcf}(T)$. We know that the columns of T span a subspace U of dimension d, so the rank of $\mathrm{rcf}(T)$ is d; that is, $\mathrm{rcf}(T)$ has d nonzero rows.

3. Let $j_1, \ldots, j_d$ be the columns of $\mathrm{rcf}(T)$ which contain the leading 1 of some nonzero row.

Output: The d vectors $X^{(j_1)}, \ldots, X^{(j_d)}$ which form a basis of U consisting of the best d vectors (subject to the constraints of spanning and linear independence) extracted from the original sorted list of $(p+1)d$ vectors from the $p+1$ old and new bases.

The justification for this algorithm is the following result from linear algebra.

Theorem 16. *Let $X^{(j)}$ for $1 \leq j \leq q$ be column vectors in $\mathbb{F}^n$. Let T be the $n \times q$ matrix with $X^{(j)}$ in column j; that is, $T_{ij} = X_i^{(j)}$. Suppose that the rank of T is d, and let the leading ones of the nonzero rows of $\mathrm{rcf}(T)$ occur in columns $j_1, \ldots, j_d$. Then the vectors $X^{(j_1)}, \ldots, X^{(j_d)}$ form a basis of the column space of T.*

Proof. This is a corollary of the fact that for any matrix the row rank equals the column rank; see Hoffman and Kunze [15] (Section 3.7, Theorem 22). Let r_k for $1 \leq k \leq q$ be the rank of the $n \times k$ submatrix consisting of the first k columns of T. Then r_k is the column rank of the submatrix, and hence as k increases, r_k will increase when the column rank increases. Since r_k is also the row rank, this will happen exactly when there is a new leading 1 in a row of $\mathrm{rcf}(T)$. Hence the positions of the leading ones in $\mathrm{rcf}(T)$ give the columns which increase the dimension of the column space as k goes from 1 to q. □

It is important to note that, since the original list of vectors is ordered, this procedure preserves the order on the vectors: that is, the basis chosen by this algorithm is the best basis with respect to the partial order on the vectors. Furthermore, the basis we extract from the $p+1$ old and new bases is better (or at least no worse) than the input bases.

Theorem 17. *Algorithm 15 never produces a basis that is worse than any of the input bases.*

Proof. We have $p+1$ bases for a d-dimensional subspace of an n-dimensional space. Altogether this gives $(p+1)d$ vectors (possibly with repetitions). We put the vectors (as columns) into a matrix of size $n \times (p+1)d$. We sort the vectors according to the objective function, with the best vectors to the left. We reduce the matrix to find a subset of the (sorted list of) vectors which is a basis of the subspace. This gives us the "best" basis using vectors from the original $p+1$ bases. We have to prove that this "best" basis is better than (or at least no worse than) any of the $p+1$ original bases. Let $X^{(1)}, X^{(2)}, \ldots, X^{(p+1)}$ be the worst vectors in the original $p+1$ bases. The order of the vectors within each basis is important, but the order of the $p+1$ bases is irrelevant, so we may assume that the $p+1$ bases have already been sorted using the max norm for bases (the norm of a basis is the max norm of its worst vector). This means that we may assume without loss of generality that $X^{(1)} \leq X^{(2)} \leq \cdots \leq X^{(p+1)}$ using our partial order on vectors. So the entire first basis

occurs (at and) to the left of the position of $X^{(1)}$. This means that all the leading ones of the reduced matrix occur (at or) to the left of the column which contains $X^{(1)}$. Therefore none of the chosen vectors is any worse than $X^{(1)}$. Since $X^{(1)}$ is no worse than any of the other worst vectors $X^{(2)}, \ldots, X^{(p+1)}$, it follows that none of the chosen vectors is any worse than any of the worst vectors in the original $p+1$ bases. In particular, the max norm of the new basis is less than or equal to the minimum of the max norms of the original $p+1$ bases. □

Algorithm 18. Main procedure: evolutionary algorithm. This procedure combines all of the previous procedures and iterates them to find the best possible basis of the nullspace.

Input:

- An $m \times n$ matrix A with entries from $\mathbb{Q}$. The nullspace of A has dimension $d > 0$.
- The population size p, a positive integer. This is the number of new bases that will be generated during each iteration of the algorithm.
- The number of generations g, a positive integer. This is the number of iterations of the algorithm that will be performed.

Procedure:

1. Use Algorithm 7 to compute a basis for the nullspace of A; then use Algorithm 10 to replace each basis vector by its unique good positive scalar multiple; and finally sort this basis according to Algorithm 13, giving the integral basis $X^{(1)}, \ldots, X^{(d)}$.
2. Repeat g times:
   (a) Call Algorithm 12 a total of p times to generate p random permutations $\pi_1, \ldots, \pi_p$.
   (b) Initialize p other $m \times n$ matrices $B_1, \ldots, B_p$ as follows:
   $$(B_k)_{i,j} = A_{i,\pi_k(j)} \text{ for } 1 \le i \le m,\ 1 \le j \le n,\ 1 \le k \le p.$$
   This is the first phase of our novel variation operator: a random permutation of the columns of the original matrix A takes the place of the mutation operation in a genetic algorithm.
   (c) Call Algorithm 7 a total of p times to compute bases for the nullspaces of the matrices $B_1, \ldots, B_p$; and then use Algorithm 10 to replace each basis vector by its unique good positive scalar multiple. Denote the good basis vectors obtained from B_k by
   $$Y^{(k,1)}, \ldots, Y^{(k,d)}\ (1 \le k \le p).$$
   (d) We need to unpermute the components of the vectors $Y^{(k,\ell)}$, since the columns of each matrix B_k are a permutation of the columns of the original matrix A. So for $1 \le k \le p$ we set
   $$Z^{(k,\pi_k(\ell))} = Y^{(k,\ell)}\ (1 \le \ell \le d).$$

(e) Use Algorithm 14 to merge and sort the old basis with the p new bases:

$$X^{(1)}, \ldots, X^{(d)} \quad \text{and} \quad Z^{(k,1)}, \ldots, Z^{(k,d)} \ (1 \leq k \leq p).$$

This is the second phase of our novel variation operator: merging and sorting the $p+1$ old and new bases takes the place of the crossover operation in a genetic algorithm.

(f) Use Algorithm 15 to extract a better basis $W^{(1)}, \ldots, W^{(d)}$ from the merged and sorted list of vectors produced by the previous step. This extraction of a better basis from the $p+1$ old and new bases takes the place of the selection operation in a genetic algorithm.

(g) Set $X^{(i)}$ equal to $W^{(i)}$ for $i = 1, \ldots, d$.

Output:

- A basis $X^{(1)}, \ldots, X^{(d)}$ for the nullspace of A which contains the best vectors from the gp bases generated by the evolutionary algorithm.

4 Relation of this algorithm to the theory of evolutionary computation

In this section we relate the algorithm to the standard operations of evolutionary algorithms.

Let Π_n denote the set of all permutations of $\{1, \ldots, n\}$. Every permutation $\pi \in \Pi_n$ permutes the columns of the original matrix A to determine a new matrix $B = A^\pi$. The new matrix B then uniquely determines a basis of good vectors for the nullspace $N(A)$ according to Algorithms 7 and 10. So we have the schematic diagram

$$\pi \in \Pi_n \longrightarrow B = A^\pi \longrightarrow Y^{(1)}, \ldots, Y^{(d)}$$

The first stage of our novel variation operator (corresponding roughly to the mutation step of a genetic algorithm) takes place at the left end of the diagram when we generate a new random permutation $\pi \in \Pi_n$. The permutation π corresponds to the genotype in a genetic algorithm. The permutation π then determines, by means of the matrix $B = A^\pi$, a new basis $Y^{(1)}, \ldots, Y^{(d)}$ of $N(A)$. This new basis corresponds to the phenotype in a genetic algorithm. The second stage of our novel variation operator takes place at the right end of the diagram when we merge and sort the old basis and the p new bases. This merging and sorting corresponds to the crossover step of a genetic algorithm. (For a similar novel variation operator which combines operations corresponding to mutation and crossover see the recent work of Ashlock and Houghten [2].) The selection stage of our evolutionary algorithm takes place at the right end of the diagram when we extract a better basis from the sorted list containing the old basis and the p new bases.

In this algorithm, an individual is a particular basis for the nullspace $N(A)$, and the population is a collection of distinct bases (possibly with vectors in common). We start with an population of size 1, and repeatedly produce a new population of size p, which we mate with the old population to produce a better population of size 1. During the selection step

we must be careful to preserve the property of linear independence for the basis of a vector space. This is not a genetic algorithm, since we are not applying a mutation operator to the genotypes, but rather generating new random genotypes, and then combining the phenotypes corresponding to the old and new genotypes. It is important to note that the good basis eventually produced by the algorithm will not in general be the phenotype corresponding to one particular genotype, but will be a combination of phenotypes corresponding to many different genotypes.

5 Open problems

5.1 Optimality of the output

An important open problem is to find sufficient conditions for the output of this algorithm to be optimal. In other words, what is the probability that we have found the optimal basis as a function of the number of generations? This will also depend on the size of the matrix, the nature of the matrix entries, and the size of the population. We can give a precise definition of the optimal basis as follows: Let A be an $m \times n$ matrix over $\mathbb{Q}$. Consider the set Π_n of all $n!$ permutations of the n columns of A, ordered in some convenient way (for example, lexicographically). Each permutation $\pi \in \Pi_n$ produces a corresponding permuted matrix A^π, and each of these permuted matrices gives a corresponding canonical basis $X^{(\pi,1)}$, $\ldots$, $X^{(\pi,d)}$ of the nullspace of dimension d. Let T be the matrix of size $n \times n!d$ consisting of $n!$ blocks of size $n \times d$ arranged horizontally in which the d column vectors of the k-th block are the nullspace basis vectors corresponding to the k-th permutation. Let $\mathrm{sort}(T)$ be the matrix obtained from T by sorting the columns according to Algorithm 13. The row canonical form of $\mathrm{sort}(T)$ has rank d; the leading ones of its rows occur in columns $1 \leq j_1 < j_2 < \cdots < j_d \leq n!d$. The nullspace basis vectors appearing in columns $j_1, \ldots, j_d$ of T will be, by definition, the optimal basis for the nullspace.

5.2 Changing the partial order on vectors

There are many other objective functions that we could use to partially order the vectors. The most important criterion is that the objective function should take into account both the size of the components and the number of nonzero components. The partial order of Definition 3 does this by first comparing the size of the components and then comparing the number of nonzero components. If we use another norm on the vectors, we can combine these two steps in one comparison. For example, we could use the sum of the absolute values of the components, or the familiar Euclidean norm, or the general norm determined by a positive real number α:

$$\|X\|_1 = \sum_{i=1}^{n} |x_i|, \qquad \|X\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}, \qquad \|X\|_\alpha = \left(\sum_{i=1}^{n} |x_i|^\alpha \right)^{1/\alpha}$$

Any one of these norms would simultaneously select in favor of vectors with smaller components and with fewer nonzero components. It is an interesting open problem to investigate

how different vector norms would affect the performance of the algorithm. A closely related problem is to study the computational complexity of the algorithm, and in particular to determine how the optimal population size depends on the matrix.

5.3 Searching over all bases of the nullspace

It is possible that the best basis of the nullspace cannot be located by this algorithm. The definition of optimality given in subsection 5.1 is restricted in the sense that it only considers vectors that belong to the canonical basis for the nullspace corresponding to some permutation π of the columns of the original $m \times n$ matrix A. This is a finite set of vectors, and so there are infinitely many vectors that cannot be obtained in this way. Suppose $\dim N(A) = d$ and that we have a basis $X^{(1)}, \ldots, X^{(d)}$ of $N(A)$. Every other basis of $N(A)$ consists of the columns of a matrix of the form CX where C is an invertible $d \times d$ matrix and X is the $d \times n$ matrix which has $X^{(j)}$ as row j for $1 \leq j \leq n$. Finding the optimal basis in the absolute sense would require searching over the infinite set of all matrices C. An attempt to develop an algorithm based on this approach using random elementary matrices (which generate all invertible matrices) produced an algorithm which was orders of magnitude less efficient than the algorithm described in this paper. Designing an efficient algorithm using elementary matrices to search over all possible bases of the nullspace is an important open problem.

6 First example: a random matrix of size 5×10

The first example is a matrix small enough that we can do an exhaustive search over all permutations of the columns.

We use a 5×10 matrix with signed single digits as its entries. As a source of pseudorandom numbers we use the first 50 digits of the expansion of $\sqrt{2}/2$ in base 19: for digits $0 \leq x \leq 9$ the matrix entry is x, and for digits $10 \leq x \leq 18$ the matrix entry is $x - 19$; see Table 1. For this example we take a population size of $p = 1$: the algorithm converges so quickly that taking a larger population size is unnecessary. The row canonical form of the matrix is displayed in Table 2. The basis for the nullspace of dimension $d = 5$ obtained from the row canonical form by Algorithms 7 and 10 is given in Table 3. The largest component (in absolute value) is 17423.

A random permutation generated using Algorithm 12 is $[2, 10, 3, 9, 7, 6, 4, 8, 5, 1]$. The corresponding permuted matrix is given in Table 4. As before we compute the row canonical form of this matrix, and then use Algorithms 7 and 10 to obtain a basis for the nullspace. However, whenever we permute the columns of the original matrix, we must unpermute the components of the resulting basis vectors. The basis we obtain after doing this is displayed in Table 5.

We now merge and sort the bases from Tables 3 and 5, and put the resulting $2d = 10$ vectors into the columns of a matrix; see Table 6. The row canonical form of this matrix has 5 nonzero rows: the leading ones of the nonzero rows occur in the first 5 columns. Therefore the better basis selected by Algorithm 15 consists of the first 5 vectors in the merged and sorted list; see Table 7. This better basis contains four vectors from the old basis of Table 3

$$\begin{bmatrix} -6 & 8 & 5 & 0 & -3 & 7 & 9 & 3 & 0 & 0 \\ 2 & 4 & 0 & -2 & -2 & 5 & 1 & -5 & -9 & -3 \\ 3 & 1 & 0 & 9 & 6 & -6 & -3 & 5 & -5 & 9 \\ 4 & 8 & 0 & -2 & 0 & 5 & -3 & -3 & 7 & 0 \\ -7 & 8 & 4 & 7 & -3 & 5 & 1 & 3 & -6 & 9 \end{bmatrix}$$

Table 1: Pseudorandom 5×10 matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \frac{857}{1520} & \frac{3089}{1520} & -\frac{2141}{1520} & -\frac{12447}{1520} & -\frac{495}{304} \\ 0 & 1 & 0 & 0 & 0 & \frac{489}{1520} & -\frac{2127}{1520} & \frac{403}{1520} & \frac{6161}{1520} & \frac{297}{304} \\ 0 & 0 & 1 & 0 & 0 & \frac{67}{80} & \frac{459}{80} & -\frac{31}{80} & -\frac{917}{80} & -\frac{45}{16} \\ 0 & 0 & 0 & 1 & 0 & -\frac{13}{152} & -\frac{5}{152} & -\frac{39}{152} & -\frac{557}{152} & \frac{99}{152} \\ 0 & 0 & 0 & 0 & 1 & -\frac{367}{304} & -\frac{375}{304} & \frac{571}{304} & \frac{2457}{304} & \frac{357}{304} \end{bmatrix}$$

Table 2: Row canonical form of 5×10 matrix

$$\begin{bmatrix} -857 & -489 & -1273 & 130 & 1835 & 1520 & 0 & 0 & 0 & 0 \\ -3089 & 2127 & -8721 & 50 & 1875 & 0 & 1520 & 0 & 0 & 0 \\ 2141 & -403 & 589 & 390 & -2855 & 0 & 0 & 1520 & 0 & 0 \\ 12447 & -6161 & 17423 & 5570 & -12285 & 0 & 0 & 0 & 1520 & 0 \\ 495 & -297 & 855 & -198 & -357 & 0 & 0 & 0 & 0 & 304 \end{bmatrix}$$

Table 3: Old basis for nullspace of 5×10 matrix

and one vector from the new basis of Table 5. The largest component (in absolute value) is now 8721.

We now repeat this process, running the evolutionary algorithm on the same 5×10 matrix for 100000 generations. The algorithm converged after only 20 generations. The permutations which resulted in a better basis are displayed in Table 8; these permutations were generated by Algorithm 12 using the `rand()` function from Maple. The final basis appears in Table 9. The largest component (in absolute value) is 352.

To compare the results of the evolutionary algorithm with the optimal basis, we ran an exhaustive search over all $10! = 3628800$ permutations of the columns. The algorithm for this exhaustive search is essentially the same as that described in Section 3, except that the loop over g in the main procedure is replaced by a loop over all permutations, and we do not call Algorithm 12. We initialize the permutation to the identity, and at the end of each iteration we use the standard algorithm to compute the next permutation in lexicographical order; see

$$\begin{bmatrix} 8 & 0 & 5 & 0 & 9 & 7 & 0 & 3 & -3 & -6 \\ 4 & -3 & 0 & -9 & 1 & 5 & -2 & -5 & -2 & 2 \\ 1 & 9 & 0 & -5 & -3 & -6 & 9 & 5 & 6 & 3 \\ 8 & 0 & 0 & 7 & -3 & 5 & -2 & -3 & 0 & 4 \\ 8 & 9 & 4 & -6 & 1 & 5 & 7 & 3 & -3 & -7 \end{bmatrix}$$

Table 4: Permuted 5×10 matrix

$$\begin{bmatrix} 0 & -3333 & -64386 & 0 & 0 & 16377 & 25995 & 0 & 3252 & 21760 \\ 0 & -317 & 6971 & 5459 & 0 & 0 & -3591 & 0 & 383 & -6408 \\ 0 & 2082 & 24270 & 0 & 0 & 0 & -20793 & 16377 & -4272 & -18634 \\ 0 & -9504 & 86868 & 0 & 16377 & 0 & -34353 & 0 & -3861 & -23458 \\ 5459 & -5687 & 33101 & 0 & 0 & 0 & -9695 & 0 & -775 & -4850 \end{bmatrix}$$

Table 5: New basis for nullspace of 5×10 matrix

$$\begin{bmatrix} 495 & -857 & 2141 & 0 & -3089 & 12447 & 0 & 5459 & 0 & 0 \\ -297 & -489 & -403 & -317 & 2127 & -6161 & 2082 & -5687 & -3333 & -9504 \\ 855 & -1273 & 589 & 6971 & -8721 & 17423 & 24270 & 33101 & -64386 & 86868 \\ -198 & 130 & 390 & 5459 & 50 & 5570 & 0 & 0 & 0 & 0 \\ -357 & 1835 & -2855 & 0 & 1875 & -12285 & 0 & 0 & 0 & 16377 \\ 0 & 1520 & 0 & 0 & 0 & 0 & 0 & 0 & 16377 & 0 \\ 0 & 0 & 0 & -3591 & 1520 & 0 & -20793 & -9695 & 25995 & -34353 \\ 0 & 0 & 1520 & 0 & 0 & 0 & 16377 & 0 & 0 & 0 \\ 0 & 0 & 0 & 383 & 0 & 1520 & -4272 & -775 & 3252 & -3861 \\ 304 & 0 & 0 & -6408 & 0 & 0 & -18634 & -4850 & 21760 & -23458 \end{bmatrix}$$

Table 6: Merged and sorted bases (vectors are columns)

$$\begin{bmatrix} 495 & -297 & 855 & -198 & -357 & 0 & 0 & 0 & 0 & 304 \\ -857 & -489 & -1273 & 130 & 1835 & 1520 & 0 & 0 & 0 & 0 \\ 2141 & -403 & 589 & 390 & -2855 & 0 & 0 & 1520 & 0 & 0 \\ 0 & -317 & 6971 & 5459 & 0 & 0 & -3591 & 0 & 383 & -6408 \\ -3089 & 2127 & -8721 & 50 & 1875 & 0 & 1520 & 0 & 0 & 0 \end{bmatrix}$$

Table 7: Better basis for nullspace of 5×10 matrix

generation	maximum	permutation
0	17423	–
1	8721	$[2, 10, 3, 9, 7, 6, 4, 8, 5, 1]$
2	1515	$[6, 1, 3, 8, 5, 2, 4, 7, 10, 9]$
8	855	$[8, 10, 4, 1, 5, 2, 9, 6, 7, 3]$
19	472	$[7, 3, 6, 5, 2, 8, 9, 1, 4, 10]$
20	352	$[8, 3, 9, 6, 4, 1, 10, 5, 2, 7]$

Table 8: Results of evolutionary algorithm for 5×10 matrix

$$\begin{bmatrix} 31 & 7 & 29 & 0 & -55 & -30 & 0 & 10 & 0 & 0 \\ -27 & 0 & -63 & 0 & 39 & 27 & 9 & 0 & 0 & 4 \\ 0 & 2 & 175 & 13 & 0 & -81 & 0 & -108 & 13 & 0 \\ 28 & 0 & 0 & 190 & -73 & 0 & 0 & -17 & 31 & -124 \\ 118 & -99 & 352 & 0 & -50 & 25 & -65 & 0 & 0 & 0 \end{bmatrix}$$

Table 9: Final basis for nullspace of 5×10 matrix

Roberts and Tesman [18] (Section 2.16.1). This allows us to avoid using unnecessary time and space generating all permutations at once. This exhaustive search produces the optimal basis, which is the same as the basis for the nullspace displayed in Table 9 up to a change of sign in the last two vectors.

In conclusion, the evolutionary algorithm with a population size of $p = 1$ achieved the optimal basis after only 20 iterations; that is, after only $20/10! \approx .00055\,\%$ of the number of iterations needed to guarantee optimality.

7 Second example: a random matrix of size 10×20

The second example is a matrix large enough that we can do a meaningful comparison of different population sizes.

We use a matrix of size 10×20; since $20! \approx 2.4329 \times 10^{18}$, in this case it is not practical (without special hardware and software) to do an exhaustive search over all permutations of the columns. The matrix entries are the signed single digits corresponding the first 200 digits of the expansion of $\sqrt{2}/2$ in base 19; see Table 10. For the integral basis (not displayed) of the nullspace obtained by Algorithms 7 and 10, the largest component (in absolute value) is 208455376722.

We first run the evolutionary algorithm on this matrix with population $p = 1$. The results for 100000 generations are summarized in Table 11; no improvement was obtained after generation 34241. The first column gives the generation, and the second column gives the largest component (in absolute value) of the basis vectors. Only those generations are listed which produced a new basis with a strictly smaller value of the largest component of the worst vector. After 34241 generations, the largest component has decreased by a factor of over 10^5 from the initial value: the new largest component is 2067577.

$$\begin{bmatrix}
-6 & 8 & 5 & 0 & -3 & 7 & 9 & 3 & 0 & 0 & 2 & 4 & 0 & -2 & -2 & 5 & 1 & -5 & -9 & -3 \\
3 & 1 & 0 & 9 & 6 & -6 & -3 & 5 & -5 & 9 & 4 & 8 & 0 & -2 & 0 & 5 & -3 & -3 & 7 & 0 \\
-7 & 8 & 4 & 7 & -3 & 5 & 1 & 3 & -6 & 9 & -1 & 5 & -2 & -2 & 4 & -2 & 0 & 8 & -4 & 3 \\
-7 & 8 & -4 & 4 & -8 & -4 & -2 & -3 & 6 & -3 & 0 & 8 & -6 & 0 & 3 & -5 & 6 & -3 & 6 & -2 \\
9 & 8 & -6 & -6 & 3 & -1 & 9 & 5 & -5 & 8 & -7 & 0 & 8 & -8 & 3 & 7 & -5 & 5 & 1 & 3 \\
-8 & -9 & -1 & 8 & 5 & -5 & 3 & 4 & -8 & 5 & 8 & -2 & 4 & 8 & -5 & -5 & -6 & -8 & 7 & -6 \\
-7 & -6 & -2 & -8 & -4 & -9 & -5 & 4 & -7 & 7 & -9 & 3 & 4 & 0 & -5 & -8 & 0 & -7 & 7 & 6 \\
2 & -8 & 7 & 4 & -6 & 0 & 6 & 9 & 9 & -5 & 8 & 7 & -4 & -1 & 9 & -7 & 3 & -1 & 0 & -7 \\
6 & -4 & -4 & 3 & 8 & -6 & 6 & -4 & -8 & 0 & 5 & 9 & -2 & 4 & 8 & 2 & 7 & -9 & -4 & -9 \\
-2 & 5 & -5 & -5 & -7 & 3 & -7 & 8 & 4 & 6 & -7 & -3 & 7 & 7 & -2 & -7 & 9 & 7 & 5 & 5
\end{bmatrix}$$

Table 10: Pseudorandom 10×20 matrix

generation	maximum	generation	maximum	generation	maximum
0	208455376722	1	63578855455	2	22720150794
3	15708086137	4	9841681986	8	4444976578
14	3723159899	15	2384366358	20	2350233776
22	2130329722	29	1809304695	34	1789414998
37	1530548119	44	1445947143	47	1415727414
54	1321394459	56	707959226	60	549051774
63	481247058	71	450743010	85	434802080
90	434525061	95	363821570	111	296208307
114	272166271	131	258599554	165	253506510
181	253177933	184	249091091	220	230114216
282	203188985	291	174886436	314	144987051
324	138680570	346	136272332	396	126277794
408	115348318	563	101428305	569	82769771
582	72318520	592	53768235	628	46718334
776	41377688	789	28556541	793	27850890
1443	26728438	1474	22461063	1632	22232106
1638	21494504	1730	20034769	1852	11043245
2823	10759335	2842	10236123	3202	6987978
3277	6719919	3747	6106066	3961	5336212
6012	5088816	7880	3398467	8948	3067510
15403	3054616	17491	2960966	21540	2827883
22839	2782883	23596	2546833	25830	2248572
28295	2226994	31533	2210785	34241	2067577

Table 11: Results of evolutionary algorithm for 10×20 matrix

Using the logarithmic variables $x = \ln g$ and $y = \ln m$ where g is the generation and m is the maximum absolute value of the components, we can do a least squares fit of a linear function $y = ax + b$ to the data points in Table 11. We obtain the graph in Figure 1 where the constants in the linear equation are $a \approx -1.014312$ and $b \approx 24.51393$. This shows that

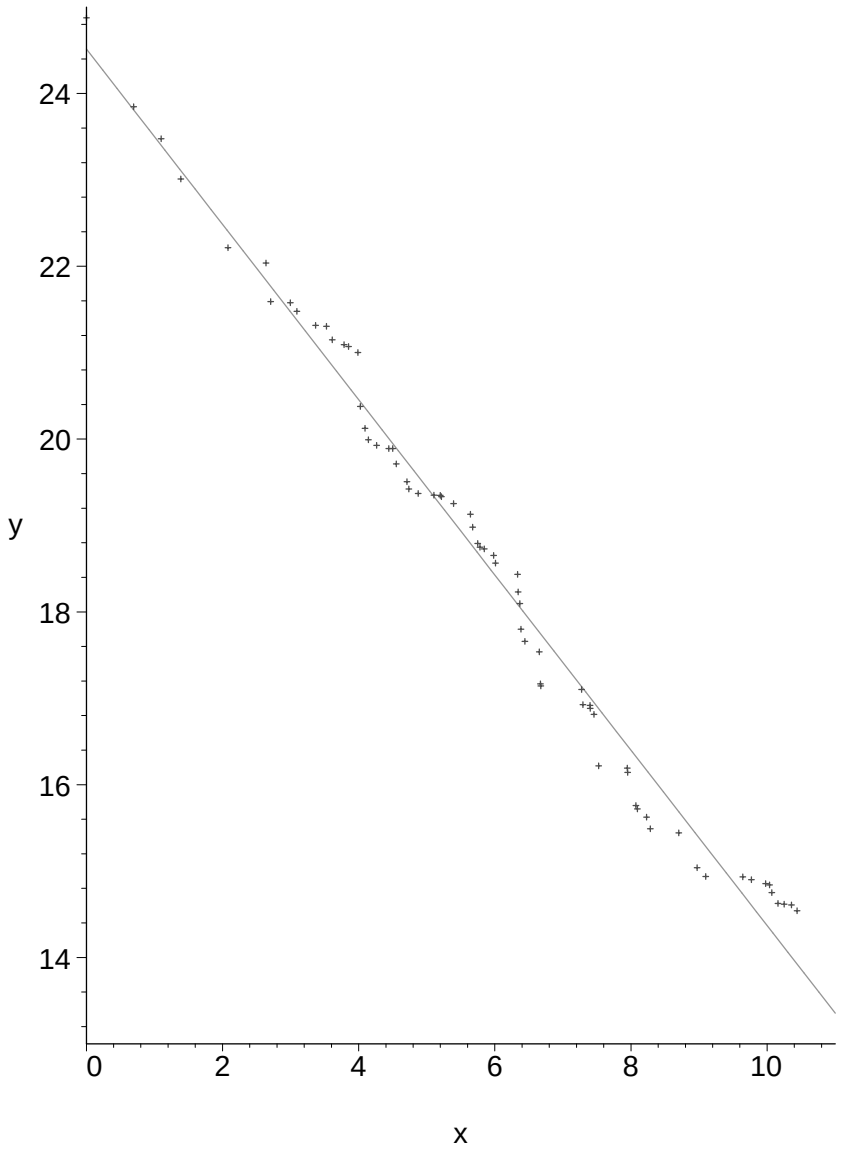

Figure 1: Least squares fit for data from Table 11

the performance of the algorithm is described very accurately by the formula

$$m = \frac{M}{g}, \quad \text{where} \quad M = e^b \approx 44285845130.$$

Figure 1 has the interesting property that the data points produced by the algorithm oscillate above and below the approximating line.

We also ran the evolutionary algorithm on this matrix with the population size p ranging from 2 to 10. For each p, the algorithm was executed for $\lceil 100000/p \rceil$ generations, so that a total of 100000 random permutations of the columns of the original matrix were used. Table 12 gives the values of the Maple kernel variables `cputime`, `bytesalloc`, `bytesused` at the end of each computation. These computations were done using Maple 8.01 (version for IBM

population	cputime	bytesalloc	bytesused
1	13978.263	6093732	288378814360
2	13811.107	6093732	264609479520
3	13446.455	6093732	256152032796
4	13689.235	6224780	252958183472
5	14261.425	6290304	250580748720
6	14432.870	6355828	249179014896
7	14809.953	6421352	248135693388
8	15236.573	6617924	247880343164
9	15916.828	6814496	247384667764
10	16095.313	7011068	246946519608

Table 12: Time and space data for 10×20 matrix with $1 \leq p \leq 10$

INTEL NT) on an IBM Thinkpad T43 with an Intel Pentium M 760 processor (2.0 GHz), 2 MB cache, and 1 GB RAM. The total CPU time decreases from $p = 1$ to $p = 3$ and then starts to increase. The total bytes allocated increases over the entire range of population sizes, and the total bytes used decreases. These data suggest that (at least for this matrix) the optimal population size is $p = 3$ (and that $p = 4$ is better than $p = 2$).

8 Third example: the expansion matrix of size 120×90 for a nonassociative operation

This section applies the evolutionary algorithm to simplify the computational results from a research project on polynomial identities for nonassociative algebras. Other related examples and applications can be found in Bremner and Peresi [8, 9]. Further information about the mathematical motivation is given in the Appendix of the present paper.

This example uses a sparse structured matrix of size 120×90. To give a precise definition of this matrix, we first consider the trilinear operation

$$[a, b, c] = 2abc + 2acb - bac - bca + 2cab + 2cba. \tag{1}$$

This operation satisfies the symmetry condition

$$[a, b, c] = [c, b, a].$$

It is not difficult to check that this is the only polynomial identity of degree 3 satisfied by operation (1).

To study polynomial identities in degree 5 (for a trilinear operation only odd degrees occur), we first consider the 3 distinct association types:

$$[[-,-,-],-,-], \qquad [-,[-,-,-],-], \qquad [-,-,[-,-,-]].$$

Here only the arrangement of brackets is important, not the permutation of the letters, so we indicate the arguments using a minus sign as a placeholder. In principle, there are $5! = 120$

monomials for each association type. However, the symmetry condition in degree 3 shows that these 360 monomials can be reduced to only 90: since

$$[[a, b, c], d, e] = [[c, b, a], d, e],$$

there are only $5!/2 = 60$ inequivalent monomials in the first type; since

$$[a, [b, c, d], e] = [a, [d, c, b], e] = [e, [b, c, d], a] = [e, [d, c, b], a],$$

there are only $5!/4 = 30$ inequivalent monomials of the second type; and since

$$[[a, b, c], d, e] = [e, d, [a, b, c]],$$

every monomial of the third type is equivalent to a monomial of the first type. We order these nonassociative monomials first by association type and then (within each association type) by lexicographical order of the permutation of the arguments. The complete list of 90 ordered nonassociative monomials appears in Table 13. We order the 120 associative monomials (the permutations of the 5 letters) lexicographically.

When each of the nonassociative monomials is expanded using formula (1), it produces a sum of 36 terms with coefficients from the set $\{1, -2, 4\}$. The expansions of the first nonassociative monomial in each association type are displayed in Table 14. Each of the terms in these expansions is a scalar multiple of one of the associative monomials.

Definition 19. *We define the* ***expansion matrix*** E *to be the* 120×90 *matrix in which the rows are labeled by the associative monomials and the columns are labeled by the nonassociative monomials, and in which the* (i, j) *entry is the coefficient of the* i*-th associative monomial in the expansion of the* j*-th nonassociative monomial. That is, each column represents the expansion of the corresponding nonassociative monomial as a linear combination of the associative monomials.*

The expansion matrix is a sparse integer matrix: in each column only 36 of the 120 entries are nonzero, giving a fill rate of 30%. The positions of the nonzero entries are displayed in Figure 2 (plus signs represent nonzero entries, spaces represent zero entries). The coefficients of the expansions in Table 14 are the nonzero entries in columns 1 and 61 of Figure 2.

The nonzero vectors in the nullspace of the expansion matrix represent linear combinations of column labels (nonassociative monomials) which evaluate to zero when expanded using formula (1); that is, these nullspace vectors represent the non-trivial polynomial identities in degree 5 satisfied by the nonassociative trilinear operation $[a, b, c]$. The maximum components (in absolute value) and the number of nonzero components for the original 20 basis vectors computed by Algorithms 7 and 10 are displayed in columns 2 and 3 of Table 15: the smallest maximum component is 19 and the largest is 129; the smallest number of nonzero components is 66 and the largest is 71. Column 4 of Table 15 gives the dimension of the subspace of the nullspace generated by the identity: we apply all 5! permutations of the 5 letters to the identity and obtain a total of 120 identities; we put these identities in the rows of a matrix of size 120×90; we then compute the rank of this matrix, which is the dimension of the subspace generated by the identity under the action of the symmetric group S_5. Every one of the original 20 identities generates the entire nullspace.

$$
\begin{array}{llllll}
[[abc]de], & [[abc]ed], & [[abd]ce], & [[abd]ec], & [[abe]cd], & [[abe]dc], \\
[[acb]de], & [[acb]ed], & [[acd]be], & [[acd]eb], & [[ace]bd], & [[ace]db], \\
[[adb]ce], & [[adb]ec], & [[adc]be], & [[adc]eb], & [[ade]bc], & [[ade]cb], \\
[[aeb]cd], & [[aeb]dc], & [[aec]bd], & [[aec]db], & [[aed]bc], & [[aed]cb], \\
[[bac]de], & [[bac]ed], & [[bad]ce], & [[bad]ec], & [[bae]cd], & [[bae]dc], \\
[[bcd]ae], & [[bcd]ea], & [[bce]ad], & [[bce]da], & [[bdc]ae], & [[bdc]ea], \\
[[bde]ac], & [[bde]ca], & [[bec]ad], & [[bec]da], & [[bed]ac], & [[bed]ca], \\
[[cad]be], & [[cad]eb], & [[cae]bd], & [[cae]db], & [[cbd]ae], & [[cbd]ea], \\
[[cbe]ad], & [[cbe]da], & [[cde]ab], & [[cde]ba], & [[ced]ab], & [[ced]ba], \\
[[dae]bc], & [[dae]cb], & [[dbe]ac], & [[dbe]ca], & [[dce]ab], & [[dce]ba], \\
[a[bcd]e], & [a[bce]d], & [a[bdc]e], & [a[bde]c], & [a[bec]d], & [a[bed]c], \\
[a[cbd]e], & [a[cbe]d], & [a[cde]b], & [a[ced]b], & [a[dbe]c], & [a[dce]b], \\
[b[acd]e], & [b[ace]d], & [b[adc]e], & [b[ade]c], & [b[aec]d], & [b[aed]c], \\
[b[cad]e], & [b[cae]d], & [b[dae]c], & [c[abd]e], & [c[abe]d], & [c[adb]e], \\
[c[aeb]d], & [c[bad]e], & [c[bae]d], & [d[abc]e], & [d[acb]e], & [d[bac]e].
\end{array}
$$

Table 13: The ordered list of 90 nonassociative monomials (column labels)

$$
\begin{aligned}
&[[a,b,c],d,e] = \\
&\qquad 4abcde + 4abced - 2dabce - 2deabc + 4eabcd + 4edabc \\
&\quad + 4acbde + 4acbed - 2dacbe - 2deacb + 4eacbd + 4edacb \\
&\quad - 2bacde - 2baced \;\; + dbace \;\; + debac - 2ebacd - 2edbac \\
&\quad - 2bcade - 2bcaed \;\; + dbcae \;\; + debca - 2ebcad - 2edbca \\
&\quad + 4cabde + 4cabed - 2dcabe - 2decab + 4ecabd + 4edcab \\
&\quad + 4cbade + 4cbaed - 2dcbae - 2decba + 4ecbad + 4edcba
\end{aligned}
$$

$$
\begin{aligned}
&[a,[b,c,d],e] = \\
&\qquad 4abcde + 4aebcd - 2bcdae - 2bcdea + 4eabcd + 4ebcda \\
&\quad + 4abdce + 4aebdc - 2bdcae - 2bdcea + 4eabdc + 4ebdca \\
&\quad - 2acbde - 2aecbd \;\; + cbdae \;\; + cbdea - 2eacbd - 2ecbda \\
&\quad - 2acdbe - 2aecdb \;\; + cdbae \;\; + cdbea - 2eacdb - 2ecdba \\
&\quad + 4adbce + 4aedbc - 2dbcae - 2dbcea + 4eadbc + 4edbca \\
&\quad + 4adcbe + 4aedcb - 2dcbae - 2dcbea + 4eadcb + 4edcba
\end{aligned}
$$

Table 14: Expansions of the first monomial in each association type

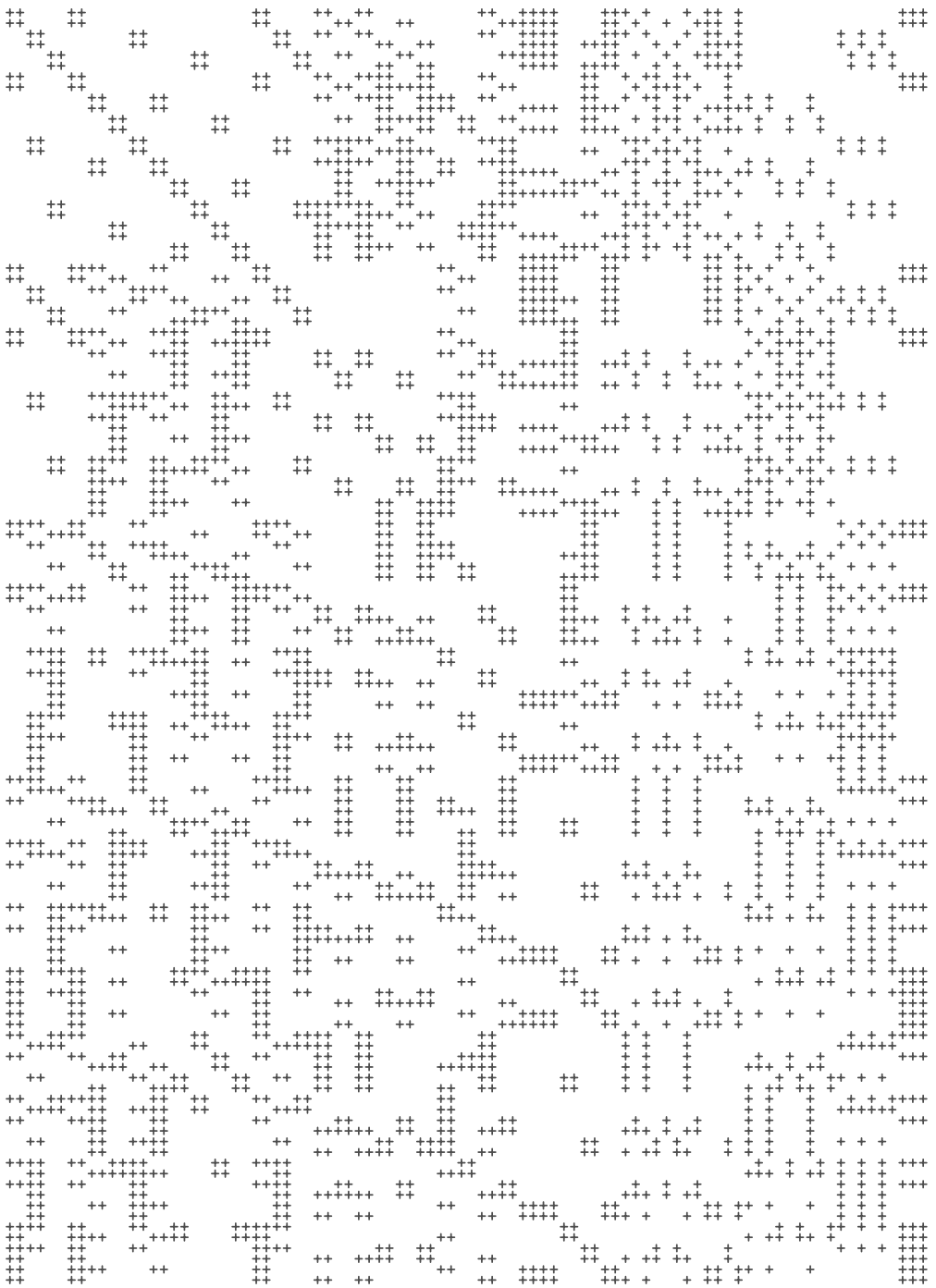

Figure 2: Nonzero entries in the expansion matrix

	original nullspace basis			final nullspace basis		
	maximum	nonzero	dimension	maximum	nonzero	dimension
1	19	68	20	1	24	5
2	27	69	20	1	24	5
3	29	66	20	1	24	5
4	29	66	20	1	24	5
5	36	71	20	1	24	5
6	42	66	20	3	42	6
7	42	66	20	3	42	6
8	51	71	20	3	42	6
9	59	67	20	3	42	6
10	67	66	20	3	42	6
11	67	66	20	3	42	6
12	76	68	20	4	48	20
13	76	68	20	5	42	20
14	76	70	20	5	42	20
15	76	70	20	5	42	20
16	77	71	20	5	42	20
17	77	71	20	5	42	20
18	83	69	20	5	42	20
19	129	70	20	5	42	20
20	129	70	20	5	42	20

Table 15: Goodness data for original and final basis vectors

generation	maximum	generation	maximum
0	129	84	83
98	77	104	76
271	67	466	59
526	42	883	36
892	30	1014	29
1920	27	1937	19
2709	15	2746	10
2921	6	64752	5

Table 16: Performance of the algorithm on the expansion matrix

The original 20 basis vectors for the nullspace are displayed in Tables 20 and 21. These results are quite bad; they represent polynomial identities that are so complicated as to be useless for the development of a structure theory of algebras. The polynomial identity corresponding to the best of these initial basis vectors is displayed in Table 17; here we have omitted the superfluous commas between the arguments of the trilinear operation.

We now run the evolutionary algorithm on the expansion matrix for 100000 generations with a population size of $p = 1$. A list of the generations and the maximum components of the worst vectors appears in Table 16. This table only lists those generations at which a decrease in the maximum component of the worst vector occurred. At the other generations, an improvement in the nullspace basis was also possible: the algorithm may have found better basis vectors, but the maximum component did not decrease. An interesting feature of the data in Table 16 is the last row: maximum component 6 was obtained at generation 2921, and then no further improvement was observed until generation 64752 when maximum component 5 was achieved. The final basis obtained is displayed in Tables 22 and 23. The maximum components (in absolute value) and the number of nonzero components for these final 20 basis vectors are displayed in columns 5 and 6 of Table 15: the smallest maximum component is 1 and the largest is 5; the smallest number of nonzero components is 24 and the largest is 48. There has been a dramatic increase in the goodness of the vectors between the original basis and the final basis. However, from column 7 of Table 15 we see that only the last 9 of the final basis vectors generate the entire nullspace.

We now study final vector number 12 in more detail; this is the second group of 6 rows in Table 23. It has 48 nonzero components with a maximum absolute value of 4. Recalling the correspondence between nonassociative monomials and columns of the expansion matrix, we see that this vector represents the polynomial identity displayed in Table 18. This is the first identity from the final basis which produces a dimension of 20: it is the best identity that generates the entire nullspace. As can be seen from column 7 of Table 15, each of the better identities generates a proper subspace of the nullspace. Identity number 12 has an interesting combinatorial structure. Half the coefficients are positive and half are negative; the coefficients occur with the following multiplicities in the two association types:

coefficient	$+2$	-2	$+3$	-3	$+4$	-4
type 1	12	6	0	3	6	6
type 2	3	9	3	0	0	0

The identity is invariant under all 6 permutations of b, c, d. This means that we can set $b = c = d$ and obtain an identity which is not multilinear but which is equivalent to the original identity over a field of characteristic 0 (see Chapter 1 of Zhevlakov, Slinko, Shestakov and Shirshov [21] for a detailed discussion of linearization of nonassociative polynomials). This delinearization process gives the equivalent but much more compact identity displayed in Table 19.

Theorem 20. *Every polynomial identity of degree 5 for the trilinear operation (1), over a field of characteristic 0, follows from the identity* $[a, b, c] = [c, b, a]$ *in degree 3 and the 10-term identity of Table 19 in degree 5.*

Proof. In the 48-term identity of Table 18 we replace c and d by b (and then we replace e by c for notational convenience). We then collect terms using the symmetry condition in degree 3 and cancel common factors in the coefficients. □

$$
\begin{aligned}
&3[[abc]de] + 3[[abc]ed] + 11[[abd]ce] + 7[[abd]ec] + 11[[abe]cd] \\
&+ 7[[abe]dc] + 15[[acb]de] + 15[[acb]ed] - 3[[acd]be] - 9[[acd]eb] \\
&- 3[[ace]bd] - 9[[ace]db] + 17[[adb]ce] + 19[[adb]ec] + 3[[adc]be] \\
&- 9[[adc]eb] + 3[[ade]bc] - 3[[ade]cb] + 17[[aeb]cd] + 19[[aeb]dc] \\
&+ 3[[aec]bd] - 9[[aec]db] + 3[[aed]bc] - 3[[aed]cb] - 3[[bac]de] \\
&- 3[[bac]ed] - 7[[bad]ce] - 11[[bad]ec] - 7[[bae]cd] - 11[[bae]dc] \\
&+ 8[[bcd]ae] - 15[[bcd]ea] + 8[[bce]ad] - 15[[bce]da] + 8[[bdc]ae] \\
&- 3[[bdc]ea] + 4[[bde]ac] - 13[[bde]ca] + 8[[bec]ad] - 3[[bec]da] \\
&+ 4[[bed]ac] - 13[[bed]ca] - 3[[cad]be] - 3[[cad]eb] - 3[[cae]bd] \\
&- 3[[cae]db] - 4[[cbd]ae] + 9[[cbd]ea] - 4[[cbe]ad] + 9[[cbe]da] \\
&+ 9[[cde]ba] + 9[[ced]ba] - 15[[dae]bc] - 3[[dae]cb] - 8[[dbe]ac] \\
&+ 5[[dbe]ca] - 9[[dce]ba] - 14[a[bcd]e] - 14[a[bce]d] - 8[a[bdc]e] \\
&- 12[a[bde]c] - 8[a[bec]d] - 12[a[bed]c] + 10[a[cbd]e] + 10[a[cbe]d] \\
&+ 6[b[acd]e] + 6[b[ace]d] + 6[d[abc]e]
\end{aligned}
$$

Table 17: The best polynomial identity from the original nullspace basis

$$
\begin{aligned}
&4[[abe]cd] + 4[[abe]dc] + 4[[ace]bd] + 4[[ace]db] + 4[[ade]bc] \\
&+ 4[[ade]cb] + 2[[aeb]cd] + 2[[aeb]dc] + 2[[aec]bd] + 2[[aec]db] \\
&+ 2[[aed]bc] + 2[[aed]cb] + 2[[bae]cd] - 2[[bae]dc] + 2[[bce]ad] \\
&- 4[[bce]da] + 2[[bde]ac] - 4[[bde]ca] - 3[[bec]ad] - 3[[bed]ac] \\
&- 2[[cae]bd] - 2[[cae]db] + 2[[cbe]ad] - 4[[cbe]da] + 2[[cde]ab] \\
&- 4[[cde]ba] - 3[[ced]ab] - 2[[dae]bc] - 2[[dae]cb] + 2[[dbe]ac] \\
&- 4[[dbe]ca] + 2[[dce]ab] - 4[[dce]ba] - 2[a[bce]d] - 2[a[bde]c] \\
&+ 2[a[bec]d] + 2[a[bed]c] - 2[a[cbe]d] - 2[a[cde]b] + 2[a[ced]b] \\
&- 2[a[dbe]c] - 2[a[dce]b] - 2[b[aec]d] - 2[b[aed]c] + 3[b[cae]d] \\
&+ 3[b[dae]c] - 2[c[aeb]d] + 3[c[bae]d]
\end{aligned}
$$

Table 18: The best polynomial identity from the final nullspace basis

$$
\begin{aligned}
&8[[abc]bb] + 4[[acb]bb] - 4[[bac]bb] + 4[[bbc]ab] - 8[[bbc]ba] \\
&- 3[[bcb]ab] - 4[a[bbc]b] + 2[a[bcb]b] - 2[b[acb]b] + 3[b[bac]b]
\end{aligned}
$$

Table 19: The compact form of the polynomial identity from Table 18

The evolutionary algorithm has proved to be very effective in this application to nonassociative algebra: we have gone from the very complicated identities of Tables 20 and 21 to the much simpler identity of Table 19, and the identity of Table 19 implies all the identities of Tables 20 and 21.

9 Further remarks

This algorithm can be adapted to fields of positive characteristic. Let $\mathbb{F}_p$ denote the field with p elements. Every nonzero congruence class modulo p has a unique best rational approximation: for $1 \leq r \leq p-1$ we find the unique pair of integers a, b satisfying the conditions:

$$r \equiv \frac{a}{b} \pmod{p}, \quad b > 0, \quad \gcd(a,b) = 1, \quad \max(|a|, b) \text{ is minimal.}$$

We then define a function $v\colon \mathbb{F}_p \to \mathbb{Q}$ by $v(0) = 0$ and $v(r) = a/b$ for $r \neq 0$. Given a vector $X = (x_1, \ldots, x_n) \in \mathbb{F}_p^n$, we define its max norm $\|X\|_\infty$ as follows. We replace each component $x_i \in \mathbb{F}_p$ by $v(x_i) \in \mathbb{Q}$, obtaining a vector $X' \in \mathbb{Q}^n$. We apply Algorithm 10 to X' to get a good vector X'' with integral components. We then define $\|X\|_\infty = \|X''\|_\infty$ for the purposes of the vector comparison in Algorithm 13. Our motivation for this method of comparing vectors over a finite field is as follows. When we use rational arithmetic to compute the row canonical form of a large sparse matrix with small integer entries, during the intermediate stages of the computation the matrix entries can have extremely large numerators and denominators. In order to control memory allocation, we do such computations over a finite field; the congruence classes represent the rational numbers that would arise if we were doing the same calculation in characteristic 0. To measure the absolute value of a congruence class, we replace it by its best rational approximation.

Acknowledgements

This work was inspired by a paper by Irvin R. Hentzel [14] and by joint work with Luiz A. Peresi [8, 9]. Hentzel's paper gives an evolutionary algorithm using linear algebra for finding a "good proof" of a polynomial identity that is already known to be true. In contrast, the algorithm of the present paper can be used for the initial discovery of a "good identity". I thank Daniel Ashlock, Mikelis Bickis, Irvin Hentzel, Michael Horsch, Luiz Peresi and Christine Soteros for making very helpful suggestions on earlier versions of this paper. This work was presented at the Bioinformatics Seminar at the University of Saskatchewan in January 2006, and at the Seminar on Nonassociative Structures and their Identities at the University of Sao Paulo in February 2006. I thank the organizers of those seminars for the opportunity to speak, and the participants for their comments. I also gratefully acknowledge the hospitality of the Institute for Mathematics and Statistics at the University of Sao Paulo during my visits there in August 2005 and February 2006. This research was supported by a Discovery Grant from NSERC (Natural Sciences and Engineering Research Council of Canada).

3	3	11	7	11	7	15	15	−3	−9	−3	−9	17	19	3
−9	3	−3	17	19	3	−9	3	−3	−3	−3	−7	−11	−7	−11
8	−15	8	−15	8	−3	4	−13	8	−3	4	−13	−3	−3	−3
−3	−4	9	−4	9	0	9	0	9	−15	−3	−8	5	0	−9
−14	−14	−8	−12	−8	−12	10	10	0	0	0	0	6	6	0
0	0	0	0	0	0	0	0	0	0	0	0	6	0	0

9	9	9	9	9	9	27	27	2	−11	2	−11	27	27	2
−11	2	−11	27	27	2	−11	2	−11	−9	−9	−9	−9	−9	−9
9	−17	9	−17	9	−17	9	−17	9	−17	9	−17	−10	1	−10
1	−9	13	−9	13	3	6	3	6	−10	1	−9	13	3	6
−16	−16	−16	−16	−16	−16	8	8	0	0	8	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

−16	−29	9	9	−29	−16	−16	−29	9	9	−29	−16	−9	−9	−9
−9	−9	−9	−29	−16	−29	−16	9	9	8	19	9	9	19	8
1	−7	−5	16	−17	23	−17	23	−5	16	1	−7	9	9	19
8	1	−7	−5	16	−17	23	1	−7	9	9	1	−7	1	−7
0	16	24	24	16	0	0	16	24	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

−29	−16	−29	−16	9	9	−29	−16	−29	−16	9	9	−29	−16	−29
−16	9	9	−9	−9	−9	−9	−9	−9	19	8	19	8	9	9
−5	16	1	−7	−5	16	1	−7	−17	23	−17	23	19	8	9
9	−5	16	1	−7	1	−7	−17	23	9	9	1	−7	1	−7
16	0	16	0	24	24	16	0	0	24	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12	12	17	10	17	10	24	24	−1	−14	−1	−14	35	28	5
−8	3	−3	35	28	5	−8	3	−3	−6	−6	−13	−32	−13	−32
10	−36	10	−36	16	−6	−2	−25	16	−6	−2	−25	−7	4	−7
4	−2	24	−2	24	10	17	10	17	−33	−3	−20	5	4	−13
−16	−16	−16	−24	−16	−24	8	8	0	0	0	0	12	12	0
0	0	0	0	0	0	0	0	0	0	0	0	0	12	0

−3	−15	14	7	−42	−21	27	−9	25	11	−15	−9	2	−5	7
−7	−11	−34	−6	15	−9	−3	13	14	−15	−3	2	19	−6	15
−1	−1	−9	27	−19	−19	−5	2	−3	−15	19	2	−17	17	−9
−3	−7	−7	−15	−3	−7	7	17	−17	13	14	7	14	11	25
8	0	8	24	0	0	−16	0	0	0	0	0	0	−24	0
0	0	0	0	0	0	0	0	24	0	0	0	0	0	0

−15	−3	−42	−21	14	7	−9	27	−15	−9	25	11	−6	15	−9
−3	13	14	2	−5	7	−7	−11	−34	−3	−15	−6	15	2	19
−9	27	−1	−1	−3	−15	19	2	−19	−19	−5	2	−9	−3	−17
17	−15	−3	−7	−7	17	−17	−7	7	13	14	7	14	11	25
0	8	0	0	8	24	0	−16	0	0	0	0	−24	0	0
0	0	0	0	0	0	0	0	0	24	0	0	0	0	0

−21	−21	−42	−3	−42	−3	−39	−39	−19	25	−19	25	−30	−51	−1
−5	9	18	−30	−51	−1	−5	9	18	39	39	18	9	18	9
−25	21	−25	21	17	39	−21	30	17	39	−21	30	11	−17	11
−17	−7	−33	−7	−33	−23	−13	−23	−13	45	18	27	−30	7	−7
28	28	40	24	40	24	−8	−8	0	0	0	0	−12	−12	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	24

0	0	−29	2	−29	2	−18	−18	−6	30	−6	30	−17	−46	12
0	6	15	−17	−46	12	0	6	15	18	18	−11	20	−11	20
−25	28	−25	28	−7	−2	−13	13	−7	−2	−13	13	−18	−6	−18
−6	−7	−26	−7	−26	−15	−30	−15	−30	48	21	59	1	39	24
12	12	24	24	24	24	−24	−24	0	0	0	0	−12	−12	0
0	0	0	0	0	24	0	0	0	0	0	0	0	0	0

−47	29	−32	−25	56	−29	−17	−37	−17	−37	39	39	−32	−25	−47
29	−29	56	−40	−5	33	33	−5	−40	25	−43	64	47	−40	−5
67	67	−37	−17	−11	−11	−25	−32	−43	25	47	64	25	−43	33
33	−11	−11	29	−47	29	−47	−43	25	−5	−40	−25	−32	−37	−17
40	0	16	0	0	48	16	0	0	0	0	0	0	24	0
0	0	0	0	0	0	0	0	0	0	48	0	0	0	0

Table 20: Original nullspace basis vectors 1–10 for the expansion matrix

29	−47	56	−29	−32	−25	−37	−17	39	39	−17	−37	−40	−5	33
33	−5	−40	−32	−25	−47	29	−29	56	−43	25	−40	−5	64	47
−37	−17	67	67	−43	25	47	64	−11	−11	−25	−32	33	33	25
−43	29	−47	−11	−11	−43	25	29	−47	−5	−40	−25	−32	−37	−17
0	40	0	48	16	0	0	16	0	0	0	0	24	0	0
0	0	0	0	0	0	0	0	0	0	0	48	0	0	0

9	0	−45	−36	−11	−25	−57	−42	−10	16	8	19	−51	−66	2
28	−16	25	−47	−61	14	49	−22	−5	21	−12	15	48	1	35
−24	76	−38	31	−12	−8	2	37	−32	13	−4	55	20	−2	−10
1	18	−26	4	−47	0	−54	−6	−12	50	−5	32	−5	−12	30
32	40	32	24	16	48	−16	−32	0	0	0	0	−24	0	24
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	9	−11	−25	−45	−36	−42	−57	8	19	−10	16	−47	−61	14
49	−22	−5	−51	−66	2	28	−16	25	−12	21	1	35	15	48
−38	31	−24	76	−32	13	−4	55	−12	−8	2	37	−10	1	20
−2	4	−47	18	−26	−6	−12	0	−54	50	−5	32	−5	−12	30
40	32	16	48	32	24	−32	−16	0	0	0	0	0	−24	0
0	24	0	0	0	0	0	0	0	0	0	0	0	0	0

−20	−34	−25	−32	−61	−38	−56	−70	21	24	−15	18	−67	−74	−9
18	−27	−9	−67	−68	−9	24	9	27	10	44	29	46	41	40
−28	50	−16	76	−34	44	−20	65	−10	34	−8	53	15	−6	27
−12	8	−58	20	−32	−24	−21	−12	−33	45	−9	22	−37	−18	9
56	48	32	48	48	48	−16	−24	0	0	0	0	0	−24	0
0	0	0	0	0	0	24	0	0	0	0	0	0	0	0

−34	−20	−61	−38	−25	−32	−70	−56	−15	18	21	24	−67	−68	−9
24	9	27	−67	−74	−9	18	−27	−9	44	10	41	40	29	46
−16	76	−28	50	−10	34	−8	53	−34	44	−20	65	27	−12	15
−6	20	−32	8	−58	−12	−33	−24	−21	45	−9	22	−37	−18	9
48	56	48	48	32	48	−24	−16	0	0	0	0	−24	0	0
0	0	0	0	0	0	0	24	0	0	0	0	0	0	0

10	17	37	35	55	62	46	53	−9	−21	9	6	43	65	−15
−27	21	3	55	62	−3	−30	15	−3	−14	−31	−11	−61	−41	−58
39	−53	25	−74	33	−11	5	−77	13	−14	−1	−35	3	−9	−27
−6	3	55	−11	34	9	9	3	51	−57	−3	−25	13	15	−33
−40	−56	−40	−48	−32	−48	32	16	0	0	0	0	24	24	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17	10	55	62	37	35	53	46	9	6	−9	−21	55	62	−3
−30	15	−3	43	65	−15	−27	21	3	−31	−14	−41	−58	−11	−61
25	−74	39	−53	13	−14	−1	−35	33	−11	5	−77	−27	−6	3
−9	−11	34	3	55	3	51	9	9	−57	−3	−25	13	15	−33
−56	−40	−32	−48	−40	−48	16	32	0	0	0	0	24	24	0
0	0	24	0	0	0	0	0	0	0	0	0	0	0	0

−9	−9	−34	−47	−34	−47	−63	−63	−13	13	−13	13	−70	−83	5
31	−33	6	−70	−83	5	31	−33	6	27	27	26	37	26	37
−35	57	−35	57	−17	27	−23	50	−17	27	−23	50	29	7	29
7	19	−33	19	−33	−5	−19	−5	−19	39	6	13	−10	−23	11
56	56	32	48	32	48	−16	−16	0	0	0	24	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

75	15	102	33	46	53	81	117	41	−11	17	−71	78	129	35
−17	29	−74	118	125	−13	−5	5	22	−57	−69	−54	−99	−14	−31
15	−119	55	−83	9	−77	35	−86	25	−65	11	−86	−1	43	−73
−17	57	67	−47	55	−15	15	105	39	−103	−14	−61	34	−9	−27
−64	−56	−64	−48	−80	−96	32	16	0	0	0	0	24	0	0
0	0	0	48	0	0	0	0	0	0	0	0	0	0	0

15	75	46	53	102	33	117	81	17	−71	41	−11	118	125	−13
−5	5	22	78	129	35	−17	29	−74	−69	−57	−14	−31	−54	−99
55	−83	15	−119	25	−65	11	−86	9	−77	35	−86	−73	−17	−1
43	−47	55	57	67	105	39	−15	15	−103	−14	−61	34	−9	−27
−56	−64	−80	−96	−64	−48	16	32	0	0	0	0	0	24	0
0	0	0	0	48	0	0	0	0	0	0	0	0	0	0

Table 21: Original nullspace basis vectors 11–20 for the expansion matrix

$$
\begin{array}{cccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & -1 & 0 & 0 & 0 & -1 & -1 & 0 & -1 & -1 & 0 & -1 & 0 & 0 \\
-1 & 0 & 0 & 0 & -1 & -1 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0
\end{array}
$$

$$
\begin{array}{cccccccccccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
-1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 0 & -1 & -1 & 0
\end{array}
$$

$$
\begin{array}{cccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & 0 & 0 & 0 \\
0 & 0 & -1 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$$
\begin{array}{cccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0
\end{array}
$$

$$
\begin{array}{cccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & -1 & 0 & -1 & 0 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0
\end{array}
$$

$$
\begin{array}{cccccccccccccccc}
0 & 0 & -2 & 2 & 0 & 0 & 0 & 3 & 0 & 3 & 0 & 0 & -2 & 2 & 0 & 0 \\
0 & 0 & 0 & -3 & 0 & 0 & 0 & 0 & -3 & 0 & -3 & 1 & -1 & 3 & 0 & 0 \\
-1 & 1 & -3 & 0 & 2 & -2 & -2 & 2 & 3 & 0 & 1 & -1 & 0 & -3 & 0 & 0 \\
0 & 2 & -2 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 3 & -2 & 2 & -3 & 0 & 0 \\
0 & 3 & 0 & 0 & -3 & 0 & 0 & 0 & 0 & -3 & 0 & 3 & -3 & 0 & 0 & 0 \\
0 & 0 & 3 & 3 & 0 & -3 & 0 & 0 & 0 & 3 & 0 & -3 & 0 & -3 & 3 & 0
\end{array}
$$

$$
\begin{array}{cccccccccccccccc}
0 & -3 & 0 & -3 & 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 & 2 & 0 \\
-2 & 0 & 0 & 0 & 0 & 3 & 0 & 3 & 0 & 0 & 3 & 0 & 3 & 0 & 0 & 0 \\
-2 & 2 & 0 & 0 & -2 & 2 & 0 & 0 & -3 & 0 & -3 & 0 & -1 & 1 & -3 & 0 \\
0 & 1 & -1 & 3 & 0 & 2 & -2 & -1 & 1 & -3 & 0 & 3 & 0 & 2 & -2 & 0 \\
0 & 0 & 0 & 0 & 3 & 3 & 0 & -3 & 0 & 0 & -3 & 0 & 0 & 0 & 0 & 0 \\
0 & -3 & -3 & 0 & 3 & 3 & 3 & 0 & 0 & 0 & -3 & 0 & 3 & 0 & -3 & 0
\end{array}
$$

$$
\begin{array}{cccccccccccccccc}
0 & 0 & 0 & 0 & 2 & -2 & -3 & 0 & 0 & 0 & 0 & -3 & 3 & 0 & 0 & 0 \\
0 & 0 & 3 & 2 & -2 & 0 & 0 & 0 & 0 & 3 & 0 & -3 & 0 & -1 & 1 & 0 \\
3 & 0 & 1 & -1 & -3 & 0 & -1 & 1 & -2 & 2 & 2 & -2 & 0 & 0 & 0 & 0 \\
3 & 0 & 0 & -2 & 2 & -3 & 0 & 0 & 0 & 0 & -3 & 2 & -2 & 3 & 0 & 0 \\
-3 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & -3 & 0 & 3 & 0 & 0 \\
-3 & 0 & 0 & 0 & -3 & 3 & 0 & 0 & -3 & 0 & 3 & 0 & 0 & 3 & -3 & 0
\end{array}
$$

$$
\begin{array}{cccccccccccccccc}
3 & 0 & -3 & 0 & -1 & 1 & -3 & 0 & 3 & 0 & 1 & -1 & 3 & 0 & -3 & 0 \\
0 & -1 & 1 & 2 & -2 & -2 & 2 & 2 & -2 & 0 & 0 & 0 & 0 & 2 & -2 & 0 \\
0 & 0 & 0 & -3 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 \\
2 & 0 & 0 & 0 & 3 & 0 & -3 & 0 & 0 & 2 & -2 & 0 & -3 & 0 & 3 & 0 \\
0 & 3 & 0 & -3 & 0 & 0 & 0 & -3 & 3 & 0 & 3 & -3 & -3 & 0 & 3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & -3 & 0 & 0 & 0 & -3 & 3 & 0 & 0
\end{array}
$$

$$
\begin{array}{cccccccccccccccc}
0 & 0 & 3 & 0 & 3 & 0 & 0 & 0 & -3 & 0 & -3 & 0 & 0 & 0 & 0 & 0 \\
0 & -2 & 2 & 0 & 0 & 0 & 0 & -2 & 2 & 0 & 0 & -3 & 0 & -3 & 0 & 0 \\
3 & 0 & 3 & 0 & 0 & 0 & 2 & -2 & 0 & 0 & 2 & -2 & 3 & 0 & 3 & 0 \\
0 & -3 & 0 & -3 & 0 & -2 & 2 & -2 & 2 & 1 & -1 & -1 & 1 & 1 & -1 & 0 \\
-3 & -3 & 0 & 0 & 0 & 0 & 3 & 3 & 0 & 0 & 0 & 0 & 3 & 3 & 0 & 0 \\
0 & 0 & 0 & -3 & -3 & 0 & -3 & -3 & 0 & 0 & 3 & 3 & 0 & 0 & 0 & 0
\end{array}
$$

Table 22: Final nullspace basis vectors 1–10 for the expansion matrix

$$
\begin{array}{rrrrrrrrrrrrrrr}
2 & -2 & -2 & 2 & 2 & -2 & -1 & 1 & 0 & 3 & 0 & -3 & 1 & -1 & 0 \\
-3 & 0 & 3 & -1 & 1 & 0 & 3 & 0 & -3 & 2 & -2 & -2 & 2 & 2 & -2 \\
0 & 3 & 0 & -3 & 0 & -3 & 0 & 3 & 0 & 3 & 0 & -3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-3 & 3 & 3 & -3 & -3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & 3 & 3 \\
-3 & -3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$$
\begin{array}{rrrrrrrrrrrrrrr}
0 & 0 & 0 & 0 & 4 & 4 & 0 & 0 & 0 & 0 & 4 & 4 & 0 & 0 & 0 \\
0 & 4 & 4 & 2 & 2 & 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & -2 & -2 \\
0 & 0 & 2 & -4 & 0 & 0 & 2 & -4 & -3 & 0 & -3 & 0 & 0 & 0 & -2 \\
-2 & 0 & 0 & 2 & -4 & 2 & -4 & -3 & 0 & -2 & -2 & 2 & -4 & 2 & -4 \\
0 & -2 & 0 & -2 & 2 & 2 & 0 & -2 & -2 & 2 & -2 & -2 & 0 & 0 & 0 \\
0 & -2 & -2 & 0 & 3 & 3 & 0 & 0 & 0 & -2 & 0 & 3 & 0 & 0 & 0
\end{array}
$$

$$
\begin{array}{rrrrrrrrrrrrrrr}
-4 & 2 & 0 & 0 & 0 & 0 & -2 & -5 & -2 & -5 & 0 & 0 & 0 & 0 & -4 \\
2 & 0 & 0 & 0 & 0 & -2 & -2 & 0 & 0 & -4 & 2 & 0 & 0 & 0 & 0 \\
-2 & -5 & 0 & 0 & -4 & 2 & 0 & 0 & -2 & -2 & 0 & 0 & -4 & 2 & 4 \\
4 & -4 & 2 & 4 & 4 & 4 & 4 & -2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & -2 & 0 & 0 & 5 & 0 & 0 & 2 & 2 & 5 & 0 & -2 & 2 & -2 & 0 \\
0 & 5 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0
\end{array}
$$

$$
\begin{array}{rrrrrrrrrrrrrrr}
-4 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 \\
-4 & 0 & 0 & 0 & 0 & -4 & -4 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 \\
5 & 2 & 5 & 2 & -2 & 4 & 0 & 0 & -2 & 4 & 0 & 0 & 2 & 2 & 2 \\
2 & -2 & 4 & -2 & 4 & -2 & 4 & -2 & 4 & 0 & 0 & 0 & 0 & 5 & 2 \\
-2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 2 & 2 & -2 \\
0 & -2 & 0 & -5 & -5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 2 & -5
\end{array}
$$

$$
\begin{array}{rrrrrrrrrrrrrrr}
2 & -4 & 4 & 4 & -4 & 2 & 2 & -4 & 4 & 4 & -4 & 2 & -2 & -2 & -2 \\
-2 & -2 & -2 & -4 & 2 & -4 & 2 & 4 & 4 & -5 & -2 & 0 & 0 & -2 & -5 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \\
-5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 5 \\
5 & 0 & 2 & -2 & 2 & -2 & 2 & 0 & 5 & 0 & -2 & 2 & 0 & 0 & 2
\end{array}
$$

$$
\begin{array}{rrrrrrrrrrrrrrr}
2 & -4 & 0 & 0 & 0 & 0 & -5 & -2 & 0 & 0 & -2 & -5 & 0 & 0 & -2 \\
-2 & 0 & 0 & 0 & 0 & -4 & 2 & 0 & 0 & 2 & -4 & 0 & 0 & 0 & 0 \\
0 & 0 & -2 & -5 & -2 & -2 & 0 & 0 & -4 & 2 & 0 & 0 & 4 & 4 & -4 \\
2 & 4 & 4 & -4 & 2 & -2 & -2 & 4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
-2 & 2 & 5 & 0 & 0 & 0 & 2 & 0 & 5 & 2 & 0 & -2 & -2 & 2 & 5 \\
0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0
\end{array}
$$

$$
\begin{array}{rrrrrrrrrrrrrrr}
0 & 0 & 0 & 0 & -4 & 2 & 0 & 0 & 0 & 0 & -4 & 2 & 0 & 0 & 0 \\
0 & -2 & -2 & -2 & -5 & -2 & -5 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 2 \\
0 & 0 & -4 & 2 & 0 & 0 & -2 & -2 & -2 & -5 & 0 & 0 & 0 & 0 & -4 \\
2 & 0 & 0 & -4 & 2 & -2 & -2 & 0 & 0 & 4 & 4 & 4 & 4 & 4 & 4 \\
0 & 0 & 0 & 5 & 2 & -2 & 0 & 0 & 5 & -2 & 2 & 2 & 0 & 0 & 0 \\
5 & 2 & -2 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$$
\begin{array}{rrrrrrrrrrrrrrr}
4 & 4 & 2 & -4 & 2 & -4 & -2 & -2 & -2 & -2 & -2 & -2 & 2 & -4 & 4 \\
4 & -4 & 2 & 2 & -4 & 4 & 4 & -4 & 2 & 0 & 0 & -5 & -2 & -5 & -2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & -5 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 5 & 2 \\
0 & 2 & 0 & -2 & -2 & 2 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 5 & -2
\end{array}
$$

$$
\begin{array}{rrrrrrrrrrrrrrr}
2 & 2 & 2 & 2 & 2 & 2 & -4 & -4 & -2 & 4 & -2 & 4 & -4 & -4 & -2 \\
4 & -2 & 4 & -4 & -4 & -2 & 4 & -2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 2 & 5 \\
2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -2 & -2 & -2 & -5 & -5 & -2 & -2 & 2 & 2 & -5 & -2 & 2
\end{array}
$$

$$
\begin{array}{rrrrrrrrrrrrrrr}
0 & 0 & -2 & -2 & 0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 0 & 0 & -5 \\
-2 & -5 & -2 & 0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 4 & 4 & 0 & 0 \\
4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 2 & -4 & 0 \\
0 & -2 & -2 & 0 & 0 & -2 & -5 & -4 & 2 & 2 & -4 & -2 & -2 & -4 & 2 \\
2 & 0 & -2 & -2 & 0 & 2 & 5 & 0 & 2 & 0 & 5 & 0 & 0 & 0 & 2 \\
2 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & -2 & 0 & 2 & 0 & 0 & 0 & 0
\end{array}
$$

Table 23: Final nullspace basis vectors 11–20 for the expansion matrix

Appendix: mathematical motivation

This Appendix summarizes the first few sections of Bremner and Peresi [8]; see that paper for complete details and further references.

By a trilinear operation we mean any linear combination of permutations of the variables a, b, c where the coefficients are rational numbers:

$$[a,b,c] = x_1\,abc + x_2\,acb + x_3\,bac + x_4\,bca + x_5\,cab + x_6\,cba \ (x_i \in \mathbb{Q}).$$

The terms on the right side of this definition represent products in some totally associative ternary algebra; that is, a vector space V together with a trilinear map

$$V \times V \times V \to V \quad \text{with} \quad (a,b,c) \mapsto abc,$$

satisfying the total associativity conditions

$$(abc)de = a(bcd)e = ab(cde).$$

The homogeneous polynomial $[a,b,c]$ is a new nonassociative trilinear operation defined on the same underlying vector space. Such operations arise naturally in the study of nonassociative structures; the most important examples are the Lie and Jordan triple products:

$$[a,b,c]_{\text{Lie}} = abc - bac - cab + cba, \qquad [a,b,c]_{\text{Jordan}} = abc + cba.$$

The polynomial identities of degrees 3 and 5 satisfied by these operations in every totally associative ternary algebra define the varieties of Lie and Jordan triple systems. For the Lie triple product, the identities are:

$$\begin{aligned} [a,a,b] &= 0, \qquad [a,b,c] + [b,c,a] + [c,a,b] = 0, \\ [a,b,[c,d,e]] &= [[a,b,c],d,e] + [c,[a,b,d],e] + [c,d,[a,b,e]]. \end{aligned}$$

For the Jordan triple product, the identities are:

$$\begin{aligned} [a,b,c] &= [c,b,a], \\ [a,b,[c,d,e]] &= [[a,b,c],d,e] - [c,[b,a,d],e] + [c,d,[a,b,e]]. \end{aligned}$$

Using the representation theory of the symmetric group S_3 (permuting a, b, c) it can be shown that trilinear operations are in one-to-one correspondence with lists of three matrices with sizes 1, 2, 1 and entries from the rational numbers:

$$\left[\, x,\ \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix},\ z \,\right]$$

For example, the matrix forms of the Lie and Jordan triple products are

$$\left[0, \begin{pmatrix} 0 & 0 \\ 0 & 3 \end{pmatrix}, 0\right] \qquad \left[2, \begin{pmatrix} 0 & -1 \\ 0 & 2 \end{pmatrix}, 0\right]$$

Every trilinear operation is equivalent to an operation in which the three matrices are in row canonical form. In Bremner and Peresi [8] it is shown that there are infinitely many inequivalent trilinear operations, all of which satisfy polynomial identities of degree 3; but only eighteen of them satisfy new polynomial identities of degree 5 that are not consequences of the identities of degree 3. For nine of these operations, identities of degree 5 with a small number of terms and small integral coefficients can be obtained directly from a large matrix (the expansion matrix) by computing its row canonical form and extracting a basis for its nullspace. For the other nine operations, the same method gives identities with a large number of terms and apparently random coefficients. The operation discussed in Example 3 of the present paper is the simplest of these nine difficult cases. That operation and its matrix form are

$$2abc + 2acb - bac - bca + 2cab + 2cba \qquad \left[2, \begin{pmatrix} -1 & -1 \\ 2 & 2 \end{pmatrix}, 0\right]$$

The remaining eight difficult cases will be discussed in Bremner and Peresi [9].

References

[1] D. A. Ashlock, *Evolutionary Computation for Modeling and Optimization*, Springer, 2006, xx+572 pages, ISBN 0387221964.

[2] D. A. Ashlock and S. K. Houghten, *A novel variation operator for more rapid evolution of DNA error correcting codes*, Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, 2005.

[3] M. R. Bremner and I. R. Hentzel, *Identities for generalized Lie and Jordan products on totally associative triple systems*, Journal of Algebra 231, 1 (2000) 387–405. MR1779606

[4] M. R. Bremner and I. R. Hentzel, *Identities for algebras of matrices over the octonions*, Journal of Algebra 277, 1 (2004) 73–95. MR2059621

[5] M. R. Bremner and I. R. Hentzel, *Invariant nonassociative algebra structures on irreducible representations of simple Lie algebras*, Experimental Mathematics 13, 2 (2004) 231–256. MR2068896

[6] M. R. Bremner and I. R. Hentzel, *Identities relating the Jordan product and the associator in the free nonassociative algebra*, Journal of Algebra and its Applications 5, 1 (2006) 77–88.

[7] M. R. Bremner, Lucia I. Murakami and I. P. Shestakov, *Nonassociative Algebras*, to appear in the *CRC Handbook of Linear Algebra*, CRC Press, 2006.

[8] M. R. Bremner and L. A. Peresi, *Classification of trilinear operations and their minimal polynomial identities*, submitted; available electronically at
`http://math.usask.ca/~bremner/research/publications/BPctompi.pdf`

[9] M. R. Bremner and L. A. Peresi, *An application of evolutionary computation to polynomial identities for nonassociative systems*, in preparation.

[10] C. Darwin, *The Origin of Species by Means of Natural Selection*, Bantam Books, 1999, ix+416 pages, ISBN 0553214632. (The quotation is from page 6.)

[11] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra* (second edition), Cambridge, 2003, xiv+786 pages, ISBN 0521826462. MR2001757

[12] F. Glover and G. A. Kochenberger (editors), *Handbook of Metaheuristics*, Kluwer, 2003, xii+556 pages, ISBN 1402072635. MR1975894

[13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989, xiii+412 pages, ISBN 0201157675.

[14] I. R. Hentzel, *Solving $AX = B$ for X with the most zeros*, XV Escola de Álgebra (Canela, Brasil, 1998), Matemática Contemporânea 16 (1999) 111–116. MR1756830

[15] K. Hoffman and R. Kunze, *Linear Algebra* (second edition), Prentice-Hall, 1971, viii+407 pages, ISBN 0135367972. MR0276251

[16] E. N. Kuzmin and I. P. Shestakov, *Nonassociative Structures*, pages 197–280 of *Algebra VI: Encyclopedia of Mathematical Sciences 57*, Springer, 1995. MR1360006, MR1060322

[17] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in C: the Art of Scientific Computing* (second edition), Cambridge, 1992, xxvi+994 pages, ISBN 0521431085. MR1201159

[18] F. S. Roberts and B. Tesman, *Applied Combinatorics* (second edition), Pearson, 2005, xxiii+824 pages, ISBN 0130796034. MR0735619 (first edition)

[19] R. D. Schafer, *An Introduction to Nonassociative Algebras*, Dover, 1995, x+166 pages, ISBN 0486688135. MR1375235

[20] H. A. Simon, *Models of my Life*, Basic Books, 1991, xxix+415 pages, ISBN 0465046401. (The quotation is from page 275.)

[21] K. A. Zhevlakov, A. M. Slinko, I. P. Shestakov and A. I. Shirshov, *Rings that are Nearly Associative*, Academic Press, 1982, xi+371 pages, ISBN 0127798501. MR0668355, MR0518614