Mathematical and Computational Applications

MDPI

Article

Using the Evolution Operator to Classify Evolution Algebras †

Desamparados Fernández-Ternero, Víctor M. Gómez-Sousa * and Juan Núñez-Valdés

Department of Geometry and Topology, Faculty of Mathematics, University of Seville, Calle Tarfia, s/n, 41012 Sevilla, Spain; desamfer@us.es (D.F.-T.); jnvaldes@us.es (J.N.-V.)

* Correspondence: victor.manuel.gomez.sousa@gmail.com

† This paper is an extended version of our paper published in Proceedings of the 5th International Conference on Numerical and Symbolic Computation: Developments and Applications, Évora, Portugal, 25–26 March 2021.

Abstract: Evolution algebras are currently widely studied due to their importance not only "per se" but also for their many applications to different scientific disciplines, such as Physics or Engineering, for instance. This paper deals with these types of algebras and their applications. A criterion for classifying those satisfying certain conditions is given and an algorithm to obtain degenerate evolution algebras starting from those of smaller dimensions is also analyzed and constructed.

Keywords: evolution algebras; derivation; classification; algorithm; applications

check for updates

Citation: Fernández-Ternero, D.; Gómez-Sousa, V.M.; Núñez-Valdés, J. Using the Evolution Operator to Classify Evolution Algebras. *Math. Comput. Appl.* **2021**, *26*, 57. https://doi.org/10.3390/mca26030057

Received: 21 June 2021
Accepted: 3 August 2021
Published: 5 August 2021

1. Introduction

Evolution algebras were firstly introduced by Tian in his Ph.D. Thesis [1] in 2004, later published in a book in 2008 [2], although, in literature, researchers only usually cite [3]. These types of algebras belongs to the family of genetic algebras and have direct applications in non-Mendelian genetics [3]. In addition, its applications to other branches of mathematics and different disciplines are numerous, being connected, among others, with graph theory (see [4–6], for instance), stochastic processes and Markov chains [7,8] group theory [9], mathematical physics [7], and biology [10]. Moreover, [11–13] for analysis concerning dynamical systems modeling chemotaxis and moment problems, respectively, can be cited.

The main operator studied in evolution algebras is the so-called evolution operator, which reveals all the dynamic information about the algebra. In this paper, we will focus on the study of this operator, specifically in the case where it is a derivation. The space of derivations of an evolution algebra is a frequent subject of study in the literature [14–17], although obtaining a complete characterization of this space is still an open question. On the other hand, there are also several studies carried out with the aim of classifying certain evolution algebras [18–20]. This paper lives in the intersection of these two subjects. Our main goal is to obtain the complete classification of evolution algebras whose evolution operator is a derivation. The structure of this paper is as follows. In Section 2, we briefly introduce the main definitions and properties of evolution algebras. In Section 3, we review the main results obtained in [21], and we develop the methods outlined in this paper in order to extend the classification of these algebras to any dimension through a characterization theorem. Finally, we present some applications of this type of evolution algebras in Section 4.

2. Preliminaries on Evolution Algebras

Let $E \equiv (E, +, \cdot)$ be an algebra over a field $\mathbb{K}$. It is said that E is an *evolution algebra* if there exists a basis $\mathcal{B} = \{e_i : i \in \Lambda\}$ of E, where Λ is an index set, such that $e_i \cdot e_j = 0$, if $i \neq j$. The basis $\mathcal{B}$ is called a natural basis.

Since $\mathcal{B}$ is a basis, the product $e_j \cdot e_j = e_j^2$ can be written as $\sum_{i \in \Lambda} a_{ij} e_i$, with $a_{ij} \in \mathbb{K}$, where only a finite quantity of a_{ij}, called *structure constants*, are non-zero for each $j \in \Lambda$ fixed. Thus, the product on E is determined by the *structure matrix* $A = (a_{ij})$.

In general, evolution algebras are neither associative nor power-associative, but they are flexible.

An evolution algebra is said to be *non-degenerate* if $e_j^2 \neq 0$, for all $j \in \Lambda$. On the contrary, the algebra is called *degenerate*.

Tian defined in [2] the *evolution operator* associated with $\mathcal{B}$ as the endomorphism $L : E \to E$, which maps each generator onto its square, that is, $L(e_j) = e_j^2 = \sum_{i \in \Lambda} a_{ij} e_i$, for all $j \in \Lambda$. The matrix representation of the evolution operator with respect to the basis $\mathcal{B}$ is the structure matrix A.

3. Evolution Algebras Whose Evolution Operator Is a Derivation

An endomorphism $d : E \to E$ is called a derivation if $d(x \cdot y) = d(x) \cdot y + x \cdot d(y)$, for all $x, y \in E$. In this section, we study those evolution algebras over $\mathbb{C}$ such that $L \in Der(E)$, i.e., the evolution operator is a derivation. This condition, as shown in [21], is equivalent to

$$A\bar{A} = 0, \tag{1}$$

$$a_{kj}a_{ji} + a_{ki}a_{ij} = 0, \ \forall k, \ \forall i \neq j, \tag{2}$$

where $\bar{A}$ is the matrix obtained from A by changing the sign of its diagonal elements.

Remark 1. *Suppose there exists $i, j \in \Lambda$ such that $a_{ij} \neq 0$ and $a_{ji} = 0$. Then, from condition (2), column i is null.*

In [21], a study is carried out to characterize these types of algebras, up to ordering of the basis elements, reaching the following results.

- If E is non-degenerate, then its dimension is even, and the structure matrix is a block diagonal matrix $Diag(M_{a_1}, M_{a_2}, \ldots)$, where

$$M_{a_i} = \begin{pmatrix} a_i & -a_i \\ -a_i & a_i \end{pmatrix},$$

  for some $a_i \neq 0$.
- If E is degenerate, the following algorithm to obtain all the structure matrices A of these algebras in dimension n is given.

  Step 1: Obtain all matrices A' of dimension $n - 1$, both degenerate and non-degenerate.

  Step 2: For each A', impose conditions

$$v^t \bar{A}' = 0 \tag{3}$$

  and (2) on matrix $A = \begin{pmatrix} A' & 0 \\ v^t & 0 \end{pmatrix}$.

Additionally, the complete classification for dimensions less than five is given in [21]. We will focus on the classification for larger dimensions, and we will give a result that allows us to characterize the non-degenerate case without using the previous algorithm.

3.1. The Algorithm

In this subsection, we analyze and modify the previous algorithm for the degenerate case in order to eliminate unnecessary conditions and improve its efficiency.

In Step 2 of the algorithm, we impose conditions (2) on matrix A. Since A is a $n \times n$ matrix, there are $\frac{(n-1)n^2}{2}$ conditions to impose. However, some of these conditions are trivially satisfied

- If $i, j, k \in \{1, \ldots, n-1\}$, then these conditions are fulfilled, since all of these indices are from the submatrix A'.

- If $i = n$ or $j = n$, then these conditions are $0 = 0$.

Therefore, there remain the conditions with $k = n$ and $i, j \in \{1, \dots, n-1\}$. If we write $v^t = (v_1, \dots, v_{n-1})$, these conditions are

$$v_i a_{ij} + v_j a_{ji} = 0, \;\; \forall 1 \leq i < j \leq n-1. \tag{4}$$

That is, all the conditions we impose on matrix A are linear equations on the vector v.

Example 1. *If $A' = (a_{ij})_{i,j \in \{1,\dots,3\}}$, then the linear equations of the conditions (3) and (4) are written in matrix form as*

$$\begin{pmatrix} -a_{11} & a_{21} & a_{31} \\ a_{12} & -a_{22} & a_{32} \\ a_{13} & a_{23} & -a_{33} \\ a_{12} & a_{21} & 0 \\ a_{13} & 0 & a_{31} \\ 0 & a_{23} & a_{32} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Taking into account these changes, we implement the second step of the algorithm in Python below

```
from sympy import Matrix

def aux(a,b,A):
L = []
n = A.shape[0]
for i in range(n):
if i==a:
L = L+[A[a,b]]
elif i==b:
L = L+[A[b,a]]
else:
L = L+[0]
return L

def algorithm(A):
n = A.shape[0]
A1 = Matrix(n,n, lambda i,j: -A[i,i] if i==j else A[j,i])
A2 = Matrix([aux(i,j,A) for i in range(n) for j in range(i+1,n)])
AA = A1.col_join(A2)
return AA.nullspace()
```

3.2. Computational and Complexity Study

In this subsection, we show a computational study of the previous algorithm, which has been implemented with an Intel Core i5-1135G7 processor (frequency of 2.4 GHz) and 16 GB of RAM. To do this study, we have considered the family of evolution algebras with a structure matrix of the form $Diag(M_{a_1}, M_{a_2}, \dots)$. Table 1 shows the computing time used to return the output of the whole procedure according to the dimension n of the algebra.

Table 1. Computing time.

Input	Computing Time (s)	Input	Computing Time (s)
$n = 2$	0.0	$n = 80$	4.047
$n = 6$	0.031	$n = 100$	7.516
$n = 10$	0.094	$n = 120$	12.641
$n = 20$	0.203	$n = 140$	19.625
$n = 30$	0.406	$n = 160$	29.313
$n = 40$	0.688	$n = 180$	41.016
$n = 60$	1.875	$n = 200$	55.797

Let us compute the complexity of the algorithm taking into account the number of operations carried out in the worst case. For this purpose, we use the big O notation: Given two functions $f, g : \mathbb{R} \to \mathbb{R}$, we say that $f(x) = O(g(x))$ if there exist $M \in \mathbb{R}^+$ and $x_0 \in \mathbb{R}$ such that $|f(x)| \leq Mg(x)$, for all $x \geq x_0$. For our algorithm, we have

- The auxiliary function has complexity $O(n)$, since the loop is executed n times.
- The definition of $A1$ has complexity $O(n^2)$.
- The definition of $A2$ has complexity $O(n^3)$, since the loop is executed $\frac{n(n-1)}{2} = O(n^2)$ times and, in each iteration, the auxiliary function is called.

That is, if functions *col_join* and *nullspace* have complexity $O(n^3)$, then the running time of the algorithm is $O(n^3)$. To check if these assumptions are correct, we calculate the least square cubic polynomial fit to our experimental data in Table 1, and we make predictions for larger dimensions ($n = 220, 240, 260, 280$). As can be seen in Table 2 and Figure 1, our data fit this model, so our algorithm has complexity $O(n^3)$.

Table 2. Real computing time and expected time by the cubic fit in higher dimensions.

Input	Computing Time (s)	Expected Time (s)
$n = 220$	74.297	73.882
$n = 240$	95.797	95.450
$n = 260$	121.969	120.867
$n = 280$	152.188	150.449

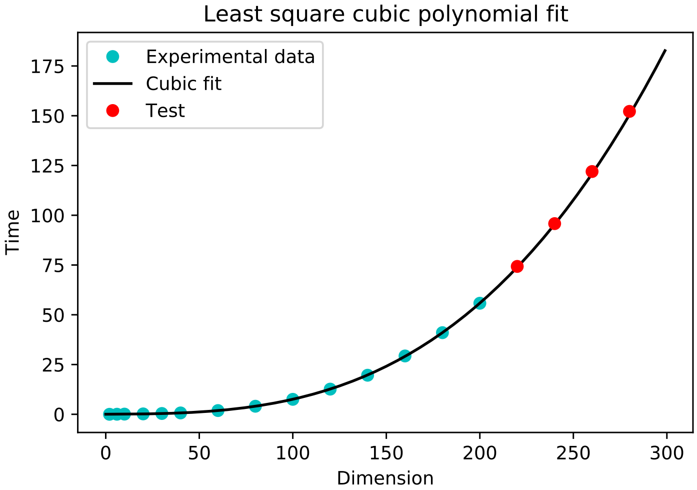

Figure 1. Least square cubic polynomial fit to the experimental data and test data for predictions.

3.3. *Five-Dimensional Evolution Algebras*

Since this dimension is odd, there do not exist non-degenerate evolution algebras. For the degenerate cases, we use the previous algorithm. The complete classification in dimension 4 is the following (see [21]):

$$A_1=\begin{pmatrix} a & -a & 0 & 0\\ -a & a & 0 & 0\\ 0 & 0 & b & -b\\ 0 & 0 & -b & b\end{pmatrix},\ A_2=\begin{pmatrix} a & -a & 0 & 0\\ -a & a & 0 & 0\\ b & -b & 0 & 0\\ c & -c & 0 & 0\end{pmatrix},\ A_3=\begin{pmatrix} a & -a & 0 & 0\\ -a & a & 0 & 0\\ 0 & 0 & 0 & 0\\ b & -b & c & 0\end{pmatrix},$$

$$A_4=\begin{pmatrix} 0 & 0 & 0 & 0\\ a & 0 & 0 & 0\\ b & 0 & 0 & 0\\ c & 0 & 0 & 0\end{pmatrix},\ A_5=\begin{pmatrix} 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ a & b & 0 & 0\\ c & d & 0 & 0\end{pmatrix},\ A_6=\begin{pmatrix} 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ a & b & c & 0\end{pmatrix},$$

with $a,b,c,d\in\mathbb{C}$. Let us make use of the algorithm for these matrices

- For A_1: We can assume that $a,b\neq 0$, since, if any of these constants is null, it would be treated as the following cases (if necessary, by a permutation of the basis). If we run the following code,

  ```
  from sympy import var
  var('a b')
  A = Matrix([[a,-a,0,0],[-a,a,0,0],[0,0,b,-b],[0,0,-b,b]])
  algorithm(A)
  ```

  then we obtain $v^t\in\langle(-1,1,0,0)^t,(0,0,-1,1)^t\rangle$, so we get the matrices of the form

  $$\begin{pmatrix} a & -a & 0 & 0 & 0\\ -a & a & 0 & 0 & 0\\ 0 & 0 & b & -b & 0\\ 0 & 0 & -b & b & 0\\ c & -c & d & -d & 0\end{pmatrix}.$$

- For A_2: We can assume that $a,b,c\neq 0$. Running the algorithm, we obtain $v^t\in\langle(-1,1,0,0)^t\rangle$. That is, matrices of the form

  $$\begin{pmatrix} a & -a & 0 & 0 & 0\\ -a & a & 0 & 0 & 0\\ b & -b & 0 & 0 & 0\\ c & -c & 0 & 0 & 0\\ d & -d & 0 & 0 & 0\end{pmatrix}.$$

- For A_3: We can assume that $a\neq 0$. Running the algorithm, we get the matrices

  $$\begin{pmatrix} a & -a & 0 & 0 & 0\\ -a & a & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ b & -b & c & 0 & 0\\ d & -d & e & 0 & 0\end{pmatrix} \text{ and } \begin{pmatrix} a & -a & 0 & 0 & 0\\ -a & a & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ d & -d & e & f & 0\end{pmatrix},$$

  depending on whether the constants b and c are null.
- For A_4: We can assume that $a,b,c\neq 0$. Running the algorithm, we get $v^t\in\langle(1,0,0,0)^t\rangle$, which corresponds to the matrices

  $$\begin{pmatrix} 0 & 0 & 0 & 0 & 0\\ a & 0 & 0 & 0 & 0\\ b & 0 & 0 & 0 & 0\\ c & 0 & 0 & 0 & 0\\ d & 0 & 0 & 0 & 0\end{pmatrix}.$$

- For A_5: We can assume that a and b are not both null, the same as c and d. Running the algorithm, we obtain $v^t \in \langle (1,0,0,0)^t, (0,1,0,0)^t \rangle$, so we get the matrices

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ a & b & 0 & 0 & 0 \\ c & d & 0 & 0 & 0 \\ e & f & 0 & 0 & 0 \end{pmatrix}.$$

- For A_6: Running the algorithm, we obtain the matrices

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ a & b & c & 0 & 0 \\ d & e & f & 0 & 0 \end{pmatrix} \text{ and } \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ d & e & f & g & 0 \end{pmatrix},$$

depending on whether the constants a, b, c are null.

3.4. Characterization in the Degenerate Case

In this subsection, we deal with the degenerate case. To do it, we firstly show the following general result.

Theorem 1. *Let E be a degenerate evolution algebra with natural basis $\mathcal{B} = \{e_i : i = 1, \ldots, n\}$ and structure matrix $A = (a_{ij})$. Then, $L \in Der(E)$ if and only if there exists a rearrangement of the basis $\mathcal{B}$ such that the structure matrix has any of the following forms:*

A)

$$\begin{pmatrix} D & O \\ & A' \end{pmatrix}, \tag{5}$$

where

- *D is a $2r \times 2r$ matrix of the form $Diag(M_{a_1}, \ldots, M_{a_r})$, for some r with $1 \leq r < \frac{n}{2}$ and $a_i \neq 0$.*
- *O is the $2r \times (n - 2r)$ zero matrix.*
- *A' is a $(n - 2r) \times n$ matrix whose only non-null elements are in the lower left $(n - k) \times k$ submatrix, for some k with $2r \leq k \leq n - 1$, and such that its columns fulfill that $c_2 = -c_1, c_4 = -c_3, \ldots, c_{2r} = -c_{2r-1}$.*

B)

$$\begin{pmatrix} O_0 & O_1 \\ A' & O_2 \end{pmatrix}, \tag{6}$$

where

- *O_0 is the $k \times k$ zero matrix, for some $k \geq 1$.*
- *O_1 is the $k \times (n - k)$ zero matrix.*
- *O_2 is the $(n - k) \times (n - k)$ zero matrix.*
- *A' is a $(n - k) \times k$ matrix.*

Proof. The right-to-left implication is carried out by checking that these matrices meet the conditions (1) and (2), which is easy to check.

For the left-to-right implication, we proceed by induction in n. We know that the result is true for $n \leq 5$. Suppose that it is true for dimension n and let us see that it is true for dimension $n + 1$.

Let A be the structure matrix of a degenerate evolution algebra E of dimension $n+1$, whose evolution operator satisfies $L \in Der(E)$. Then, there exists A' such that

$$A = \begin{pmatrix} A' & 0 \\ v^t & 0 \end{pmatrix},$$

where A' is the structure matrix of an evolution algebra E' of dimension n whose evolution operator L' is a derivation. Then, by induction hypothesis, we are in one of the following cases:

Case 1: E' is non-degenerate. Then, $A' = Diag(M_{a_1}, M_{a_2}, \dots)$, for some $a_i \neq 0$. From condition (3), we obtain that $v_2 = -v_1$, $v_4 = -v_3, \dots, v_n = -v_{n-1}$. Thus, matrix A is like in A).

Case 2: A' is like (5). Let A'' be the lower $(n-2r) \times n$ submatrix in A'.

If A'' is null, from condition (3), we obtain that $v_2 = -v_1$, $v_4 = -v_3, \dots, v_{2r} = -v_{2r-1}$. Thus, A is like in A), being $k = n$.

In another case, we can assume that the $(n-k) \times k$ submatrix of A'' does not have any null rows, since, in this case, we could reorder the basis so that this non-null matrix would be of size $(n-k-1) \times k$ and could be treated as one of size $(n-k-1) \times (k+1)$, in which that null row would no longer appear.

Since no row is null, from Remark 1, we obtain that $v_{k+1} = v_{k+2} = \cdots = v_n = 0$. Again, by condition (3), $v_2 = -v_1$, $v_4 = -v_3, \dots, v_{2r} = -v_{2r-1}$. Thus, A is like in A).

Case 3: A' is like (6). Let A'' be the lower $(n-k) \times k$ submatrix in A'. If A'' is null, then A is trivially like in B).

In another case, we can assume, as before, that, in the $(n-k) \times k$ submatrix of A'', there is no null row. Then, by Remark 1, we obtain $v_{k+1} = v_{k+2} = \cdots = v_n = 0$. Thus, A is like in B).

□

3.5. *Six-Dimensional Evolution Algebras*

The non-degenerate cases are those of the form $Diag(M_a, M_b, M_c)$, with $a, b, c \neq 0$. By the previous theorem, the degenerate cases are

$$\begin{pmatrix} a & -a & 0 & 0 & 0 & 0 \\ -a & a & 0 & 0 & 0 & 0 \\ 0 & 0 & b & -b & 0 & 0 \\ 0 & 0 & -b & b & 0 & 0 \\ c & -c & d & -d & 0 & 0 \\ e & -e & f & -f & 0 & 0 \end{pmatrix}, \begin{pmatrix} a & -a & 0 & 0 & 0 & 0 \\ -a & a & 0 & 0 & 0 & 0 \\ 0 & 0 & b & -b & 0 & 0 \\ 0 & 0 & -b & b & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ c & -c & d & -d & e & 0 \end{pmatrix}, \begin{pmatrix} a & -a & 0 & 0 & 0 & 0 \\ -a & a & 0 & 0 & 0 & 0 \\ b & -b & 0 & 0 & 0 & 0 \\ c & -c & 0 & 0 & 0 & 0 \\ d & -d & 0 & 0 & 0 & 0 \\ e & -e & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\begin{pmatrix} a & -a & 0 & 0 & 0 & 0 \\ -a & a & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ b & -b & c & 0 & 0 & 0 \\ d & -d & e & 0 & 0 & 0 \\ f & -f & g & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} a & -a & 0 & 0 & 0 & 0 \\ -a & a & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ b & -b & c & d & 0 & 0 \\ e & -e & f & g & 0 & 0 \end{pmatrix}, \begin{pmatrix} a & -a & 0 & 0 & 0 & 0 \\ -a & a & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ b & -b & c & d & e & 0 \end{pmatrix},$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ a & 0 & 0 & 0 & 0 & 0 \\ b & 0 & 0 & 0 & 0 & 0 \\ c & 0 & 0 & 0 & 0 & 0 \\ d & 0 & 0 & 0 & 0 & 0 \\ e & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ a & b & 0 & 0 & 0 & 0 \\ c & d & 0 & 0 & 0 & 0 \\ e & f & 0 & 0 & 0 & 0 \\ g & h & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ a & b & c & 0 & 0 & 0 \\ d & e & f & 0 & 0 & 0 \\ g & h & i & 0 & 0 & 0 \end{pmatrix},$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ a & b & c & d & 0 & 0 \\ e & f & g & h & 0 & 0 \end{pmatrix} \text{ and } \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ a & b & c & d & e & 0 \end{pmatrix}.$$

4. Applications

Among the most recent references referring to the applications of the evolution operator (it cannot be said that there are many in the literature), the two following ones can be cited: that by Absalamov and Rozikov [22], who investigate in the field of biology the dynamical system generated by a gonosomal evolution operator of sex linked inheritance depending on parameters and the one by Padmanabhan, related to physics, who studies the Planck scale by the modification of the time evolution operator due to the quantum structure of spacetime [23]. We have tried to find some applications of this operator related to both scientific and non-scientific fields, as they might be in the latter case economics and commerce, for example, although we have not quite achieved what we wanted. The one that we present below is one of them, quite modest, in our opinion and that could even seem artificial or forced, but the authors believe that it can serve as a starting point and model for obtaining other more useful and consistent ones, similar or in the way of [24–26], for instance.

Suppose we have two different environments, m and m', and certain types of objects $\{o_1, o_2, \dots\}$ in these environments. Let us see how to associate an evolution algebra to this assumption in such a way that the evolution operator models the movement of objects between the two environments.

Let us consider the vector space with basis $\{e_1, e'_1, e_2, e'_2, \dots\}$ and let us identify e_i with an object of type o_i in the environment m and e'_i with an object of the same type in m'. Let T be the total amount of objects in these environments. Now, consider the evolution algebra whose structure matrix is $Diag(M_{-1}, M_{-1}, \dots)$. Then, we have that $L(e_i) = -e_i + e'_i$ and $L(e'_i) = e_i - e'_i$, so the evolution operator models the transfer of objects between environments. It can be illustrated with the following:

Example 2. *Let us consider the basis $\{e_1, e'_1, e_2, e'_2\}$, the structure matrix $Diag(M_{-1}, M_{-1})$, and let $T = 2e_1 + e'_1 + e'_2$. This means that there are two units of o_1 in m, while, in m', there is one unit of o_1 and another of o_2. If we want to model the transfer of a unit of o_1 from m to m', we act as follows:*

$$T \mapsto T + L(e_1) = (2e_1 + e'_1 + e'_2) + (-e_1 + e'_1) = e_1 + 2e'_1 + e'_2.$$

Let us now analyze some cases in which the evolution algebra is degenerate. For example, the evolution algebra with structure matrix

$$\begin{pmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ -n & n & 0 & 0 \\ n & -n & 0 & 0 \end{pmatrix} \text{ or } \begin{pmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ n & -n & 0 & 0 \\ -n & n & 0 & 0 \end{pmatrix}$$

can be used when the movement of two objects are carried out at the same time and in the same or opposite directions, respectively. In this case, n is the number of units of o_2 transferred for each unit of o_1.

The above cases may have direct applications in biology, for instance in cellular exchanges. Carrying out some modifications, we can also model, for example, the sales of a store.

Let $\{o_1, o_2\}$ be two items with prices $\{p_1, p_2\}$, respectively. Let us consider the basis $\{e_1, e'_1, e_2, e'_2, \eta\}$ and the structure matrix

$$\begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ p_1 & -p_1 & p_2 & -p_2 & 0 \end{pmatrix}.$$

If we identify the basis vectors as before and η with a monetary unit, then the evolution operator models a transaction (sale or return) in the store.

5. Conclusions

Classifications in evolution algebras by means of the behavior of the evolution operator are scant, and even more for high dimensions. The characterization theorem given in this paper allows for obtaining the complete classification in the case of derivations, in any dimension. However, some related problems remain open. We show some of them here.

- When the evolution operator is a derivation, it is easy to check that the equality $L^n(x \cdot y) = \sum_{k=0}^{n} \binom{n}{k} L^{n-k}(x) \cdot L^k(y)$ holds for $n > 1$. An operator is said to be a *derivation of order n* when it satisfies the above equality for n. It could be studied when the evolution operator is a derivation of orden n, for $n > 1$.
- When the evolution operator is a derivation, it is also easy to check that the equality $L(x_1 \cdots x_n) = \sum_{i=1}^{n} x_1 \cdots x_{i-1} \cdot L(x_i) \cdot x_{i+1} \cdots x_n$ holds for $n > 2$. An operator is said to be a *n-derivation* when it satisfies the above equality for n. It could be studied when the evolution operator is an n-derivation, for $n > 2$.
- Classify evolution algebras whose evolution operator satisfies other equalities. For example, the case in which the evolution operator is an endomorphism of algebras is analyzed in [21], that is, when $L(x \cdot y) = L(x) \cdot L(y)$. This case could be studied in depth.
- Transfer the results obtained to other branches of Mathematics that have direct connections with evolution algebras.

Author Contributions: The authors claim to have contributed equally and significantly in this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Consejería de Economía, Innovación, Ciencia y Empleo, Junta de Andalucía Research Group FQM-326.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tian, J.P. Evolution Algebra Theory. Ph.D. Thesis, University of California, Riverside, CA, USA, 2004.
2. Tian, J.P. *Evolution Algebras and Their Applications*; Springer: Berlin, Germany, 2008.
3. Tian, J.P.; Vojtechovsky, P. Mathematical concepts of evolution algebras in non-Mendelian genetics. *Quasigr. Relat. Syst.* **2006**, *14*, 111–122.
4. Elduque, A.; Labra, A. Evolution algebras and graphs. *J. Algebra Appl.* **2015**, *14*, 1550103. [CrossRef]
5. Núñez, J.; Rodríguez, M.L.; Villar, M.T. Certain particular families of graphicable algebras. *Appl. Math. Comput.* **2014**, *246*, 416–425. [CrossRef]
6. Núñez, J.; Silvero, M.; Villar, M.T. Mathematical tools for the future: Graph Theory and graphicable algebras. *Appl. Math. Comput.* **2013**, *219*, 6113–6125. [CrossRef]
7. Cadavid, P.; Rodiño, M.L.; Rodriguez, P.M. The connection between evolution algebras, random walks and graphs. *J. Algebra Appl.* **2018**, *19*, 2050023. [CrossRef]
8. López, F.; Núñez, J.; Recacha, S.; Villar, M.T. Connecting Statistics, Probability, Algebra and Discrete Mathematics. *arXiv* **2021**, arXiv:2104.02470.
9. Wears, T.H. On algebraic solitons for geometric evolution equations on three-dimensional Lie groups. *Tbilisi Math. J.* **2016**, *9*, 2. [CrossRef]
10. Falcón, O.J.; Falcón, R.M.; Núñez, J. Classification of asexual diploid organisms by means of strongly isotopic evolution algebras defined over any field. *J. Algebra.* **2017**, *472*, 573–593. [CrossRef]
11. Li, T.; Viglialoro, G. Boundedness for a nonlocal reaction chemotaxis model even in the attraction-dominated regime. *Differ. Integral Equ.* **2021**, *34*, 315–336.

12. Viglialoro, G.; Woolley, T.E. Solvability of a Keller-Segel system with signal-dependent sensitivity and essentially sublinear production. *Appl. Anal.* **2020**, *99*, 2507–2525. [CrossRef]
13. Ghasemi, M.; Infusino, M.; Kuhlmann, S.; Marshall, M. Moment problem for symmetric algebras of locally convex spaces. *Integral Equ. Oper. Theory* **2018**, *90*, 1–19. [CrossRef]
14. Alsarayreh, A.; Qaralleh, I.; Ahmad, M.Z. Derivation of three-dimensional evolution algebra. *JP J. Algebra Number Theory Appl.* **2017**, *39*, 425–444. [CrossRef]
15. Cadavid, P.; Rodiño, M.L.; Rodriguez, P.M. Characterization theorems for the spaces of derivations of evolution algebras associated with graphs. *Linear Multilinear Algebra* **2020**, *68*, 1340–1354. [CrossRef]
16. Camacho, L.M.; Gómez, J.R.; Omirov, B.A.; Turdibaev, R.M. The derivations of some evolution algebras. *Linear Multilinear Algebra* **2013**, *61*, 309–322. [CrossRef]
17. Mukhamedov, F.; Khakimov, O.; Omirov, B.; Qaralleh, I. Derivations and automorphisms of nilpotent evolution algebras with maximal nilindex. *J. Algebra Appl.* **2019**, *18*, 1950233. [CrossRef]
18. Cabrera, Y.; Siles, M.; Velasco, M.V. Classification of three-dimensional evolution algebras. *Linear Algebra Appl.* **2017**, *524*, 68–108. [CrossRef]
19. Houida, A.; Ural, B.; Isamiddin, R. On classification of two-dimensional evolution algebras and its applications. *J. Phys. Conf. Ser.* **2020**, *1489*. [CrossRef]
20. Murodov, S.N. Classification of two-dimensional real evolution algebras and dynamics of some two-dimensional chains of evolution algebras. *Uzbek. Mat. Zh.* **2014**, *2*, 102–111.
21. Fernández-Ternero, D.; Gómez-Sousa, V.M.; Núñez-Valdés, J. The evolution operator of evolution algebras. *Linear Multilinear Algebra* **2021**. [CrossRef]
22. Absalamov, A.T.; Rozikov, U.A. The dynamics of gonosomal evolution operators. *J. Appl. Nonlinear Dyn.* **2020**, *9*, 247–257. [CrossRef]
23. Padmanabhan, T. Probing the Planck scale: The modification of the time evolution operator due to the quantum structure of spacetime. *J. High Energy Phys.* **2020**, *11*, 13. [CrossRef]
24. Akararungruangkul, R.; Kaewman, S. Modified differential evolution algorithm solving the special case of location routing problem. *Math. Comput. Appl.* **2018**, *23*, 34. [CrossRef]
25. Kaewman, S.; Srivarapongse, T.; Theeraviriya, C.; Jirasirilerd, G. Differential evolution algorithm for multilevel assignment problem: A case study in chicken transportation. *Math. Comput. Appl.* **2018**, *23*, 55. [CrossRef]
26. Sriboonchandr, P.; Kriengkorakot, N.; Kriengkorakot, P. Improved differential evolution algorithm for flexible job shop scheduling problems. *Math. Comput. Appl.* **2019**, *24*, 80. [CrossRef]