Two new operations over extended index matrices and their applications in Big Data

Krassimir Atanassov*,†, Veselina Bureva†

*Dept. of Bioinformatics and Mathematical Modelling
Institute of Biophysics and Biomedical Engineering,
Bulgarian Academy of Sciences
105 Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
†Intelligent Systems Laboratory
“Prof. Asen Zlatarov” University
Burgas–8010, Bulgaria
E-mail: krat@bas.bg, vbureva@btu.bg

Abstract—The Index Matrices (IMs) are extensions of the matrices of algebra. Over the IMs different operations, relations and operators are defined. In the present paper, two new operations over IMs are defined and some of their properties are studied. An application of these operations for describing of Big Data procedure is discussed. Hortonworks Data Platform (HDP) is used to provide capabilities for data warehouse processing in the Big Data environment. Apache Hive is selected for data warehouse construction and querying. Firstly, data warehouse for product sells is implemented. The tables for employees, customers, products and product sells are created. Thereafter for product sells is implemented. The tables for employees, warehouse construction and querying. Firstly, data warehouse

Index Terms—Big Data, Extended index matrix, Operation.

I. INTRODUCTION

The CONCEPT of Index Matrix (IM) was introduced in 1984 and in more details - in 1987 [2], but the full description of the research over them was published in [3] exactly 30 years later.

Different extensions and modifications of the concept of an IM are described in [3]. One of them is an Extended IM (EIM), introduced firstly in [4]. They include as partial cases standard IM with elements of real or complex numbers, the (0, 1)-IM with elements from set {0, 1}, the logical IM with elements variables, propositions or predicates, the intuitionistic fuzzy IMs. The elements of the EIM can be each objects, in this number - whole IMs.

Different relations, operations and operators are defined over IMs and more general - over EIM. Only part of them have analogues in the theory of the standard matrices (see, e.g., [6], [7]).

Here, two new operations over EIMs are defined and some of their properties are studied. It will be obvious that these new operations can be transfer over each one of the partial cases of the EIM.

Firstly, we give the definition of an EIM.

Let I be a fixed set of indices,

\[ I^n = \{ (i_1, i_2, ..., i_n) | (\forall j : 1 \leq j \leq n)(i_j \in I) \} \]

and

\[ I^* = \bigcup_{1 \leq n \leq \infty} I^n. \]

Let \( \chi \) be a fixed set of some objects. In the particular cases, they can be either real numbers, or only the numbers 0 or 1, or logical variables, propositions or predicates, etc.

Let operations \( \circ, \cdot, \ast : \chi \times \chi \rightarrow \chi \) be fixed and let \( (\chi, \circ, e_\circ) \) and \( (\chi, \cdot, e_\cdot) \) be groups with unit elements \( e_\circ \) and \( e_\cdot \), respectively. For example, when operation \( \"\circ\" \) is \( \"\ast\" \) or \( \"\cdot\" \), \( e_\circ \) will be 0, while when it is \( \"\times\" \) or \( \"\:=\" \) it will be 1.

In some cases, it is suitable to define the unit element by \( \bot \) and it to be an empty object.

An EIM with index sets \( K \) and \( L \) \((K, L \subset I^*)\) and elements from set \( \chi \) is called the object (see, [4], [3]):

\[ [K, L, \{a_{k_i, l_j}\}] \equiv \begin{bmatrix} l_1 & \ldots & l_n \end{bmatrix}_{k_i \ldots k_m} \begin{bmatrix} a_{k_1, l_1} & \ldots & a_{k_1, l_n} \\ \vdots & \ddots & \vdots \\ a_{k_m, l_1} & \ldots & a_{k_m, l_n} \end{bmatrix} \]

where \( K = \{k_1, k_2, ..., k_m\}, L = \{l_1, l_2, ..., l_n\}, 1 \leq i \leq m, 1 \leq j \leq n : a_{k_i, l_j} \in \chi. \)

In [3], for the IMs \( A = [K, L, \{a_{k_i, l_j}\}], B = [P, Q, \{b_{p_i, q_j}\}] \), operations that are analogous to the usual matrix operations of addition and multiplication are defined, as well as other, specific ones. Here, we give only three of these operations, that will be used below.

Addition

\[ A \oplus_\circ B = [K \cup P, L \cup Q, \{c_{e_i, v_j}\}] \]
where

\[ c_{u,w} = \begin{cases} a_{k,i}, & \text{if } t_u = k_i \in K \text{ and } v_w = l_j \in L - Q, \\
 0, & \text{otherwise} \end{cases} \]

\[ b_{v,w} = \begin{cases} a_{k,i} \circ b_{p,v}, & \text{if } t_u = p_r \in P \text{ and } v_w = q_s \in Q - L, \\
 0, & \text{otherwise} \end{cases} \]

**Structural subtraction**

\[ A \ominus B = [K - P, L - Q, \{c_{u,v}\}], \]

where here and below “-” is the set-theoretic difference operation and

\[ c_{u,v} = a_{k,i}, \text{ for } t_u = k_i \in K - P \text{ and } v_w = l_j \in L - Q. \]

**Projection**

\[ pr_{M,N}A = [M, N, \{b_{k,i}\}], \]

where \( M \subseteq K, N \subseteq L \), and for each \( k_i \in M \) and each \( l_j \in N, b_{k,i} = a_{k,i}. \)

II. DEFINITIONS OF THE TWO NEW OPERATIONS

Let the two EIMs \( A = [K, L, \{a_{k,i}\}] \) and \( B = [P, Q, \{b_{p,v}\}] \) are given.

The first, simpler, operation is defined by

\[ A \circ B = [K - P, L - Q, \{c_{u,v}\}], \]

where and below for every two arbitrary sets \( X, Y \):

\[ X \div Y = (X - Y) \cup (Y - X); \]

\[ c_{u,v} = \begin{cases} a_{k,i}, & \text{if } u = k_i \in K - P \text{ and } v = l_j \in L - Q, \\
 b_{p,v}, & \text{if } u = p_r \in P - K \text{ and } v = q_s \in Q - L, \\
 0, & \text{otherwise} \end{cases} \]

The second operation is defined by

\[ A \odot B = [K \cup P, L \cup Q, \{c_{u,v}\}], \]

where

\[ c_{u,v} = \begin{cases} a_{k,i}, & \text{if } u = k_i \in K - P \text{ and } v = l_j \in L, \\
 b_{p,v}, & \text{if } u = p_r \in P - K \text{ and } v = q_s \in Q, \\
 0, & \text{otherwise} \end{cases} \]

The geometrical interpretations of both operations are shown on Figures 1 and 2.

From the definitions and geometrical interpretations of both operations we see immediately that the following assertions are valid.

**Theorem 1.** For every two EIMs \( A \) and \( B \), for each operation \( o \) and for operation \( * \) defined for every two \( x, y \in X \) by \( x * y = e \), there are follows:

\[ A \odot B = (A \circ B) \oplus (B \odot A), \]

\[ A \odot B = A \oplus (B \odot A). \]

**Proof.** For the definitions of the operations \( \oplus \) and \( \odot \) we obtain

\[ (A \oplus B) \oplus (B \odot A) = [K - P, L - Q, \{c_{u,v}\}] \oplus [P - K, Q - L, \{d_{r,s}\}] \]

where \( c_{u,v,w} = a_{k,i} \), \( d_{r,s,w} = b_{p,v} \), for each operation \( o \) and \( * \) cannot be applied over the elements of both EIMs because there are not at least two elements from the both EIMs that have equal indices.

The second equality is proved by a similar way.

**Theorem 2.** For every two EIMs \( A \) and \( B \), for each operation \( \circ \):

\[ A \odot B = pr_{K-P,L-Q}A \oplus (pr_{P-K,Q-L}B), \]

\[ A \odot B = pr_{K-P,L}A \oplus (pr_{P-K,Q}B) \oplus pr_{P-Q}A \]

The proof is similar to the above one.

Let \( M \) be the set of all EIMs with elements from \( X \).

Let \( I_\emptyset = \{[0, 0, \bot]\} \).

**Theorem 3.** \( M, \circ, I_\emptyset \) is a commutative group.

**Proof.** From the definition of operation \( \circ \) it follows directly that for each two EIMs \( A, B \in M \), \( A \odot B \in M \).

Using the well-known equality for every three sets \( X, Y \) and \( Z \):

\[ (X \div Y) \div Z = X \div (Y \div Z), \]

for every three EIMs \( A, B, C \in M \), where

\[ C = [D, E, \{c_{d,v}\}] \]

we obtain

\[ (A \odot B) \odot C = ([K, L, \{a_{k,i}\}] \odot [P, Q, \{b_{p,v}\}]) \odot [D, E, \{c_{d,v}\}] \]

\[ = [K \cup P, L \cup Q, \{x_{u,v}\}] \odot [D, E, \{c_{d,v}\}] \]

\[ \text{when } X \text{ is a set (class) in the sense of NBG-set theory of all predicates, then } M \text{ will be a set (class).} \]
Now, for EIM \( I \) we obtain
\[
A \bullet I_{\emptyset} = [K \div \emptyset, L \div \emptyset, \{c_{u_{0}, w_{1}}\}] \quad (\text{where each element } c_{u_{0}, w_{1}} \text{ coincides with the element } a_{k_{i}, l_{j}} \text{ from } A \text{ for } u_{0} = k_{i}, w_{1} = l_{j})
\]

Analogously is checked that
\[
I_{\emptyset} \bullet A = A.
\]

From the well-known equality \( X \div Y = Y \div X \) it follows that
\[
A \bullet B = [K \div P, L \div Q, \{c_{u_{0}, w_{1}}\}] \\
m = [P \div K, Q \div L, \{c_{u_{0}, w_{1}}\}] = B \bullet A,
\]
i.e., the operation \( \bullet \) is commutative.

Finally,
\[
A \bullet A = [K \div K, L \div L, \{x_{u_{0}, w_{1}}\}] = [\emptyset, \emptyset, \{x_{u_{0}, w_{1}}\}] = I_{\emptyset},
\]
because of lack of indices, element \( x_{u_{0}, w_{1}} \) must be \( \perp \).

**Theorem 4.** \((M, \bullet, I)\) is a commutative monoid.

The proof is similar to this of Theorem 3, without the last part, i.e., as above, we can check that \((M, \bullet, I)\) is a commutative monoid, but it is not a group because the fact that there is not a set \( Y \) that for some non-empty set \( X \) to be valid \( X \cup Y = \emptyset \).

### III. An Example of Difference Operations in the Big Data Environment

Hortonworks Data Platform (HDP) is an open source framework for distributed storage and processing of huge datasets retrieving from different sources. HDP is used to discover insights from structured and unstructured data in the cloud or on-premises. It includes Big Data tools as Hadoop,
Yarn, MapReduce, Hbase, Hive, Flume, Kafka, Druid [5]. The example of the difference operations is performed in the environment of Apache Hive. Apache Hive is a data warehouse used for reading, writing, and managing large amounts of data stored in distributed storage in SQL. Apache Hive flexibility can be extended using the used-defined functions (UDF) [1]. In the current investigation a data warehouse for product sells is implemented. The tables are uploaded using previously prepared csv files. Apache Hive supports only the sets operations Union All and Union [Distinct]. The Intersect and Except (Minus) operations are not included. In the current investigation for the first time the new index matrix operations performing operation difference will be applied by the analogy of the SQL JOIN clause and the logical operator NOT EXISTS in the Big Data environment. The tables Customers and Employees from bigdatadb data warehouse (Fig. 3). The authors will not compare the indexed field Number – the comparison will be performed using the columns for the first name and the last name of the tables. There are from the same data type. The tables Employees (Fig. 4) and Customers (Fig. 5) have the following form:

An example of the standard JOIN operation has the following form:

```sql
SELECT FName, Lname, CFname, CLName
FROM Employees
JOIN ProductSales
ON Employees.Number=ProductSales.EmployeeN
JOIN Customers
ON Customers.CNumber=ProductSales.CustomerN
```

![Fig. 3. Data warehouse bigdatadb view in Apache Hive environment](image)

![Fig. 4. Table of Employees](image)
The result presents the names of the customers and the employees. The SQL query in the Apache Hive environment is presented on the Fig. 6.

The simulation of difference operation is performed using the **NOT EXIST** logical operator in the **SELECT** statement and comparison of the desired columns from the tables in the sub-query (Fig. 7).

```sql
SELECT *
FROM customers
WHERE NOT EXISTS (SELECT *
FROM employees
WHERE (employees.FName = customers.CFname
AND employees.LName = customers.CLname));
```

The result of the query presents customers not included in the list of the employees. The columns for first names and last names are compared.

The same query with replaced tables in the query and sub-query can be executed to receive the employees not included in the table of the customers (Fig. 8).

The result of the query is presented on the Fig. 10. It combines the employees that are not customers and the customers that are not employees.

The combination of the previous queries using the **UNION** operator is executed (Fig. 9).
IV. CONCLUSION

Two new operations are introduced in the paper. In future, other properties of them will be studied and some other applications will be discussed. A part of them will be related to Big Data and Data Mining as continuation of the discussed in the present paper.

ACKNOWLEDGMENT

The authors acknowledge the support from the project UNITe BG05M2OP001-1.001-0004 /28. 02.2018 (2018-2023).

REFERENCES