Square Roots of Total Boolean Matrices

Enumeration issues

Ch. Massouros[1] and Ch. Tsitouras[2]
Department of Applied Sciences
TEI of Chalkis
GR34400, Psahna, Greece
[1]masouros@teihal.gr [2]tsitoura@teihal.gr

S. Voliotis[3]
Department of Electrical Engineering
TEI of Chalkis
GR34400, Psahna, Greece
[3]svoliotis@teihal.gr

***Abstract*— This paper studies the square roots of total Boolean Matrices. The basic matrices are introduced and it is proved that all the square roots of the total Boolean Matrix can be constructed from the basic ones. Next the basic matrices of order 2, 3, 4, 5 are enumerated and they are connected with directed graphs. Finally some issues are discussed concerning applications of self-organization and routing in wireless sensor networks**

Keywords-component; Binary relations, Boolean Matrices, Directed Graphs, Mathematica, Wireless Sensor Networks

I. Introduction

An algebraic system $(S,+,\cdot)$ is called a *semiring* if $(S,+)$ and $(S,\cdot)$ are semigroups connected by ring-like distributivity. The Boolean domain $B=\{0, 1\}$ becomes a semiring under the addition: $0+1=1+0=1+1=1$, $0+0=0$ and the multiplication: $0\cdot 0 = 0\cdot 1=1\cdot 0=0$, $1\cdot 1=1$. This semiring is called *binary Boolean semiring*. A *Boolean matrix* is a matrix with entries from the binary Boolean semiring. A square Boolean matrix is called *total* if all its entries are equal to 1. A Boolean matrix is called *good* if its square is the total matrix [1], i.e. the good matrices are the square roots of the total matrix. Every relation ρ in a finite set H with card$H = n$, can be represented by a Boolean matrix M_ρ and conversely every n×n Boolean matrix defines on H a binary relation. Indeed, if $H=\{a_1, \ldots, a_n\}$, then the $n\times n$ Boolean matrix is constructed as follows: the element (i, j) of the matrix is 1, if $(a_i, a_j) \in \rho$ and it is 0 if $(a_i, a_j) \notin \rho$ and vice versa. If in the Boolean matrix the i and j rows are interchanged and, at the same time, the corresponding i and j columns are interchanged as well, then the deriving new matrix and the initial one, are called *isomorphic*.

II. Basic Matrices

Definition 2.1**.** Basic matrix is a good matrix which is converted to a not good one, through the replacement of any unit entry to 0.

From the above Definition it derives that:

Proposition 2.1**.** Each basic matrix generates a family of good matrices, by replacing one or more 0 entries with 1.

One can observe that the families of good matrices which are generated by two non isomorphic basic matrices contain common members.

Proposition 2.2**.** All the good matrices are generated from the basic ones.

Now let $A=(a_{ij})_{nxn}$ be a Boolean matrix. In order for A to be a basic matrix, A^2 must be total. Thus all the sums $b_{ik}=\sum_{j=1}^{n} a_{ij}a_{jk}$, $i, k = 1,\ldots,n$ must be equal to 1, while having the minimum number of the products $a_{ij}a_{jk}$ equal to 1. So b_{11} becomes 1, if only a_{11} is equal to 1. With this granted, b_{21} equals to 1 if $a_{21}=1$ and so on.

Thus, letting a_{j1}, $j=1,\ldots,n$, be equal to 1, the first column of A^2 consists only of units, while A contains the minimum number of units.

Next, assuming, in addition to the above, that $a_{1j}=1$, the whole j-column of A^2 consists of units only. Thus:

Proposition 2.3**.** The nxn matrix which has all its first-row and first-column entries equal to 1, is basic and moreover it is the basic matrix with minimum number of unit entries.

Corollary 2.1. The nxn matrices which have all the entries of their i row and their i column, $i=1,\ldots,n$ equal to 1 are basic and isomorphic to each other.

Definition 2.2**.** A nxn matrix which has all the entries of its i row and its i column equal to 1, $i=1,\ldots,n$, is called minimum basic matrix.

Remark. Up to isomorphism there is only one non minimum basic matrix.

From the minimum basic matrix, we can construct all the other basic matrices. e.g. the proposition holds:

Proposition 2.4**.** Let A_i be the minimum basic matrix which has units in its i row and in its i column. Then:

i. if one entry a_{ij}, with $i\neq j$, is changed to 0, then the matrix becomes again basic if all the rest elements of the j column are changed to 1.

ii. if one entry a_{ji}, with $i\neq j$, is changed to 0, then the matrix becomes again basic if all the rest elements of the j row are changed to 1.

iii. if a_{ii}, is changed to 0, then the matrix becomes again basic if all the rest elements of the main diagonal are changed to 1.

Open Problem. The construction of an algorithm, which will produce all the nxn basic matrices from the minimum one, is a hitherto open question.

In [2] the good matrices of order 2, 3, 4, 5 are enumerated. For the self-sufficiency of this paper, we repeat these results in the Table I below:

TABLE I.

order	2	3	4	5
Boolean Matrices (BM)	16	512	65.536	33.554.432
Nonisomorphic BM	10	104	3.044	291.968
Good BM	3	73	6.003	2.318.521
Nonisomorphic Good BM	2	17	304	20.660

On the other hand, only a small number of non isomorphic basic Boolean matrices exist, compared to all the non isomorphic Boolean ones. The number of non isomorphic basic Boolean matrices is calculated with the use of Mathematica package [3] which is constructed for this purpose and consists part of the contents of this paper. The results of these calculations are given in the cumulative Table II below for the orders 2, 3, 4 and 5:

TABLE II.

order	2	3	4	5
Basic BM	2	21	316	23.409
Nonisomorphic basic BM	1	6	21	243

In the appendix we give four Mathematica functions that produce (i) Good BM (ii) Nonisomorphic BM (iii) Basic BM (iv) Nonisomorphic BM. So the second line of Table II is produced writing in Mathematica Frond End:

```
In[1]:=Basic[2]
Out[1]= {{{0,1},{1,1}},{{1,1},{1,0}}}
In[2]:=Length[%]
Out[2]=2
In[3]:=Length[Basic[3]]
Out[3]=21
```

etc.

Analogously the third line of Table II is derived writing:

```
In[4]:=Cardin[2]
Out[4]= {{{0,1},{1,1}}}
In[5]:=Length[Cardin[3]]
Out[5]=6
```

etc.

III. Basic Matrices and Directed Graphs

A directed graph consists of a finite set V, whose members are called vertices and a subset A of V×V whose members are called arcs. Thus A is a binary relation in V. So a Boolean matrix can be associated with every directed graph.

Proposition 3.1**.** If the associated Boolean matrix of a directed graph is a good one, then any one of its vertices can be visited from any other vertex through a path which consists of only two arcs.

In Figure 1, we draw the directed graphs which have associated Boolean matrices the 2x2 basic ones.

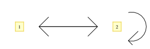

Figure 1. Non-IsomorphicBasic Directed Graph of order 2

In Figure 2, we draw the directed graphs which have associated Boolean matrices the 3x3 basic ones.

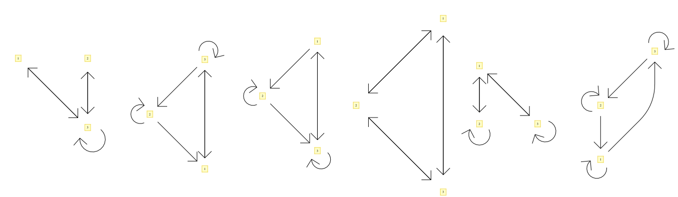

Figure 2. Non-IsomorphicBasic Directed Graph of order 3

Finally in Figure 3, we draw the directed graphs which have associated Boolean matrices the 4x4 basic ones.

IV. Comments on Good Boolean Matrices Applications

Consider the problem of organizing a network. We select a Good Boolean Matrix of order *n* and the associated directed graph with *m* directional arcs. In the first level we organize the nodes of the network in groups according to the directed graph. We construct a hierarchical structure of similar groups. We select one node from each group in level *i* and form groups in level *i*+1 with connections according to the directional graph.

Applying the proposed approach in a network consisting of *N* nodes, we can create a connectivity graph which limits the dimensions of the routing table providing connectivity at a bounded communication cost (measured in hops). Both the routing table dimensions and communication cost depend on the order *n* of the Boolean Matrix and the parameter *m*. Furthermore, this approach allows for easy nodes insertion and removal making it suitable for defining the connectivity graph in networks for networks of dynamically varying topology as well as for networks where the node positions may be fixed but the organization of the links among them is affected by fluctuating parameters. As such, the proposed approach can be exploited to realize self-organization and routing in wireless sensor networks where the available node resource both in terms of processing and memory are limited, thus calling of a reduced routing table [4] while the communication between the sensors and the base station that collects the sensed data has to be achieved in a bounded number of hops to save the overall system energy. Applying the proposed approach in such systems can also assist in the efficient support of mobility. Another purpose of applying this framework is the collection of indirect trust information which is important for the enhancement of security in these networks. For example, in [5], the application of a semi-ring framework has been proposed for the collection of trust information and has been proven to provide substantial flexibility. Even in high capacity optical networks, where the positions of the nodes are fixed, the ability to self-organize and arrange the available resources (e.g. tunable transceivers) to realize the light-paths that provide the desired performance has been identified as a key requirement for scalability and security [6]. Although optical nodes are quite powerful, the speed that the link organization algorithm should be executed mandates the design of an algorithm that converges after a fixed number of steps.

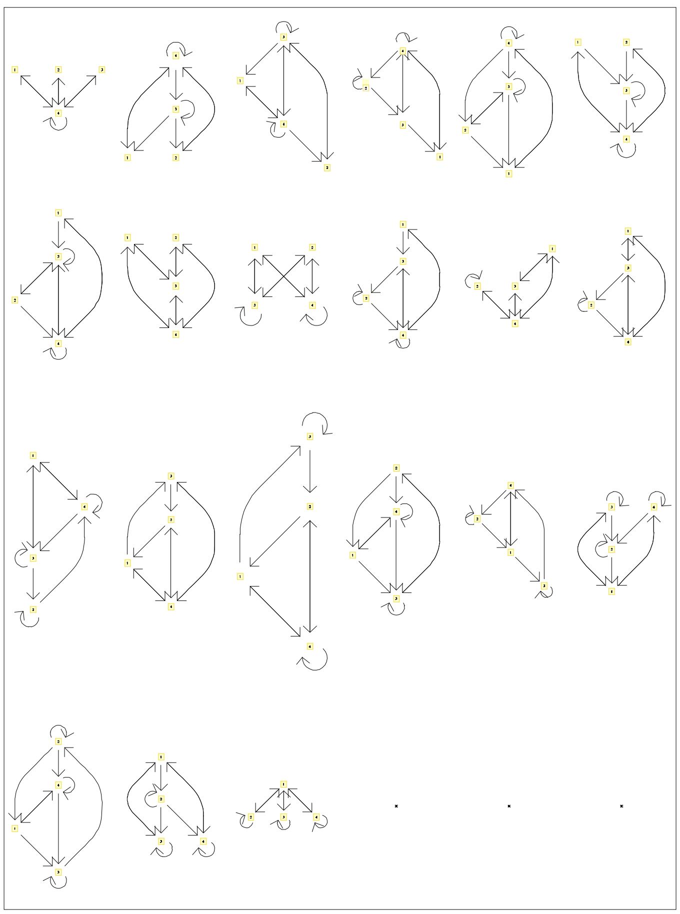

Figure 3. Non-IsomorphicBasic Directed Graph of order 4

V. APPENDIX

- The Mathematica function that returns the good matrices, i.e. The square roots of a *nxn* total Boolean Matrix:

```
Good[n_] := Module[{c, i1, z},
c = Tuples[Tuples[{0, 1}, n], n];
z = Table[Min[Flatten[c[[i1]].c[[i1]]]],
          {i1, 1, 2^(n*n)}];
Return[
Select[Transpose[{c,z}],#[[2]]>0&][[All, 1]]]
                   ]
```

- The Mathematica function that evaluates the basic matrices:

```
Basic[n_]:=Module[{oo,c,cprod,ind,exo},
oo={};
c=Good[n];
While[Length[c]>0,
      cprod=Table[Total[Total[c[[j1]]]],
                 {j1,1,Length[c]}];
      ind=Position[cprod,Min[cprod]][[1,1]];
      exo=Select[c,Max[c[[ind]]-#]==0&];
      oo=Append[oo,c[[ind]]];
      c=Complement[c,exo]
      ];
Return[oo]
                 ]
```

- The Mathematica function that evaluates the isomorphic matrices:

```
IsomorphTest1[a_List] := Module[{p, a1},
p = Permutations[Range[1, Length[a]]];
Return[
  Table[a1 = a;
        a1 = ReplaceAll[a1, a1[[All,
            Table[j2, {j2, 1, Length[a1]}]]]->
            a1[[All, p[[j1]]]]];
             ReplaceAll[a1, a1[[Table[j2, {j2,
                           1, Length[a]}]]] ->
              a1[[p[[j1]]]]],{j1,1,Length[p]}]
       ]
                             ]
```

- The Mathematica function that evaluates the isomorphic basic matrices:

```
Cardin[di_]:=Module[{h2,len,ooi,temp},
h2=Basic[di];
ooi={};
While[Length[h2]>0,
      temp=Union[IsomorphTest1[h2[[1]]]];
      len=Length[Union[temp]];
      ooi=Append[ooi,h2[[1]]];
      h2=Complement[h2,temp]
      ];
Return[ooi]
                   ]
```

REFERENCES

[1] P. Corsini: Binary relations and hypergroupoids, Italian J. Pure & Appl. Math., 7, pp. 11-18, 2000.

[2] Ch. G. Massouros, Ch. Tsitouras: Enumeration of hypercompositional structures defined by binary relations, Italian J. Pure & Appl. Math., to appear.

[3] S. Wolfram, The Mathematica Book, 5th ed., Wolfram Media, 2003.

[4] Aditya Karnik and Anurag Kumar, “Distributed Optimal Self-Organisation in a Class of Wireless Sensor Networks”, INFOCOM 2004, Hong-Kong, March 7-11, 2004.

[5] G. Theodorakopoulos and J. S. Baras, "On Trust Models and Trust Evaluation Metrics for Ad-Hoc Networks," IEEE Journal on Selected Areas in Communications (JSAC), Vol. 24, No. 2, Feb. 2006 pp. 318-328.

[6] Jae-seung Yeom, Ozan K. Tonguz, “Security and self-organization in transparent optical networks: an overview”, ACM International Conference on Access networks, Athens, Greece, 2006