Alternate Algorithm for DSM Partitioning

Power of Adjacency Matrix Approach

Using the power of the adjacency matrix approach, a binary DSM is raised to its n^{th} power to indicate which tasks can be traced back to themselves in n steps; thus constituting a cycle (Yassine *et al.*, 1999). However, Deo (1974) reported that when a matrix is raised to its 'n^{th}' power, the non-zero numbers represent the 'edge sequence' and not a 'path' (an edge may appear more than once in an edge sequence but not in a path).

For illustration, consider an activity DSM as seen in Figure 1(a). Initially topological sorting is performed to determine the acyclic activities (dependent and independent). This process is achieved by tracing the empty rows and empty columns for identifying the acyclic activities at the start and end of the project respectively (refer Figures 1(b) to 1(d)). Activities D and E does not contain any 'X' marks along the rows and hence they are moved to the top of the matrix as shown in Figure 1(b)-1(c). Similarly, activities C and J contain empty columns and they are moved to the bottom of the matrix and omitted from further consideration as seen in Figure 1(d). The active matrix after topological sorting is displayed in Figure 1(e).

This active matrix as represented in Figure 2(a) (same as Figure 1(e)) is converted to binary matrix by replacing the cells with 'X' marks as '1' and all other empty cells as '0' as shown in Figure 2(b). Now, the binary matrix is raised to its powers till 'n' ('n' – number of activities in the active matrix). Figures 2(c) to 2(h) represents the powers of the binary matrix.

Analyzing the non-zero value of the diagonal cell for instance, say at power '5' as in Figure 2(f). This implies that the length of edge sequence/cycle formed is '5'. Consider the non-zero value along the diagonal for activity F is '2'. This implies that there are '2' numbers of edge sequences (of length '5') can be formed starting with activity F. Hence, from the above observation, the '2' numbers of edge sequences of length '5' are FHIHIF and FHGHIF as visualized in Figure 1(f). The five-step cycles formed with activity F are not 'simple' cycles ('Simple' cycles are the cycles with no repetition of nodes except the first one). As there are repeating edges, the cycles identified from Figure 2(f) are not 'simple' cycles. For matrices with densely populated 'X' marks, the diagonal cell values adds confusion and it fails to identify the cycles/blocks. Hence, this method cannot identify the blocks for any network pattern.

But, the significance of the non-zero value along the diagonal at the n^{th} power can be utilized in detecting minimum length 'simple' cycles. This is because, edges cannot occur more than

	A	B	C	D	E	F	G	H	I	J	K
A											X
B	X				X						
C	X			X							X
D											
E				X							
F	X								X		X
G	X				X			X			
H						X	X		X		X
I								X			
J		X		X							X
K		X									

Figure 1(a) Activity DSM

	D	A	B	C	E	F	G	H	I	J	K
D											
A											X
B		X			X						
C	X	X									X
E	X										
F		X							X		X
G		X			X			X			
H						X	X		X		X
I								X			
J	X		X								X
K			X								

Figure 1(b) Topological Sorting

	D	E	A	B	C	F	G	H	I	J	K
D											
E	X										
A											X
B		X	X								
C	X		X								X
F			X						X		X
G		X	X					X			
H						X	X		X		X
I								X			
J	X			X							X
K				X							

Figure 1(c) Topological Sorting

	D	E	A	B	F	G	H	I	K	C	J
D											
E	X										
A									X		
B		X	X								
F			X					X	X		
G		X	X				X				
H					X	X		X	X		
I							X				
K				X							
C	X		X						X		
J	X			X					X		

Figure 1(d) Topological Sorting

	A	B	F	G	H	I	K
A							X
B	X						
F	X					X	X
G	X				X		
H			X	X		X	X
I					X		
K		X					

Figure 1(e) Active Matrix after Topological Sorting

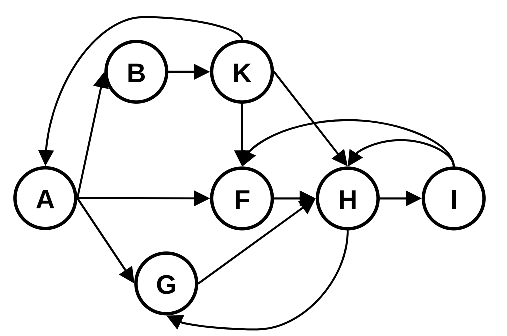

Figure 1(f) Graphical Representation of the Active Matrix

Figure 1 Block Diagram to Illustrate Topological Sorting

	A	**B**	**F**	**G**	**H**	**I**	**K**
A							**X**
B	**X**						
F	**X**					**X**	**X**
G	**X**				**X**		
H			**X**	**X**		**X**	**X**
I					**X**		
K		**X**					

Figure 2(a) Active Matrix after Topological Sorting

	A	B	F	G	H	I	K
A	0	0	0	0	0	0	1
B	1	0	0	0	0	0	0
F	1	0	0	0	0	1	1
G	1	0	0	0	1	0	0
H	0	0	1	1	0	1	1
I	0	0	0	0	1	0	0
K	0	1	0	0	0	0	0

Figure 2(b) Binary Matrix

	A	B	F	G	H	I	K
A	**0**	1	0	0	0	0	0
B	0	**0**	0	0	0	0	1
F	0	1	**0**	0	1	0	1
G	0	0	1	**1**	0	1	2
H	2	1	0	0	**2**	1	1
I	0	0	1	1	0	**1**	1
K	1	0	0	0	0	0	**0**

Figure 2(c) Power ‘2’

	A	B	F	G	H	I	K
A	**1**	0	0	0	0	0	0
B	0	**1**	0	0	0	0	0
F	1	1	**1**	1	0	1	1
G	2	2	0	0	2	1	1
H	1	1	2	2	**1**	2	4
I	2	1	0	0	2	**1**	1
K	0	0	0	0	0	0	**1**

Figure 2(d) Power ‘3’

	A	B	F	G	H	I	K
A	**0**	0	0	0	0	0	1
B	1	**0**	0	0	0	0	0
F	3	1	**0**	0	2	1	2
G	2	1	2	**2**	1	2	4
H	5	4	1	1	**4**	3	4
I	1	1	2	2	1	**2**	4
K	0	1	0	0	0	0	**0**

Figure 2(e) Power ‘4’

	A	B	F	G	H	I	K
A	**0**	1	0	0	0	0	0
B	0	**0**	0	0	0	0	1
F	1	2	**2**	2	1	2	5
G	5	4	1	**1**	4	3	5
H	6	4	4	4	**4**	5	10
I	5	4	1	1	4	**3**	4
K	1	0	0	0	0	0	**0**

Figure 2(f) Power ‘5’

	A	B	F	G	H	I	K
A	**1**	0	0	0	0	0	0
B	0	**1**	0	0	0	0	0
F	6	5	**1**	1	4	3	4
G	6	5	4	**4**	4	5	10
H	12	10	4	4	**9**	8	14
I	6	4	4	4	4	**5**	10
K	0	0	0	0	0	0	**1**

Figure 2(g) Power ‘6’

	A	B	F	G	H	I	K
A	0	0	0	0	0	0	1
B	1	0	0	0	0	0	0
F	7	4	**4**	4	4	5	11
G	13	10	4	**4**	9	8	14
H	18	14	9	9	**12**	13	25
I	12	10	4	4	9	**8**	14
K	0	1	0	0	0	0	0

Figure 2(h) Power ‘7’

Figure 2 Block Diagram to Illustrate Power of Adjacency Matrix Approach

once (other than the first and last) in minimum length edge sequences and hence the cycles formed are always ‘simple’.

Proposed Partitioning Procedure

The algorithm or procedure for the formation of partitioned DSM is represented in the form of a flowchart in Figure 3 and is explained below:

1. Consider an activity DSM.

2. Check for the presence of any ‘X’ mark along the upper diagonal of the matrix. If there are no marks along the upper diagonal it implies that the matrix is partitioned. Hence, stop the procedure or else continue with step 3.

(a) Topological Sorting

3. Check for empty rows in the matrix and move all those empty rows to the top of the matrix and the corresponding columns to the left of the matrix and omit those activities for further consideration.
   - The remaining activities in the matrix form the active matrix.

4. For the reduced (active) matrix, check for any empty columns and move all those empty columns to the right and the corresponding rows to the bottom of the active matrix and block those activities from further consideration.

5. Repeat steps 3 & 4 until there are no empty rows & columns in the active matrix

(b) Identification of Loops

6. Select the activities in the active matrix along with the dependency relationships for identifying loops

Loops are identified by the repeated process of powers of adjacency matrix and condensation of the strongly connected components.

(i) Adjacency Matrix Multiplication

7. Convert the active matrix into a binary matrix (adjacency matrix) by replacing the ‘X’ marks with ‘1’ and all other empty cells as ‘0’.

8. Raise the power of the adjacency matrix until you identify any non-zero entry along the diagonal. The maximum power can be anywhere till the order of number of activities in the current active matrix.
   - If there occurs any non-zero entry along the diagonal, stop the process or else continue the matrix multiplication until the power is raised to ‘n’, where n is the number of activities in the current active matrix.
   - The resultant matrix where the matrix multiplication is stopped is the end matrix.

9. On analyzing the end matrix, the corresponding ‘power’ at which the end matrix is identified signifies the length of the cycle and the ‘number’ formed along the diagonal during the matrix multiplication represents the number of cycles that particular activity is involved in.

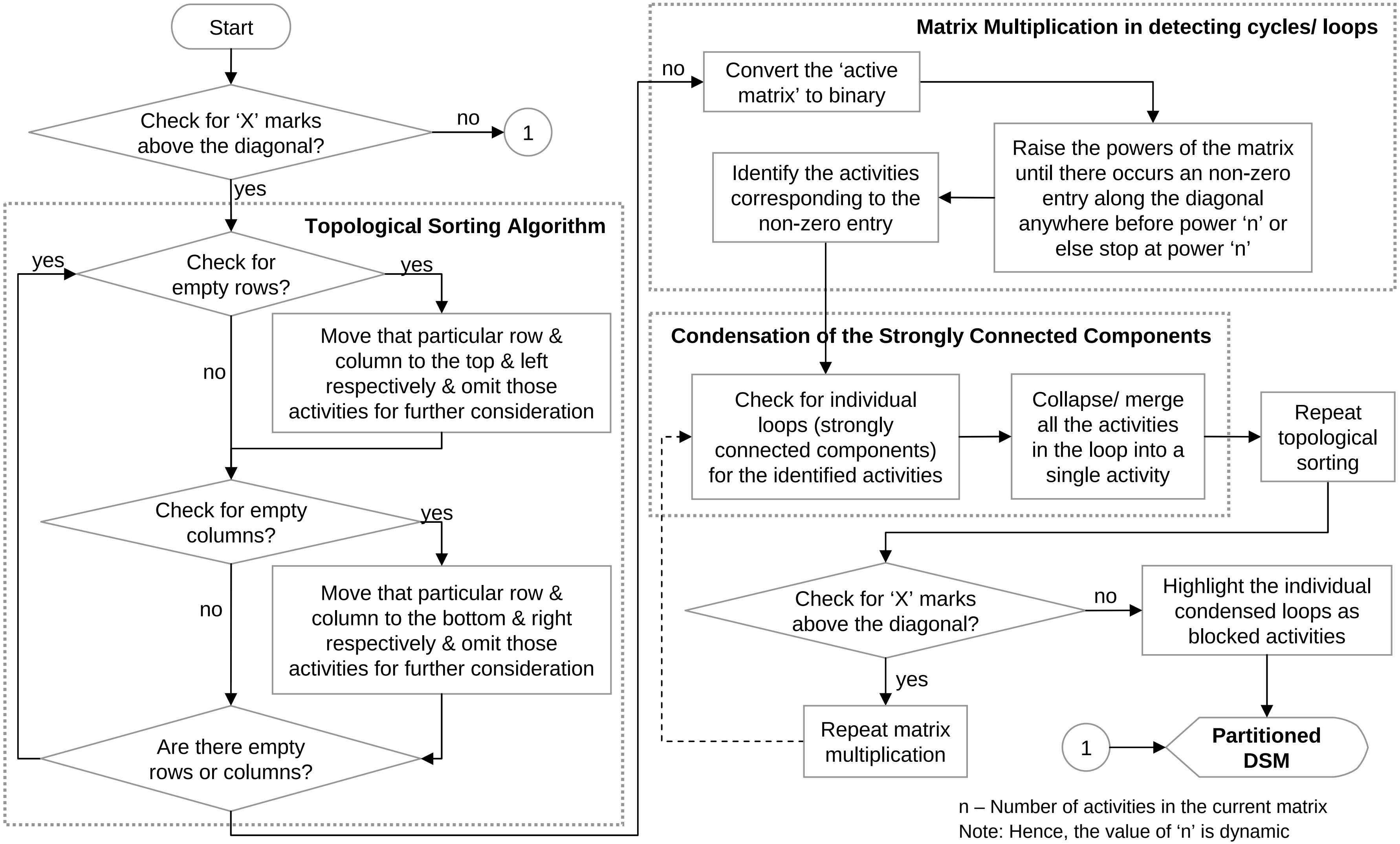

Figure 3 Flowchart for Proposed Partitioning Algorithm

- o If the power is square, the corresponding length of the cycle is 2; if it is cube, the corresponding cycle length is 3 and so on.

Consider an active matrix of size ‘4X4’. On raising the binary matrix to its powers, say at cube, if non-zero entry occurred along the diagonal, the matrix multiplication process would be stopped. The ‘cube’ matrix forms the end matrix. On analyzing the end matrix, if the number along the diagonal for an activity say ‘H’ was ‘2’ it implies that activity ‘H’ is involved in two numbers of three-step cycle.

(ii) Strongly Connected Components & Condensation

10. Trace out the pairs/cycles among the identified activities.
11. Move those activities (which form a cycle) together and merge/condense all those activities together
    - o If the cycles are overlapping, bring all those activities together and merge them together

(c) Identification of Blocks

12. The condensed matrix forms the present active matrix
13. Repeat topological sorting for the condensed matrix and check for any upper diagonal marks. If there are marks above the diagonal it implies the presence of loops and hence repeat the process from step 7 or else continue to step 14
14. The condensed activities represent blocks and the resultant matrix is the partitioned DSM.

Illustration

The example shown in Figure 1(a) is used for the present illustration. After topological sorting, the matrix of 11 activities reduces to 7 activities as shown in Figure 1(d). Remaining activities form the active matrix as revealed in Figure 1(e).

This active matrix is converted to a binary matrix as seen in Figure 4(a). On raising the binary matrix to its powers, non-zero value occurred along the diagonal at power ‘2’ as in Figure 4(b) for activities G, H and I. This implies that activities G, H and I are involved in a cycle of length ‘2’ (interdependent). The numbers along the diagonal for the above activities was 1, 2, and 1 and it corresponds to the number of cycles each activity is involved in. The above concepts imply that activities G and I are each involved in one cycle of length ‘two’ and activity H is involved in two cycles of length ‘two’. The two ‘two-step’ cycles are traced out as G-H-G and I-H-I. Since activity H is common in both the loops; all the three activities had been combined together as a single activity ‘GHI’ through condensation process and is illustrated in Figure 4(c).

Perform topological sorting to detect any acyclic activities. If the matrix still has marks above the diagonal, repeat the steps such as matrix multiplication, condensation and topological

	A	B	F	G	H	I	K
A	0	0	0	0	0	0	1
B	1	0	0	0	0	0	0
F	1	0	0	0	0	1	1
G	1	0	0	0	1	0	0
H	0	0	1	1	0	1	1
I	0	0	0	0	1	0	0
K	0	1	0	0	0	0	0

Figure 4(a) Binary (Active) Matrix

	A	B	F	G	H	I	K
A	**0**	1	0	0	0	0	0
B	0	**0**	0	0	0	0	1
F	0	1	**0**	0	1	0	1
G	0	0	1	**1**	0	1	2
H	2	1	0	0	**2**	1	1
I	0	0	1	1	0	**1**	1
K	1	0	0	0	0	0	**0**

Figure 4(b) Power ‘2’

	A	**B**	**F**	**GHI**	**K**
A					**X**
B	**X**				
F	**X**			**X**	**X**
GHI	**X**		**X**		**X**
K		**X**			

Figure 4(c) Active Matrix after Condensation & Topological Sorting

	A	B	F	GHI	K
A	0	0	0	0	1
B	1	0	0	0	0
F	1	0	0	1	1
GHI	1	0	1	0	1
K	0	1	0	0	0

Figure 4(d) Binary Matrix

	A	B	F	GHI	K
A	**0**	1	0	0	0
B	0	**0**	0	0	1
F	1	1	**1**	0	2
GHI	1	1	0	**1**	2
K	1	0	0	0	**0**

Figure 4(e) Power ‘2’

	A	**B**	**FGHI**	**K**
A				**X**
B	**X**			
FGHI	**X**			**X**
K		**X**		

Figure 4(f) Active Matrix after Condensation

	A	B	FGHI	K
A	0	0	0	1
B	1	0	0	0
FGHI	1	0	0	1
K	0	1	0	0

Figure 4(g) Binary Matrix

	A	B	FGHI	K
A	**0**	1	0	0
B	0	**0**	0	1
FGHI	0	1	**0**	1
K	1	0	0	**0**

Figure 4(h) Power ‘2’

	A	B	FGHI	K
A	**1**	0	0	0
B	0	**1**	0	0
FGHI	1	1	**0**	0
K	0	0	0	**1**

Figure 4(i) Power ‘3’

	ABK	**FGHI**
ABK		
FGHI	**X**	

Figure 4(j) Active Matrix after Condensation

	ABK	**FGHI**
ABK		
FGHI	**X**	

Figure 4(k) Final Condensed Matrix after Topological Sorting

	D	**E**	**A**	**B**	**K**	**F**	**G**	**H**	**I**	**C**	**J**
D											
E	**X**										
A					**X**						
B		**X**	**X**								
K				**X**							
F			**X**		**X**				**X**		
G		**X**	**X**					**X**			
H					**X**	**X**	**X**		**X**		
I								**X**			
C	**X**		**X**		**X**						
J	**X**			**X**	**X**						

Figure 4(l) Partitioned DSM

Figure 4 Block Diagram for Formation of Partitioned DSM (Proposed Approach)

sorting to form the final condensed matrix and is illustrated through Figures 4(d) to 4(k). Once the matrix multiplication cannot identify any loops, the above process is stopped. The condensed activities are unmerged into the normal activities and this matrix forms the partitioned DSM and is as shown in Figure 4(l). This partitioned DSM reveals two blocks through the above approach.

Discussion & Conclusion

Mathematical proof for the proposed algorithm is attempted (Maheswari, 2006).

This algorithm, which is based on powers of adjacency matrix, can partition any matrix COMPLETELY.

The drawback with this approach is its incapability in identifying ALL the individual cycles/loops (it can identify only the minimum length cycle available in a particular block). Further, it is not as EFFICIENT as available alternate algorithms. Hence, this algorithm is not recommended for developing DSM software.

References

1. Deo, N. (1974) *Graph Theory with Applications to Engineering and Computer Science*, Prentice Hall – India.
2. Maheswari, J. U. (2006) Modeling Activity Sequencing for Construction Projects using Dependency Structure Matrix, *PhD Thesis*, IIT Madras.
3. Yassine, A., D. Falkenburg, and K. Chelst (1999) Engineering Design Management: An Information Structure Approach. *International Journal of Production Research*, 37 (13), 2957 – 2975.