



GLOBAL JOURNAL OF SCIENCE FRONTIER RESEARCH: F
MATHEMATICS AND DECISION SCIENCES
Volume 21 Issue 2 Version 1.0 Year 2021
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 2249-4626 & Print ISSN: 0975-5896

A Note on Simulating Predecessor-Successor Relationships in Critical Path Models

By Gregory L. Light

Providence College

Abstract- For any n entities, we exhaust all possible ordered relationships, from rank (or the highest number of connections in a linear chain, comparable to matrix rank) 0 to $(n - 1)$. As an example, we use spreadsheets with the “RAND” function to simulate the case of $n = 8$ with the order-length = 3, as from a total of 10000 possibilities by the number of combinations of selecting 2 (a pair of predecessor-successor) out of 5 ($= \text{card}\{A, B, C, D\} + 1$) matching-destinations followed by an exponentiation of 4 ($= 8 - \text{card}\{A, B, C, D\}$). Since the essence of this paper is about ordered structures of networks, our findings here may serve multi-disciplinary interests, in particular, that of the critical path method (CPM) in operations with management. In this connection, we have also included, toward the end of this exposition, a linear algebraic treatment that renders a deterministic mathematical programming for optimal predecessor-successor network structures.

Keywords: CPM simulation, directed graphs, ordered networks, circuitry designs, logical flows, transmission patterns, optimal organizational rank.

GJSFR-F Classification: MSC 2010: 91G50



Strictly as per the compliance and regulations of:



RESEARCH | DIVERSITY | ETHICS



A Note on Simulating Predecessor-Successor Relationships in Critical Path Models

Gregory L. Light

Abstract- For any n entities, we exhaust all possible ordered relationships, from rank (or the highest number of connections in a linear chain, comparable to matrix rank) 0 to $(n - 1)$. As an example, we use spreadsheets with the "RAND" function to simulate the case of $n = 8$ with the order-length = 3, as from a total of 10000 possibilities by the number of combinations of selecting 2 (a pair of predecessor-successor) out of 5 (= $\text{card}\{A, B, C, D\} + 1$) matching-destinations followed by an exponentiation of 4 (= $8 - \text{card}\{A, B, C, D\}$). Since the essence of this paper is about ordered structures of networks, our findings here may serve multi-disciplinary interests, in particular, that of the critical path method (CPM) in operations with management. In this connection, we have also included, toward the end of this exposition, a linear algebraic treatment that renders a deterministic mathematical programming for optimal predecessor-successor network structures.

Keywords: CPM simulation, directed graphs, ordered networks, circuitry designs, logical flows, transmission patterns, optimal organizational rank.

I. INTRODUCTION

In simulating alternative optimal solutions of a critical path model [1], it is a simple matter to vary the needed time durations for the activities of the project in question, but it becomes more challenging to make variations in the involved predecessor-successor relationships as the geometric patterns of the associated network flow charts lend to overwhelming varieties. We thus consider the apparatus of the activity-node incidence matrix composed of entries $-(\text{activity } i, \text{incidence time node } j)$ in values $-1, 0$ and $+1$, but this is where ambiguities arise: the "start time" is ill-defined since activity i can start at an earliest time or else at a later time point [2]. The fact of the matter is that the time interval in any pair of time nodes (points) is totally independent of the needed time interval for activity i to be completed. To be sure, this predecessor-successor relationship can be devoid of the connotation of time altogether [3], [4]. For example, of eight relatives (activities) in a family gathering with the longest lineage being from great grandfather (activity A), to grandfather (activity B), to father (activity C), and to son (activity D), how many incidence matrices (modulo permutation of the eight activity-labels) are possible? That is, we are pursuing the general theme of network or organization structure that has a directional orientation.

As such, in this paper we define any time point (that is yet to be solved for in a critical path model, "CPM") to be the contact time from some predecessor(s) to its (their) successor(s), which coexist in the *same* (time) column respectively as $1(s)$ and -

Author: Department of Finance, Providence College, Providence, Rhode Island 02918 USA. e-mail: glight@providence.edu

$I(s)$ of the incidence matrix, with the meaning of “contact” being the act of transferring the finished work from a predecessor to any (all) of its successors. Every activity has thus exactly two contact time points, appearing as (-1) and $(+1)$ in its associated row amid all other O s. In this way, by subtracting an earlier time point from a later time point, we obtain a time interval, which we constrain to be greater than or equal to any given values.

As our study here is about network or, in general, graph theory, what we have observed may have multi-disciplinary applications, such as traffic control [5], [6], civil engineering [7], electric/data-logic circuitry [8], [9], and neural network [10]. From an extensive literature research, we did not find any similar work that used spreadsheets to simulate predecessor-successor relationships, with the closest being [11].

In the next section, we will first obtain a formula that calculates the total number of activity-node incidence matrices given some n activities and then show by an example how to simulate all the possible activity-node incidence matrices for the case of five contact time points of eight activities. Lastly, we will close with a summary remark in Section 3, where we outline a mathematical schema to solve for an optimal network or organization structure in general.

II. ANALYSIS

Without loss of generality, we consider a project comprising of δ activities, $A, B, C, D, E, F, G,$ and $H,$ and pursue the total number of network possibilities as built on predecessor-successor relationships in the context of critical path analysis. Then a familiar spreadsheet treatment is, as an example, to enter $-I$ for activity C in the row of C that intersects a column as associated with a (yet-to-be-solved-for) time point in a tabular array, and $+I$ in the same row of another column for a later time point. In this way the “sum product” of Row C with the header time row gives the time interval between the “contact time” of C with its immediate predecessor(s) and that with its immediate successor(s); this time interval is to be constrained to be greater than or equal to the needed time interval for C to be completed. The objective is then to minimize the ending time point of the “last activity/activities” (which has/have no successors). A natural question arises: viz., how many network flowcharts of the predecessor-successor relationships are possible? This answer can be found by a consideration of the number of columns as the contacting time points, where the least is 2 - - when all the eight activities have no predecessors or successors with $-I$ s all in the first column headed by time 0, and I s all in the second column headed by the longest time needed by the activities, and also where the greatest is 9 - - when the δ activities form a linear chain of predecessor-successor relationship, from A to $B,$ to $C,$..., and to $H.$ As such, the task now is to pursue the cases of the number of columns from 3 to $\delta.$

For the case of exactly 3 columns, as an illustration, we may have the following spreadsheet depiction:

Ref

8. Keller, J., Rauber, T. and Rederlechner, B., Scalability Analysis for Conservative Simulation of Logical Circuits, *VLSI Design*, 9 (3), 1999, 219-235.

Activity	0		
A	-1	1	0
B	0	-1	1
C	-1	0	1
D	-1	0	1
E	-1	1	0
F	-1	1	0
G	-1	0	1
H	0	-1	1

indicating the existence of precisely two contact time points. As such, there must exist two activities that form a linear chain; find such two activities and label them by *A* and *B*. Since each activity row has one (-1) and one (+1), by combinations of selecting two cells for these two non-zero numbers from three available cells in a row, we obtain 3 varieties for any of the remaining six rows, from *C* to *H*, i.e., $3^6 = 729$ ways. Incidentally, the above display, as one of the 729 possibilities, translates to

Activity	Immediate predecessors
A	-
B	A, E, F
C	-
D	-
E	-
F	-
G	-
H	A, E, F

Similarly, for the case of exactly 4 columns, we may have, as an example, the following spreadsheet depiction:

Activity	0			
A	-1	1	0	0
B	0	-1	1	0
C	0	0	-1	1
D	0	-1	0	1
E	-1	0	1	0
F	-1	1	0	0
G	0	0	-1	1
H	0	-1	1	0

showing the existence of a combination of $C(4, 2)$ that is to be raised to the 5th power, i.e., a total of 7,776 ways. It is now clear that the grand total number of the predecessor-successor relationships of any eight activities is (in *M.S. Excel* notation):

$$C(2, 2)^7 + C(3, 2)^6 + C(4, 2)^5 + C(5, 2)^4 + C(6, 2)^3 + C(7, 2)^2 + C(8, 2)^1 + C(9, 2)^0$$

$$= 1 + 729 + 7776 + 10000 + 3375 + 441 + 28 + 1 = 22351;$$

thus, for any $n > 1$ activities, we have the total possible number of the activity-node incidence matrices equal to:

$$\sum_{m=2}^{n+1} C(m, 2)^{n-m+1} .$$

We now present a way to simulate the predecessor-successor relationships for eight activities as above and for the case of five columns, which incidentally has the greatest number of the ordering varieties - - 10,000, among which being:

Activity	0				
A	-1	1	0	0	0
B	0	-1	1	0	0
C	0	0	-1	1	0
D	0	0	0	-1	1
E	-1	1	0	0	0
F	-1	1	0	0	0
G	0	0	0	-1	1
H	0	-1	0	0	1

where the first four rows are (fixed) values and the remaining four rows carry the same set of five formulas; in matrix notation, we have for activity *E*

- for entry (5, 1) = -1 (which was in cell "\$D44"):=ROUND(RAND(),0)
- for entry (5, 2) = 1:=IF(\$D44=-1, ROUND(RAND(),0),-ROUND(RAND(),0))
- for entry (5, 3) = 0:=IF(AND(MIN(\$D44:E44)=0, MAX(\$D44:E44)=0), -ROUND(RAND(),0), IF(MAX(\$D44:E44)=1,0, ROUND(RAND(),0)))
- for entry (5, 4) = 0:=IF(AND(MIN(\$D44:F44)=0,MAX(\$D44:F44)=0), -1, IF(MAX(\$D44:F44)=1,0,ROUND(RAND(),0)))
- for entry (5, 5) = 0:=IF(MAX(\$D44:G44)=1,0,1)

Note that one can control the probabilities of the distribution of -1, 0, and 1 over the available cells in a row by suitably modifying the formula *RAND*. With the needed time durations of the activities also randomly generated, one proceeds to solve for a critical path and, if so desired, uses a spreadsheet macro to generate any collection of examples, such as

Activity	0	40	53	78	106	106
A	-1	1	0	0	0	40
B	0	-1	1	0	0	13
C	0	0	-1	1	0	25
D	0	0	0	-1	1	28
E	0	-1	1	0	0	13
F	-1	0	0	0	1	106
G	-1	1	0	0	0	40
H	0	0	-1	1	0	25

or



Activity	0	26	62	106	156	156
A	-1	1	0	0	0	26
B	0	-1	1	0	0	36
C	0	0	-1	1	0	44
D	0	0	0	-1	1	50
E	0	-1	0	0	1	130
F	-1	1	0	0	0	26
G	0	0	-1	1	0	44
H	0	0	0	-1	1	50

III. SUMMARY

In this note, we have obtained a general formula that calculates the total number of predecessor-successor relationships of n entities, or equivalently, the total number of activity-node incidence matrices of n activities, and we have also shown a way to simulate them on spreadsheets. As such, one may gain insight into some “optimal” network structures given the individual activities’ required space time lengths. In fact, future studies may pursue such optimization by including the whole activity-node incidence matrix, of n (activities) by $(n + 1)$ (time points, beginning with 0), as decision variables in CPM problems with the following added constraints:

- (1) The $(1, 1)$ -entry = -1 , with the entire (first) column free from 1 ; this can readily be accomplished by constraining this column to be less than or equal to 0.
- (2) The $(n, n+1)$ -entry = 1 , with the entire (last) column free from -1 ; this can readily be accomplished by constraining this column to be greater than or equal to 0.
- (3) Each row contains exactly one -1 and one $+1$. Here, consider taking the exponential (*EXP* in *M.S. Excel*) of every entry for any of the n rows and constrain every row sum equal to $e^{-1} + e^1 + (n - 2)$.
- (4) For any column from the 2^{nd} to the n^{th} , it is either filled entirely with 0s or with at least one pair of $(+1, -1)$ for the simple reason that these $(n - 1)$ columns represent predecessor-successor contact time points. Here, consider: (i) adding a positive integer $a > 1$ uniformly to each of the n entries in any column for these $(n - 1)$ columns, (ii) taking the product of these n transformed entries for any of these $(n - 1)$ columns, $[(a + 1)^j a^i (a - 1)^k]$, $i + j + k = n$, $i, j, k \geq 0$, (iii) dividing the preceding product by $(a^2 - 1)$ and denoting the remainder by R , (iv) setting the column-sum-of-squares for each of these $(n - 1)$ columns equal to S , and (v) constraining the product $R \times S = 0$. Then, if $R = 0$, there exists at least one pair of $(1, -1)$ in this column; otherwise, $S = 0$, showing the entire column being of 0s. The case of $\{1, 0, \dots\}$, or $\{0, -1, \dots\}$, without the opposite sign, cannot occur since it would render $R \times S > 0$. The significance of such implementation is that one would be solving for the optimal “rank” (the highest order/length of a linear chain) of any network (organization). We caution, however, that to implement the above approach, one would need to use computing packages more robust than ordinary spreadsheets. In addition, the optimal solution cannot be unique since the trivial column(s) composed of all 0s can be placed anywhere within these $(n - 1)$ columns. This undesirable situation can be removed by constraining the column-sum-of-squares to be non-increasing from the 2^{nd} to the n^{th} column so that the “0-column(s)”

can precede only the $(n + 1)^{st}$ column. Note that this interchangeability among the “middle” $(n - 1)$ columns is owing to the additive commutativity of vector spaces, $a_2v_2 + a_3v_3 + \dots + a_nv_n$; i.e., the decision variables a_2, a_3, \dots, a_n as the $(n - 1)$ time points in *CPM* can be re-arranged. To be clear, for any critical activity (row) i , one has $a_j - a_k = \text{the needed time for the completion of } i$, with a_j and a_k uniquely solved in a *CPM*; however, for any predecessor-successor pair of non-critical activities, their contact time a_s has slack variability so that the employed algorithm may present different solutions of a_s upon permutations of a_2, a_3, \dots, a_n . In summary, one may thus arrive at a *CPM* with an optimal predecessor-successor structure in network designs for diverse fields.

REFERENCES RÉFÉRENCES REFERENCIAS

1. Conant, D. D., Examining critical and near-critical paths: an Excel-based classroom exercise, *J. Edu. Bus.*, 93(6), 2018, 285-291.
2. Winston, W. L. and Albright, S. C., *Practical Management Science, Spreadsheet Modeling and Applications*, New York, Duxbury (1997).
3. Sing, T. F., Ooi, J. T. L., Wong, A. L. and Lum, P. K. K., Network connectivity and office occupiers' space decision: the case of Suntec City, *J. Property Invest. Fin.*, 24 (3), 2006, 221-238.
4. Martínez, A. B., Galván, R. S. and Palacios, T. B., Study of factors influencing knowledge transfer in family firms, *Intang. Capital*, 9 (4), 2013, 1216-1238.
5. Iryo, T., Mori, H., Kitaoka, H., Ishida, M. and Asakura, Y., Traffic flow model of network simulators for estimation of CO, *J. Adv. Transp.*, 46 (3), 2012, 290-301.
6. Cheng, Y., Optimal train traffic rescheduling simulation by a knowledge-based system combined with critical path method, *Sim. Prac. Theo.*, 4 (6), 1996, 399-413.
7. Process connectivity in a naturally prograding river delta, *Water Resources Res.*, 53 (3), 2017, 1841-1863.
8. Keller, J., Rauber, T. and Rederlechner, B., Scalability Analysis for Conservative Simulation of Logical Circuits, *VLSI Design*, 9 (3), 1999, 219-235.
9. Mahmood, A., Herath, J. and Jayasumana, J., An Improved Data Flow Architecture for Logic Simulation Acceleration, *VLSI Design*, 2 (3), 1994, 259-265.
10. Karol, K. and Andrea, B., Development of control system by Studgard neural network simulator, *Acta Montanistica Slov.*, 8 (4), 2003, 217-219.
11. Kiran, K. and Maneeth. P., Optimization of Time Restriction in Construction Project Management Using Lingo and M. S. Excel, *Int. J. Engin. Res. Appl.*, 5 (9), 2015, 79-89.