Creative Construction Conference 2016

Floyd-Warshall in Scheduling Open Networks

Zoltán A. Vattai

Budapest University of Technology and Economics, Budapest, 1111, Hungary

Abstract

CPM, PERT, MPM, PDM are well known abbreviations and techniques extensively used at estimating and managing time performance of different kind of - amongst them of construction - projects. Common in them is that they are based on and demonstrated by the analogy of a special problem in Graph Theory namely the problem of finding The Longest Path(s) between two nodes in a weighted directed graph. It is less frequently mentioned that the problem has its pair as a 'dual' problem that is known as The Minimal Potentials' Problem interpreted on a set of potentials having pair-wise relative restrictions (lower and/or upper bounds) amongst the potentials. Main differences of the techniques mentioned at the beginning are in preparing and interpreting input and output data and in correspondence of graph elements and of time characteristics (events, processes, lead and lag times) of project components. It is also common in them that determining feasible solution(s) is usually based on a kind of roll-on typed algorithm (e.g. Dijkstra's Algorithm) calculating early and late times via series of consecutive steps starting from a base point (from start or from finish) increasing the set of examined elements of the graph step by step in an appropriate order (forward pass and backward pass), thus solving actually the 'dual' problem. Applying a modified Floyd-Warshall algorithm all-pair longest paths can be determined and identified, and all difficulties of and restrictions on composing the logical time model of the project (represented by the graph) can be eliminated except of the only thing: exclusion of positive loops in the weighted directed graph.
The paper discusses application of modified Floyd-Warshall algorithm to calculate the time model of the project with no concern on whether it has one or more starting and/or ending point(s), whether it includes logical loop(s) or not, whether it is a connected model or not, whether it necessitates positive (lower bound) or negative (upper bound) or multiple restrictions amongst the time data of the project elements or not – that is: to schedule open networks.

Keywords: Construction Management, Network Techniques, Open Networks, Floyd-Warshall, Scheduling.

1. Introduction

In early applications (CPM [3], PERT [4]) graph structures for scheduling had been restricted to a tightly defined topology referred as Network. By its definition a network is a directed weighted connected graph with one starting node, with one ending node, with no arrow loops and with no negative weights on the arrows. Necessity of these restrictions on graph structure can be ascribed mainly to early solution algorithms and to capabilities of early computers the applications had been run on. (In our context, later on, we use the terms “arrow”, “edge” and “directed edge” as synonyms. Similarly, when mentioning a “graph” or “network” we mean a directed weighted graph structure.)

It can be shown that without the rest of before mentioned restrictions, on general directed weighted graphs, valid and calculable time models can be constructed for use of any level of project and/or production management. Moreover, in widely known MPM [5,6] and PDM [4] techniques loops and negative weights have been implicitly integrated in the models resulting in no any unexpected, contrary and/or unsolvable conditions. Furthermore the practice of originating all initial steps/tasks from one single starting node in the model, and/or directing all finishing procedures/tasks into one single ending node may integrate unintended and misguiding information when modelling progresses in multi-project management context.

The need for revising restrictions of „traditional“ network techniques emerged at a joint R&D project of Hungarian Railways Company (MÁV) and of Budapest University of Technology and Management (BME), 1989-93, aiming to develop a Computer Aided Decision Support System for planning and managing reconstruction and maintenance works on the Hungarian railways’ system.

The challenge the management had to face was the task of Permanent Scheduling of works on a three-years slipping time-span looking over thousands of jobs with accuracy of minutes. No expressed start, no expressed end, widely diverging responsibilities, dispersed locations all around the country, but one complex must-be-operating, under-traffic railway system and a restricted common pool of some significant specialized resource series.

Traditional Scheduling techniques (including traditional Network Techniques) proved to be insufficient. „The project to be scheduled was not a project."

2. The scheduling problem

The scheduling problem, with tasks of pre-set durations and with pre-set precedence relations amongst them can be derived from the primal-dual problem-couple of The Longest Path Problem on a weighted graph (CPM, PERT) and The Minimum Potentials' Problem with lower and/or upper bounds on differences of pairs of potentials (MPM/PDM).

Exposed or not, usual algorithms developed for to solve the scheduling problem are focusing on The Minimum Potential's Problem meanwhile executing a kind of Implicit Labelling Technique such as Dijkstra's greedy algorithm for finding the shortest paths on a graph [1]. (Calculations are started at the origin and are rolling towards the terminal node, then back – "forward pass" and "backward pass").

Setting The Longest Path Problem in the focus of examinations necessity of restrictions on the graph structure can be reduced radically. Length of any path on a weighted graph is defined as a pure sum of weights of the arrows constituting the given path. At a pure addition it is irrelevant in what an order the numbers are added together. Thus, no need for pre-set origin and for pre-set terminal node for calculation. Neither the count of numbers is relevant. Length and elements constituting the longest paths can be identified. And, knowing the longest paths time potentials can be assigned along them to the nodes and arrows as early and late times.

2.1. Scheduling with homogenous restrictions

Preserving the inter-relation between the Longest Path Problem and The Minimum Potentials' Problem the Scheduling Problem can be summarized as:

$$\pi_i \geq 0; \quad \forall i \quad i \in \mathrm{N} \tag{1}$$

$$\pi_j - \pi_i \geq \tau_{ij} \quad \forall ij \quad ij \in \mathrm{E} \tag{2}$$

$$\pi_{\max} \rightarrow \min \tag{3}$$

$$\tau_{ij} \geq 0 \quad \forall ij \quad ij \in \mathrm{E} \qquad \text{(In CPM/PERT models non-negative weights allowed)} \tag{4}$$

(Where N is set of nodes (i), E is set of edges (ij), τ_{ij} is the lower bound also presented by the weight w_{ij} on edge ij, π_i is the time potential to be assigned to the node i)

2.2. Homogenizing mixed restrictions

According to the rules of elementary algebra, multiplying inequality representing bound on difference of a pair of potentials by minus one, any upper bound can be equally substituted by a lower bound (reversing the direction of subtraction, that is direction of edge, and changing the sign of the limit value). Thus, a mixed bounding system can be transformed to a homogeneous one having lower bounds only.

$$\pi_j - \pi_i \leq \tau_{ij} \quad / \cdot (-1) \qquad \text{(upper bound)} \tag{5}$$

$$\pi_i - \pi_j \geq -\tau_{ij} \qquad \text{(upper bound turned to lower bound)} \tag{6}$$

Analogically, any fixed duration of a task can be set by a pair of lower and of upper bounds having the same limit values (τ_{sf}, as duration) between its start (s) and its finish (f).

$$(\pi_f - \pi_s = \tau_{sf}) \quad \equiv \quad (\pi_f - \pi_s \geq \tau_{sf}) \;\cup\; (\pi_s - \pi_f \geq -\tau_{sf}) \tag{7}$$

As a consequence of above, loop of directed edges is given (between the starting and finishing nodes of the task), negative weight is given (upper bounding for fixed duration), while analogy of the Longest Path Problem is still held and the model keeps calculable.

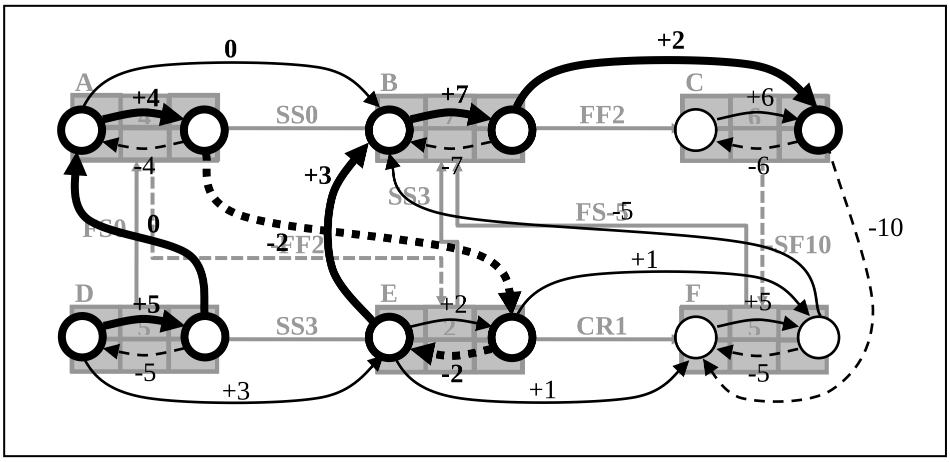

Figure 1. – A radiographic view (standard DiGraph representation) of a typical MPM/PDM network with fixed durations (boxes and arrows in grey in the background), with loops, positive (lower bounds - solid arrows) and negative (upper bounds - broken arrows) weights.

The only problem may emerge when any positive loop (sum of weights of edges of the loop is greater than zero) gets involved in the model. (Here we have to mention that researches are going on studying behavior of positive loops used as generators of periodically repeated jobs in production time models.)

A less-known interesting consequence of negative weights in the model is the phenomenon called Paradox Situation, when decreasing the duration of a task may result in an increment in completion time of the project. Such a situation may occur when edges with negative weights are involved in the longest path. (Consider MPM task E in Figure 1.)

3. The Modified Floyd-Warshall Algorithm

Implying its Computing Science origin the much cited Floyd-Warshall algorithms [2,8] are frequently explained as tripartite loop routines organized around a single core of simple calculations on a proper tabular representation of the graph. (This later is usually referred as weighted structure matrix or distance matrix of the graph).

Though it is rarely declared (or less evident) main idea of Floyd-Warshall routines is based on a simple triviality. Namely: considering a connected directed graph, if a path exists from node *i* to node *k* and also a path exists from node *k* to node *j*, then consequently a path do exists from node *i* to node *j* (at least the one via node *k*). In this context we do refer node *k* as a transfer node on a path from node *i* to node *j*. (Principle of transitivity of nodes)

Extending the above triviality it can be stated that in case of any graph there exist path from all the nodes from which path leads to a selected transfer node *k* to all the nodes to which path leads from the mentioned node *k*. It means that testing all the nodes of the graph as transfer nodes one by one we can gain certainty of existence of any and all the paths throughout the entire graph. It is also easy to see that tracking nodes this way all existing paths consisting of at least two edges on the graph get be considered once and only once. With information about the single edges in the other hand we can conclude that all-pairs analyses of the graph can be kept in hand this way. Some managerial problems solvable this way have been discussed by Vattai, 2006 [7].

3.1 General formulation of all-pairs calculations

For discussing the scheduling problem referred in the heading we use the denotations below:

G[N,E]	: refers to a graph (G) having the set of nodes (N) and the set of edges (E)
G[N,E,W]	: refers to a graph as above but having weights (W) on edges respectively
P	: refers to the set of all paths on the graph
P_{ij}	: refers to all paths from node i to node j on the graph
O^+	: refers to set of positive origins of the graph
T^+	: refers to set of positive terminal nods of the graph
ij	: refers to the edge from node i to node j on the graph
M	: marker value in tabular representation of graph, reads „no connection”
$\underline{\underline{W}}$	: initial tabular representation (weighted structure/adjacency matrix) of the graph
w_{ij}	: element of $\underline{\underline{W}}$ referring to the „weight” of the edge from node i to node j
n	: number of nodes of the graph
k	: index of outer cycle, also refers to transfer node actually selected
$\underline{\underline{A}}^k$	: transformed matrix representation of the graph in cycle k
a_{ij}^k	: element of $\underline{\underline{A}}^k$ referring to ij pair of nodes (connection from i to j)
a_i^k	: row vector i of matrix $\underline{\underline{A}}^k$
$á_i^k$	: column vector i of matrix $\underline{\underline{A}}^k$
i,j	: running indices of nodes, also referring to rows and columns of matrices

Using denotations above general routines of selecting nodes as transfer nodes one by one (outer cycle) and testing connections (inner cycles) together with performing necessary modifications if any (core) can be formulated as shown in Figure 2.

$$\underline{\underline{A}}^0 = \underline{\underline{W}} \qquad \{ \text{ initialization } \}$$

$$\underline{\underline{A}}^k = \Phi(\underline{\underline{A}}^{k-1}), \quad k = 1,2,\ldots,n \qquad \{ \text{ outer cycle } \}$$

Inner calculations of matrix-transformation function Φ: { inner cycles and core }

$$a_{ij}^k = \left\{ \begin{array}{ll} \varphi(a_{ik}^{k-1}, a_{kj}^{k-1}, a_{ij}^{k-1}) & | \quad i \neq k \quad j \neq k \quad a_{ik}^{k-1} \neq M \quad a_{kj}^{k-1} \neq M \quad a_{ij}^{k-1} \neq M \\ a_{ij}^{k-1} & \text{otherwise} \end{array} \right\} \quad \forall\, ij$$

where $\varphi(a_{ik}^{k-1}, a_{kj}^{k-1}, a_{ij}^{k-1})$ refers to a properly selected trhee-variable function (core)

Remark: at Floyd (1962) $M = +\infty$ and $\varphi(a_{ik}^{k-1}, a_{kj}^{k-1}, a_{ij}^{k-1}) = \min\{a_{ij}^{k-1}, a_{ik}^{k-1}+a_{kj}^{k-1}\}$

Figure 2. – Formulation of general routines of all-pairs analyses

3.2. All-pairs Longest Path

Algorithms for to calculate $\underline{\underline{A}}^n$ matrix reading all-pairs longest paths are differing from routines used for calculating all-pairs shortest paths in „sign” only as shown in Figure 3.

$$M = -\infty \qquad w_{ij} = \left\{ \begin{array}{ll} w_{ij} & | \quad ij \in E \\ M & \text{otherwise} \end{array} \right\} \quad \forall\, ij$$

$$\varphi(a_{ik}^{k-1}, a_{kj}^{k-1}, a_{ij}^{k-1}) = \max\{a_{ij}^{k-1}, a_{ik}^{k-1}+a_{kj}^{k-1}\}$$

Figure 3. – Marker value and core function for all-pairs longest distances

By modified Floyd-Warshall algorithm we actually produce a specific Transitive Closure of the graph with values of length of longest paths in all-pairs relation. Here we have to point at that any positive value in the diagonal of the resulting $\underline{\underline{A}}^n$ matrix invalidate the calculation indicating existence of at least one positive loop on the graph.

4. Deriving solution of Potentials' Problem

4.1. Deriving solution of Minimal Potentials' Problem - for closed networks

Having the valid specific transitive closure of the structure matrix of the graph Overall Execution Time (Π) of the project modelled and deadlines (time-potentials) such as Earliest Times (π_j) and Latest Times (π'_i) of events (represented by nodes of the graph) can be read in matrix $\underline{\underline{A}}^n$ almost directly.

$$\pi_j = \max\{0, \max_i a_{ij}^n\} \quad \forall j \tag{8}$$

$$\Pi = \max_j \pi_j = \max_{ij} a_{ij}^n \tag{9}$$

$$\pi'_i = \min\{\Pi, \min_j(\Pi - a_{ij}^n)\} \quad \forall i \tag{10}$$

4.2. Deriving solution of Potentials' Problem - for open networks

We refer a directed weighted graph as „Open Network" if it has more origins and/or more terminal nodes. To identify the chain of arrows of the longest distances on an open network we have to introduce the terms "Positive Origins", and "Positive Terminal Nodes".

A Positive Origin of a graph is a node with at least one leaving arrow of positive weight but with no entering arrow of positive weight. A Positive Terminal Node of a graph is a node with at least one entering arrow of positive weight but with no leaving arrow of positive weight.

It is easy to get convinced that the weight of the first and that of the last arrow of the Longest Path(s) on a Graph having at least one arrow with positive weight cannot be negative. (Leaving the arrows with negative weights from the beginning and/or from the end of a path would result in a „longer" path. Length of a path is pure sum of weights of its arrows.) So any Longest Path must lead from Positive Origin to Positive Terminal Point.

In case of a closed network in a feasible solution of the Minimum Potentials' Problem the minimum time-span between the only (positive) origin and the only (positive) terminal node is provided and it equals to the length of the longest path(s) between them.

To keep this analogy for open networks, that is to keep the correspondence between potentials and path lengths we have to modify the original Potentials' Problem:

$$\pi_i \geq 0; \quad \forall i \quad i \in N \tag{11}$$

(Non-negative potentials to be assigned to all nodes of the graph.)

$$\pi_{\min} = 0; \tag{12}$$

(Fix the minimum value of the potentials' system to zero.)

$$\pi_j - \pi_i \geq \tau_{ij} \quad \forall ij \quad ij \in E \tag{13}$$

(Difference of pairs of potentials are limited by lower bounds represented by the weights of edges ($\tau_{ij} = w_{ij}$) of the graph. No restriction on the value and/or sign of the weights.)

$$\pi_j - \pi_i \rightarrow \min \quad \forall ij \quad i \in O^+ \quad j \in T^+ \quad P_{ij} \in P \tag{14}$$

(The time-span between positive origins and positive terminal nodes between which at least one path exists should be at minimum.)

This way we eliminate false floats from the schedule that would be posed by the original (Minimum) Potentials' Problem. This elimination may get high importance in a multi-project management context such the one referred at the beginning (MÁV) where the Start and the Finish of individual projects inter-related are not mutual and should be handled individually. (Out of some common resources and/or some common interests the local projects have their own preferences.)

$$\pi_j = \max\{0, \max_i a_{ij}^n\} \quad \forall j \tag{15}$$

$$\pi'_i = \min\{\pi_j, \min_j(\pi_j - a_{ij}^n)\} \quad \forall i \quad j \in T^+ \quad a_{ij}^n \neq M \tag{16}$$

$$\pi_j = \max\{\pi'_i,\ \max_i\,(\pi'_i + a_{ij}^n)\} \qquad \forall\, j \qquad i \in O^+ \tag{17}$$

4.3. Identifying the Critical Path (Dominant Sub-Graph)

„Critical Nodes“ of the overall longest path (better said: of Dominant Sub-Graph – frequently referred as „Critical Path”) can be recognized by checking condition if the earliest and latest times equal to each other ($\pi_i = \pi'_i$), while „Critical Edges“ (all between critical nodes) can be recognized by checking if difference of time potentials at ending (π_j) and at originating (π_i) nodes of the edge equals to the weight (w_{ij}) of it.

$$\pi_i = \pi'_i \qquad \pi_j = \pi'_j \qquad \pi_j - \pi_i = w_{ij} \qquad ij \in E \tag{18}$$

All and any Longest Path(s) between any pair of nodes between which any path exist can also be identified with the help of the original structure matrix ($\underline{\underline{A}}^0$) of the graph and the transitive closure of it ($\underline{\underline{A}}^n$) produced by the modified Floy-Warshall Algorithm.

An edge (*ik* or *ij*) is the first or the only edge of a path of the given length (a_{ij}^n) between two nodes (*i* and *j*) if the length (weight) of the edge (a_{ik}^0 $k \neq j$) and the length of the rest of the path (without that edge) (a_{kj}^n) adds the length of the path (a_{ij}^n) considered or the length of the edge (a_{ij}^0) equals to the length of the path (a_{ij}^n) itself.

$$a_{ij}^n = a_{ik}^0 + a_{kj}^n \qquad \text{or} \qquad a_{ij}^n = a_{ij}^0 \tag{19}$$

Starting from the first node (*i*) of the path to direction of the last one (*j*) of it the edges of the path with the given length (a_{ij}^n) can be identified one-by-one on a forward-pass. This way length of paths and of edges are used as some kind of implicit labels.

Acknowledgement

The author would thankfully evoke the memory of his former tutors, later colleagues and friends three of whom we can never turn for help to any more. Members of the 1989-93 MÁV-BME R&D research team were: Tibor Vígh+, head of Office of Exposed Projects, Hungarian Railways Company; Károly Bacher+, József Monori+, László Neszmélyi and Zoltán András Vattai, tutors of Department of Construction Technology and Management, Budapest University of Technology and Economics.

References

[1] E. W. Dijkstra, A Note on Two Problems in Connexion with Graphs, Numerische Mathematic 1, 269-271, 1959
[2] Robert W. Floyd, Algorithm 97: Shortest Path, Communications of the ACM 5(6): 345, June 1962
[3] James E. Kelley Jr., Morgan R. Walker, Critical-path Planning and Scheduling, Proceedings of Eastern Joint Computer Conference, p. 160-173, Boston MA, December 1959
[4] Joseph J. Moder, Cecil R. Phillips, Edward W. Davis, Project Management with CPM, PERT and Precedence Diagramming, Van Nostrand Reinhold, New York, 1983
[5] B. Roy, Les problémes d' ordonnancement, Dunod, Paris, 1964
[6] B. Roy and B. Sussmann, Les problémes d' ordonnancement avec contraintes disjonctives, Note DS no 9. bis, SEMA, Montrouge, 1964
[7] Z. A. Vattai, Floyd-Warshall Again, 7th International Conference On Organization, Technology And Management In Construction, Zadar, Croatia, 2006
[8] S. Warshall, A Theorem on Boolean Matrices, Journal of the ACM, Vol 9, 1, 11-12, 1962