Project scheduling algorithms for construction projects with stretchable activities

L Mályusz and Z Vattai
Department of Construction Technology and Management, Budapest University of Technology and Economics
malyusz.levente@epk.bme.hu

Abstract. Project management requires increasingly complex, sophisticated scheduling models, preferably in the simplest possible way, for which the rapid expansion of computing capacity provides an increasing opportunity. In this paper, we examine deterministic models that, as a further development of the traditional CPM and PDM models (with four well-known minimum and maximum priority relationships) allow activity to be stretchable. The aim of the study is a) to implement different algorithms for time analysis on computers, b) to compare their speeds on large-scale real and artificial projects; c) to prepare proposals for selecting the best fitting algorithm(s) to the specific model(s). The comparison was based on real network. The results show that depending on the models, different algorithms perform well, so we recommend that different algorithms be implemented in the scheduling tools and let the tool decide which algorithms will be optimal for the computation time.

1. Introduction
Critical path calculation is an essential part of project scheduling problem. Since the beginning of the history of critical path more and more relevant mathematical models were developed for engineering project scheduling problem. In this study we use PDM schedule network with the four traditional precedence relationships (SS,SF,FS,FF) with an improvement to the technique, namely stretchable activities are allowed. The stretchable activities have a minimum and a maximum possible duration and their activity time can be between two these values. This study’s general aim is to make an efficient algorithm for this model. This study is a branch of a bigger research project was started in [1].

The complete time analysis that defines the project duration and the early/late start/finish dates for all activities for CPM network schedules was developed in 1957 and introduced to the scientific community in [2,3]. The more flexible PDM results from [4,5,6], among others.

Many algorithms transform traditional precedence relationships into SS relationships [7,8] before time analysis. Maximal SS relationship in its minimal equivalent form results from Roy [4].

Algorithms #1-#3 are iterative algorithms used when maximal relationships are in the network. Algorithm #1, developed by [9] based on the work of the shortest path algorithm for non-directed networks from [10,11]. The modified FIFO algorithm (Algorithm #2), which is explained later, is based on the idea presented by [10,12] for finding the shortest path on non-directed networks. The DEQUEUE implementation used in Algorithm #3 uses the idea of [13], who developed their algorithm for finding the shortest path in non-directed networks. Algorithm #2 and #3 have the same theoretical complexity as Algorithm #1 but are thought to be faster in linear-construction-like networks.
Algorithms #4/a-b are designed explicitly for Monte Carlo analysis based on algorithms #1-#3. In the first instance, they use Algorithm #1-#3, and they save the relationships’ calculation order. During the second phase, which involves all the remaining instances, the calculation uses the calculation order defined during the first instance, making the time for calculating a new instance much faster.

Several articles have compared different shortest path algorithms for different applications [14,15,16]; however, none of them investigated project schedules. This paper aims to study the characteristics of Algorithms #1-#3 and their modification developed for Monte Carlo simulation (Algorithm #4 a-b) on construction-like networks.

1.1. Algorithms

In this paper the following algorithms were used to solve the project scheduling problem:

1. algorithm generic label-correcting (critical path algorithm for cyclic network),
2. algorithm modified label-correcting with FIFO implementation,
3. algorithm modified label-correcting with Dequeue implementation,
4. a. two phased version of algorithm of 2,
4. b. two phased version of algorithm of 3.

Algorithms 4.a and 4.b are designed explicitly for Monte Carlo simulation. Algorithm 4.a includes Algorithm 2. for the first instance, with the only difference that the calculation sequence of arcs is stored in an array. This calculation order is used for the rest of the instances. A more detailed description of the algorithms is in [1,17].

This research aims to decide which algorithms perform better for time analysis in a presence of stretchable activities. The following two research objectives have been set to satisfy the goal:

Comparing the speed of the algorithms using the example project,
Evaluating the results and making recommendations.

In this study algorithms were investigated through a PDM network of a construction project. The original construction network was modified in accordance with the objectives of the research, namely we created new PDM networks (NET1, NET2,…) increasing the number of the stretchable activities based on the original PDM text.

1.2. Case Study

Our selected project was the architectural reconstruction of the Memorial Building of Nobel-laureate writer Imre Kertész. The project included removal of non-authentic contemporary features of the existing building, partial reconstruction of the load-bearing structures (floors, walls), reproducing historical features such as roofing, cladding, doors and windows, internal wooden wall covers, stairs etc., all kinds of finishes and building mechanics and electric systems and - of course - installing modern multimedia and exhibition facilities. Though the quantities were less than that of a larger family house the quality expectations and complexity of works necessitated a network time model to establish a logical guideline for managing the on-site works. Since the works were financed by public funds strict expectations against the time model were also set by the employer who managed the tendering and the public procurement procedure.

The winner contractor developed a network time model of a hundred-some technological processes (activities) and about 150 technological dependencies (predecessor links) among them together with some calendar restrictions ("employer's schedule", milestones) and some resource-related expectations. Since our examinations were focused on the lean inner logic of the "engineers' network" (engineers' logic) we prepared a kind of CPM network (containing start-to-start relations only) off the original MPM network model, free of any superposed non-network originated features such as calendar constraints, and cost- or resource limitations. Using the purified model, we systematically added the "newer" network elements one-by-one (such as stretchable activities, upper-bound typed limitations, non-positive loops) and studied their effect on the running time of calculations (forward pass and backward pass) by the upper-mentioned algorithms.
To characterize the topology of the network we studied the inner sub-chains and junction extremities of its graphical representation (the “Graph”).

We found that the contractors "original" time model involved 99 activities (including 20 milestones), 141 links, one source (originating node), one sink (terminating node), 5 positive sources (with no entering arrow of positive weight), 6 positive sinks (with no leaving arrow of positive weight), 39 FFmin links (27.66%), 92 FSmin links (65.25%), 8 SSmin links (5.67%). Orderliness (links in the upper triangle) was 85.82%, maximum number of links per task was 10, maximum number of predecessors per task was 7 and maximum number of successors per task was 6.

Considering the weights and the time potentials (time results) we found that the overall execution time was 270 t.u. (time units, here workdays), maximum duration 255 t.u. (on-site management), mean duration 26 t.u., maximum lag 260 (would be avoided in general), mean lag 4 t.u., maximum free slack 50 t.u., mean free slack 6 t.u., maximum total slack 50 t.u., mean total slack 15 t.u.

There was one single critical path in the network, involving 76.77% of the activities ("over-tied" or "rigid" network).

Keeping the core of original network our modifications (extra links, maximum-typed limitations) were superposed on that step-by-step.

### Table 1. Variants of the example and their parameters

| Number of stretchable activities | NET1 | NET2 | NET3 | NET4 | NET5 | NET6 |
|----------------------------------|------|------|------|------|------|------|
| Number of activities (after transformation) | 1    | 2    | 3    | 5    | 9    | 17   |
| Number of relationships (after transformation) | 99   | 100  | 101  | 103  | 107  | 115  |

### 1.3. Results

Main results are collected in Table 1 and Fig.1. Among the algorithms Dequeue is the slowest but two phased Dequeue has the shortest running time. The advantage of 4.b. is bigger if the calculated number of instances is bigger.

### Table 2. Average completion (running) time in nanoseconds for one instance, number of instances is 100000

| Number of stretchable activities | 1     | 2     | 3     | 5     | 9     | 17    |
|----------------------------------|-------|-------|-------|-------|-------|-------|
| 1. BF                            | 3005  | 3033  | 3147  | 3284  | 3450  | 4937  |
| 2. modBF                         | 2389  | 2406  | 2437  | 2492  | 2582  | 3116  |
| 3. Dequeue                       | 181182| 180094| 180920| 182846| 183767| 185821|
| 4a. 2phase mBF                   | 1827  | 1850  | 1928  | 1905  | 1945  | 2228  |
| 4.b.2phase Deq                   | 1711  | 1729  | 1729  | 1758  | 1827  | 1987  |
Fig. 1. Time in nanoseconds for one instance

Results (Table 1 and Fig. 1) show that for Monte Carlo analysis, Algorithm #4/b (based on DEQUEUE implementation) performs better than Algorithm #4/a. The most important practical consequence of this study is that Monte Carlo simulation for large construction-like networks containing stretchable activities can be performed within a couple of seconds.

2. Conclusion and further research
In our example 2phase Dequeue algorithm performed the best. We would like to improve these algorithms’ speed further and extend the capabilities of the models, namely improve the flexibility in order of activities.

The speed of different time analysis algorithms was studied in this paper to find the most suitable for analyzing construction-like schedule networks containing stretchable activities. Three algorithms (Algorithms #2-4) have, and their modifications for MC simulation (Algorithms #5/a-c) have been implemented for computers. Algorithms were investigated based on a real life construction project. We found that the two phase DEQUEUE algorithm performed better than the others. The algorithms were implemented in C++ (Visual Studio).

References
[1] Malyusz, L. & Hajdu, M. & Vattai, Z. Comparison of different algorithms for time analysis for CPM schedule networks; vol 127, July 2021, Automation in Construction, 2021, Available at: https://doi.org/10.1016/j.autcon.2021.103697
[2] Kelley, J.E. & Walker, M.R., 1959. Critical-path planning and scheduling. Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference on - IRE-AIEE-ACM ‘59 (Eastern). Available at: http://dx.doi.org/10.1145/1460299.1460318.
[3] Kelley, J.E., 1961. Critical-path planning and scheduling: Mathematical basis. Operations Research, 9(3), pp. 296–320. Available at: http://dx.doi.org/10.1287/opre.9.3.296.
[4] Roy, G.B., 1962 Graphes et ordonnancement (Graphs and scheduling), Revue Française de Recherche Opérationnelle n° 25, 4e trimestre pp323-33
[5] Fondahl, J. W., 1962. A Non-Computer Approach to the Critical Path Method for the Construction Industry, Technical Report #9. 2nd edition. Department of Civil Engineering, Stanford
University, Stanford, CA, 1962, p. 155 Available at: http://hdl.handle.net/2027/mdp.3901500453970 Accessed at: 03/05/2020

[6] IBM. 1964, Users’ Manual for IBM 1440 Project Control System (PCS)

[7] Kerbosc, JAGM & Schell, HJ., 1975, Network planning by the extended metra potential method (EMPM) TH Eindhoven. ORS, Vakgr. operationele research report, vol. KS-1.1, Technische Hogeschool Eindhoven. https://pure.tue.nl/ws/portalfiles/portal/4368315/104181.pdf.

[8] Elmaghraby, S.E. & Kamburowski, J., 1992. The Analysis of Activity Networks Under Generalized Precedence Relations (GPRs). Management Science, 38(9), pp. 1245–1263. Available at: http://dx.doi.org/10.1287/mnsc.38.9.1245.

[9] Hajdu, M., 1997. Network Scheduling Techniques for Construction Project Management. Nonconvex Optimization and Its Applications, p. 352 Kluwer Academic Publishers. Available at: http://dx.doi.org/10.1007/978-1-4757-5951-8.

[10] Bellman, R., 1958. On a routing problem. Quarterly of Applied Mathematics, 16(1), pp. 87–90. Available at: http://dx.doi.org/10.1090/qam/102435.

[11] Ford, L. R. Jr., 1956. Network Flow Theory. Paper P-923. Santa Monica, California: RAND Corporation. Accessed at: https://apps.dtic.mil/dtic/tr/fulltext/u2/422842.pdf on May 3, 2020.

[12] Moore, E.F., 1959. The shortest path through a maze. Proceedings of an International Symposium on the Theory of Switching 2-5 April, 1957, Part II. Cambridge, Massachusetts: Harvard University Press. 1959, pp. 285–292 MR 114710.

[13] Pape, U., 1974. Implementation and efficiency of Moore-algorithms for the shortest route problem. Mathematical Programming, 7(1), pp.212–222. Available at: http://dx.doi.org/10.1007/bf01585517.

[14] Ajwani, D. et al., 2018. An Empirical Comparison of k-Shortest Simple Path Algorithms on Multicores. Proceedings of the 47th International Conference on Parallel Processing, 2018, pp-1-12 Available at: http://dx.doi.org/10.1145/3225058.3225075.

[15] Xie, D. et al., 2016. Practical private shortest path computation based on Oblivious Storage. 2016 IEEE 32nd International Conference on Data Engineering (ICDE). Available at: http://dx.doi.org/10.1109/icde.2016.7498254.

[16] Wang, X.Z., 2018. The Comparison of Three Algorithms in Shortest Path Issue. Journal of Physics: Conference Series, 1087 (2018), pp. 022011. Available at: http://dx.doi.org/10.1088/1742-6596/1087/2/022011.

[17] Ahuja, R. K., Magnanti T.L., Orlin J.B., 1993. Network flows, Theory, Algorithms and Applications. Prentice Hall, New Jersey. [ISBN-13: 978-0136175490]

Acknowledgements
The research reported in this paper has been supported by the National Research, Development, and Innovation Fund (TUDFO/51757/2019-ITM, Thematic Excellence Program