https://doi.org/10.3311/PPar.13257

Periodica Polytechnica Architecture, 50(1), pp. 63–66, 2019

Calendars in Time-Cost Trade-Off

Helga Csordas[1*]

[1] Department of Construction Technology and Management, Faculty of Architecture, Budapest University of Technology and Economics, H-1111 Budapest, Műegyetem rkp. 3., Hungary
[*] Corresponding author, e-mail: hcsordas@ekt.bme.hu

Received: 05 October 2018, Accepted: 18 February 2019, Published online: 16 April 2019

Abstract

In project management, there are two main operation problems. Scheduling and cost optimisation. These are interrelated and have mathematically proven solutions for the basics. However, in case of applying arbitrary calendars, there may be generated such effects in scheduling that make the known time-cost trade-off model unusable. In consideration of these effects, this paper aims to apply known algorithms that have been successful for other problems.

Keywords

scheduling, calendar, time-cost trade-off

1 Introduction

A network model in civil engineering practice must be suitable to handle two features in scheduling.

The first is the possibility of changing process durations depending on their start times. This is the key to apply calendars.

The second one uses maximal constraints for activities and connections. This is useful and important in practice. In linear programming method, it is possible to give only minimal constraints. For applying maximal constraint, it must be converted by multiplying the assumption with (−1). It creates negative process time and turning back arc, which generates loops in high probability.

In case of these assumptions, there are no restrictions to apply the models either activity on edge (AOE) or activity on node (AON). Here, notations are related to the model AOE.

In this review, there are no restrictions for any of the two generalisations. The project is modelled on a $[N;A]$ digraph. Let N be the set of nodes, A be the set of arcs. Let s and r be the source and the sink in the digraph. Every process of the project has a possible minimal and an acceptable maximal working time (a_{ij} and b_{ij} respectively). Both have a necessary cost ($K(a_{ij})$ and $K(b_{ij})$). Between them, the cost changing is assumed to be linear, the cost intensity is shown by Eq. (1).

$$c_{ij} = \frac{K\left(a_{ij}\right) - K\left(b_{ij}\right)}{b_{ij} - a_{ij}} \tag{1}$$

The body of calendarisation is that every process has a given necessary working time $\left(\tau_{ij}; a_{ij} \le \tau_{ij} \le b_{ij}\right)$, departure time ($\mu_i$) and calendar vector ($d_{ij}$) as the work pattern of the resource. It is shown by Eq. (2).

$$d_{ij} = \begin{cases} 1, & if\ \_t\_is\_a\_workday \\ 0, & else \end{cases} \qquad t = 0,\ldots,T \tag{2}$$

The problem is defined in period T, which is the maximum acceptable project duration. The calendarised process time $\left(\vartheta_{ij}\left(\mu_i\right)\right)$ can be indirectly determined from Eq. (3).

$$\tau_{ij} = \operatorname{sgn}\left(\tau_{ij}\right) \sum_{t=\mu_i}^{\min\left\{\mu_i + \vartheta_{ij}\left(\mu_i\right)\right\}} d_{ij}\left(t\right) \tag{3}$$

Remark

- $\vartheta_{ij}\left(\mu_i\right)$ *is dependent on* τ_{ij} *and* μ_i*, so it is most likely not constant.*
- *If* $\tau_{ij} > 0$ *and* $\tau_{ij} > \sum_{t=\mu_i}^{T} d_{ij}\left(t\right)$ *then* $\vartheta_{ij}\left(\mu_i\right) = \infty$.
- *If* $\tau_{ij} < 0$ *and* $\tau_{ij} < -\sum_{t=\mu_i}^{0} d_{ij}\left(t\right)$ *then* $\vartheta_{ij}\left(\mu_i\right) = -\sum_{t=\mu_i}^{0} d_{ij}\left(t\right)$.

The time-cost trade-off problem gives a scheduling to the wanted deadline with minimal cost level. The basis of it is scheduling. This paper examines the possibilities in case of the scheduling presented above.

Cite this article as: Csordas, H. (2019) "Calendars in Time-Cost Trade-Off", Periodica Polytechnica Architecture, 50(1), pp. 63–66.
https://doi.org/10.3311/PPar.13257

2 Literature review

The first scheduling models were presented in the late 1950s by Bellmann (1958) and Dijkstra (1959). The problem in these works is very simplified; negative or changeable process durations and loops are not allowed. The solutions are based on linear programming. Scheduling is a longest path problem.

There are many generalisations of the problem. Franck et al. (2001) already showed a proper model for calendarisation. Negative duration and loops are solved even in some project management software.

The time-cost trade-off problem was presented at first in work of Kelley and Walker (1959). They gave a solution based on linear programming on AOE network. Fulkerson (1961) and Kelly (1961) gave another solution based on maximal flow algorithm. This problem can be originated to minimal cost flow algorithm, which is in Ahuja et al. (1993). Klafszky (1969), then Hajdu and Klafszky (1993), showed the acceleration of the problem. These solutions are also based on maximal flow algorithms.

There are many generalisations for this problem. Mályusz and Hajdu (2009) deal with using benefits or outcomes on nodes. Csordas and Malyusz (2006), and Csordás (2009; 2011) show different techniques to apply technological changes in the model. Changeable process times are included in the Cai et al. (2007) book, which also deals with cost optimisation in case of logistical problems. They worked out differentminimal cost problems according to the constraints determined on nodes. The actual value of transit times can be determined according to the departure time. This is the same as in the calendarised scheduling problem.

3 Research method

There are many proven optimal solutions in the literature. After studying them, they must be examined to see if they are capable of handling the conditions.

If a known algorithm is appropriate only with restrictions, then the possibility of applying the generalisations needs to be researched.

It is an established custom to adapt a solution worked out for other conditions. The mentioned logistical problem managed the time parameters in the same way. So, it is worth examining the solutions.

4 Results

4.1 Review of the literature

4.1.1 Calendarisation of scheduling

Because of arbitrary calendars and maximal constraints, there may form such loops (H) in scheduling, of which loop rate (ρ_H) has a changeable prefix. The loop rate is a feature of the loop. It is known in "constant" scheduling problems, if the loop rate is positive, the scheduling does not have a finite solution. But in case of calendarised process times, the loop rate is variable. If the current loop rate is positive, it can be counted round the loop again from the check value, which comes from the loop rate. This iteration can be continued while the check value is not larger than the start value. It is called critical loop. The known algorithm for the "constant" scheduling problem is able to handle this feature.

4.1.2 Time-cost trade-off problem

The algorithm of Hajdu and Klafszky can manage negative process times and loops. It has been proven that it gives the first optimal solution. So, it is obvious to try it in case of calendars (Hajdu and Klafszky, 1993).

The principle of the algorithm is simple. It based on a maximal flow – minimal cut problem. In consideration of the costs, it starts with zero flow and maximal process times ($0 \le \tau_{ij} = b_{ij}$ and $a_{ij} = \tau_{ij} \le 0$). In every cycle, after scheduling it finds a minimal cut, which determines the minimal cost rise and a shorter project duration. The measure of time reduction is given from the cut, which reduces all the potentials uniformly with this value after the cut. The arcs, which give the price of the cut, are on critical paths. The changes are unidirectional. The potentials decrease, the arcs become part of some critical paths.

In view of the principle, applying this algorithm shows some difficulties:

- If maximal process times are used in scheduling, the risk of overrunning T is the highest. In this case, there is no start scheduling.
- As the calendarised process times are inconstant, the time reduction cannot be uniform after the minimal cut.
- Process time reduction can be only on the arcs of critical paths. As the calendarised process times are not constant, it may change the critical paths. It is not guaranteed that arcs stay on critical paths on further steps.

4.1.3 Conversion of the logistical problem

In the Cai et al. (2007) book, the examined problem is a logistical task, where the aim is to find only one transit path with minimal cost. So, the scheduling is the shortest path problem. There are many examples in literature, where the solution of the shortest path problem is usable for the maximal path problem after multiplying the algorithm by (−1).

The book shows many solutions for different conditions. If time-cost trade-off problem waiting times are unlimited, so the proper solution for analysing is the TVSP-AWT-S (Time-Varying Shortest Path problem with Arbitrary Waiting times – Speed Up). According to the notation of the book, the basic element is $d_a(j,t)$, which is the cost of a $P(s,j)$ path. The algorithm examines $t=1,\ldots,T$ time intervals, increasing it one at a time. It gives the achievable nodes within t time and their costs respectively. The basic element is shown by Eq. (4). The notations of the book are a little different.

$$d_a(j,t)=min\left\{\begin{array}{c} d_a(j,t-1)+c(j,t-1) \\ \min\limits_{(i,j)\in A}\ \min\limits_{(u,\gamma)\in\text{“}(i,j,t)}\left\{d_a(i,u)+c(i,j,u)+c_\gamma(i,j,\gamma,u)\right\}\end{array}\right\} \tag{4}$$

It means that the cost of $P(s,j)$ path within t time is the minimum of the following cases

- the cost of the $P(s,j)$ path in $t-1$ time and the cost of waiting for a time unit
- the cost of a $P(s,i)$ path in u time, the cost of the transit on arc (i,j) according to u department time and the γ transit speed up according to u.

In the examined time-cost trade-off problem parameters are similar. However, there is no cost of waiting on nodes. The basic element must be modified as it examines all paths together as shown by Eq. (5).

$$\sigma_j(t)=\min\left\{\sigma_j(t-1);\sum_{ij\in A}\left[\sigma_j(u_i)+(b_{ij}-\tau_{ij})\cdot c_{ij}\right]\right\} \tag{5}$$

In the course of path finding, connecting $i_1,i_2,\ldots,i_k$ nodes must be considered; these can also be connected with each other. If a path variant is determined for $i_h\,(h=1,\ldots,k)$, which has connection to other mentioned $(i_{h1},i_{h2},\ldots,i_{hl})$ nodes, then u_{ih} gives an upper limit to the occurrence of $u_{ih1},u_{ih2},\ldots,u_{ihl}$. Following this effect, means further complications.

The TVSP-AWT-S algorithm can be adapted to the examined time- cost trade-off problem, when considering the new basic element. This is shown in Fig. 1.

The model, in practice, uses maximal constraints. It creates negative time parameters and loops in the network. The question is, whether this algorithm can manage these features.

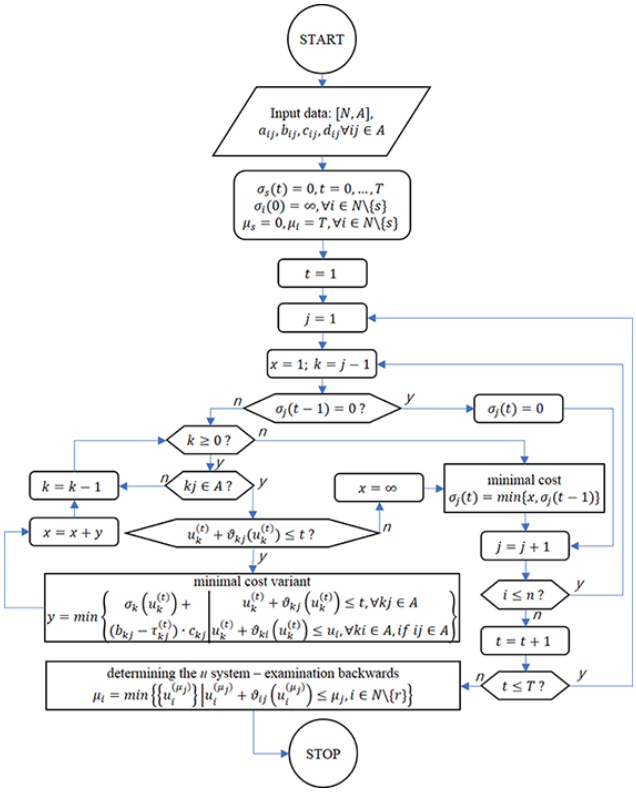

Fig. 1 Flowchart of the time-cost trade-off algorithm based on logistical problem

The algorithm analyses only the direct incoming arcs. So, in the case of using negative time parameters, it does not give an optimal solution. This is shown in an example in Fig. 2.

The network on the left shows input data. On arcs, there are a_{ij}, b_{ij}, c_{ij}, and $d_{ij}(t)=1$ constant. The network in the middle shows the basic elements in case of $t=0$. It is clear, that neither node 2, nor node 3 are available in this time. The network on the right shows the basic elements in case of $t=1$. According to the algorithm, node 3 is still not available. But this is not true. On path $P1\{1,3\}=\{(1,3)\}$ the necessary time is $\tau_{1,3}=1$. On path $P2\{1,3\}=\{(1,2);(2,3)\}$ the necessary time is $\tau_{1,2}+\tau_{2,3}=2+(-1)=1$. So $\sigma_3(1)=(b_{1,3}-\tau_{1,3})\cdot c_{1,3}+\left[(b_{1,2}-\tau_{1,2})\cdot c_{1,2}+(b_{2,3}-\tau_{2,3})\cdot c_{2,3}\right]$ $=(2-1)\cdot 10+\left[(3-2)\cdot 10+(0-1)\cdot 0\right]=20$. But the algorithm counts the cost from the previous iteration, so $\sigma_3(1)=\sigma_2 2(0)+(b_{2,3}-\tau_{2,3})\cdot c_{2,3}=\infty$. The algorithm should know the value $\sigma_2(2)=10$, which is not determined yet as it belongs to the next iteration where $t=2$. So negative process times are not useable in the algorithm.

The scheduling allows loops, which length is not positive. The previous conclusion excludes using negative

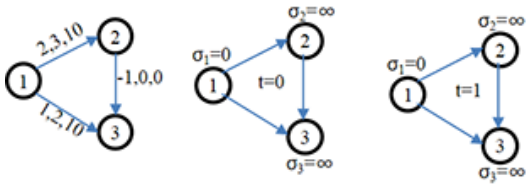

Fig. 2 Counterexample for negative time parameters

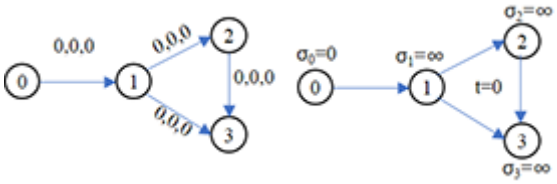

Fig. 3 Counterexample for loop

process times. However, a special case still exists. Namely, if all the process times are 0 in the loop. This is shown in Fig. 3.

The network on the left shows input data. The network on the right shows the basic elements in case of $t = 0$. Nodes in the loop relate to each other directly or indirectly. However, the algorithm only examines the direct connections. So, the basic elements in the loop can never decrease.

4.1.4 Advice for the calendarised time-cost trade-off algorithm

Considering the observation of the basic algorithm, it gives the idea of starting the problem at the other end. What if all process times are minimum $\left(a_{ij} = \tau_{ij}\right)$? This results areas follows.

- The scheduling gives the possible minimal project duration. The risk of overrunning T is the lowest.
- The risk of evolving a critical loop is the lowest. The process times of minimal constraints are the lowest, the process times of maximal constraints are the highest. So, the loop rates are the lowest. In other words, the loops are the loosest.
- It gives the highest cost level.
- There is a high likelihood that it is not an optimal solution, which permits extra costs only on critical paths. Here, every arc has extra cost where $c_{ij} > 0$, but it is not sure that all of them are on any critical paths.
- The effect of the slowdown paradox does notdevelop.

Remark

Slowdown paradox is a known feature in scheduling. It occurs when three processes are in slow - quick - slow sequence. Here, by slowdown the quick process, the total project duration becomes shorter.

5 Future tasks

If this system is the first solution of the algorithm, the task is to find the most effective way of decreasing the cost level. Here, the maximal flow-minimal cut problem is not optimal, so it cannot be used directly. On the other hand, it is useful to see what kind of time - cost system is optimal. It must find the scheduling where extra costs are only on critical paths.

References

Ahuja, R. K., Magnati, T. L., Orlin, J. B. (1993) "Network Flows: Theory, Algorithms and Applications", Prentice-Hall, Englewood Cliffs, NJ, USA.

Bellman, R. (1958) "On a Routing Problem", Quarterly of Applied Mathematics, 16(1), pp. 87–90.

Cai, X., Sha, D., Wong, C. K. (2007) "Time-Varying Network Optimization", Springer, New York, NY, USA.

Csordas, H., Malyusz, L. (2006) "A Network Flow Algorithm For Time-Cost Trade-off With Technological Decision", In: 7th International Conference Organization, Technology and Management in Construction, Zadar, Croatia, Sept. 20-22, 2006.

Csordás, H. (2009) "Optimal selection of recourses in projects based on the classical time - cost trade - offs", Periodica Polytechnica Social and Management Sciences, 17(1), pp. 47–55. https://doi.org/10.3311/pp.so.2009-1.05

Csordas, H. (2011) "Activities with multi-parameters in time-cost trade-off", Pollack Periodica, 6(2), pp. 37–48. https://doi.org/10.1556/Pollack.6.2011.2.4

Dijkstra, E. W. (1959) "A Note on Two Problems in Connexion With Graphs", Numerische Mathematik, 1(1), pp. 269–271. https://doi.org/10.1007/BF01386390

Franck, B., Neumann, K., Schwindt, C. (2001) "Project scheduling with calendars", OR-Spektrum, 23(3), pp. 325–334. https://doi.org/10.1007/PL00013355

Fulkerson, R. D. (1961) "A Network Flow Computation for Project Cost Curves", Management Science, 7(2), pp. 167–178. https://doi.org/10.1287/mnsc.7.2.167

Hajdu, M., Klafszky, E. (1993) "An algorithm to solve the cost optimization problem through an activity on arrow type network (CPM/cost problem)", Periodica Polytechnica Architecture, 37(1-4), pp. 27–40.

Kelley, J. E. (1961) "Critical-Path Planning and Scheduling: Mathematical Basis", Operations Research, 9(3), pp. 296–320. https://doi.org/10.1287/opre.9.3.296

Kelley, J. E. Walker, M. R. (1959) "Critical Path Planning and Scheduling", In: Proceedings of the Eastern Joint Computer Conference, Boston, USA, pp. 160–173. https://doi.org/10.1145/1460299.1460318

Klafszky E. (1969) "Hálózati folyamok" (Network Flows), Bolyai János Matematikai Társulat kiadványa, Budapest, Hungary. (in Hungarian)

Mályusz, L., Hajdu, M. (2009) "How would you like it: cheaper or shorter?", Organization, Technology & Management in Construction: An International Journal, 1(2), pp. 59–63.