

Notes on Cycles in Activity-on-Arc Networks[#]

Anna Černá – Jan Černý*

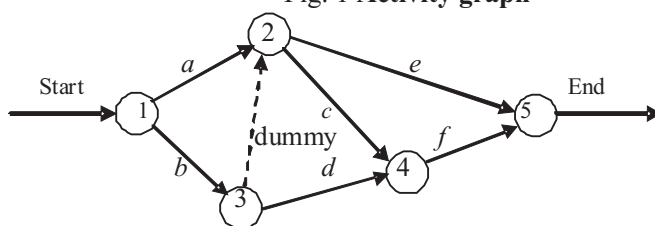
Introduction

This paper deals with sets of **activities** which create **products** (i.e., goods or services). Each activity is supposed to be elementary and indivisible into sub-activities. From our point of view, a product is given when we know the necessary set of activities which create it. Therefore, we can speak about a product $A = \{a, b, c, \dots\}$ where a, b, c, \dots are the activities.

As usual, each activity has its **length**, i.e., a number of time units. We shall consider the deterministic case, not the random one.

A **precedence relation** “ \rightarrow ” is supposed to be defined on the set A . We can say that the activity a precedes the activity b and we write $a \rightarrow b$ if b can start only after a is finished. The set A , with the relation “ \rightarrow ”, can be represented by an **activity network** (activity graph) of the type shown in Fig. 1.

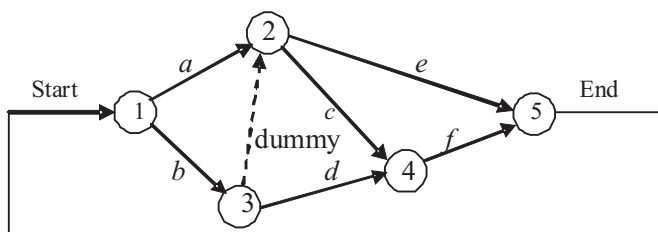
Fig. 1 Activity graph



Cycles of Activities

We intend to study situations where the same product A is periodically repeated. The simplest case, widely treated in literature (see [3], [6], [7]), supposes that the repetition starts only when the preceding products have been finished, as in Fig. 2.

Fig. 2 Activity graph for repeating production, “start-after-end” type



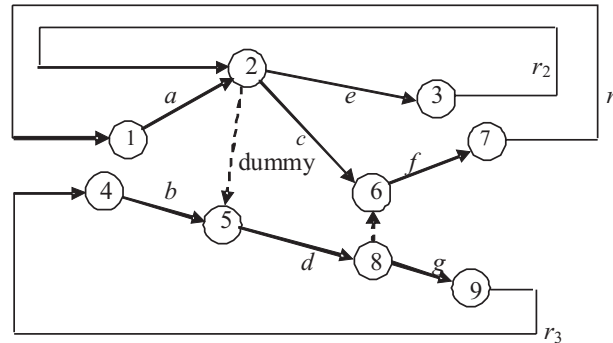
[#] The paper is one of the results of the Czech Grant Agency project No. 402/06/0084 *Modelling and Optimization of Decision Processes within Municipal and Regional Administration*

* Ing. Anna Černá, CSc., – senior lecturer; Prague University of Economics, Faculty of Management.
Prof. RNDr. Jan Černý, DrSc., Dr.h.c.; Prague University of Economics, Faculty of Management.

Several existing papers deal with directed cyclic graphs. However, the problems they deal with are different from ours. Papers [4], [8] and in some sense [2] as well, are oriented on finding cycles, whereas [1] deals with the reduction of cyclic graphs and [5] is limited to “start-after-end” graphs and their transformation to acyclic ones.

In many practical cases the “start-after-end” situation does not occur, e.g., a building company plans to build several identical single-family houses. In this case, the activity network for one house looks like the one in Fig. 1. Nevertheless, the activity cycle applied to the building of all the houses is not in the form “start-after-end”, since e.g. the foundation of the house $n + 1$ can be excavated and built much sooner than the roof of the house n is finished. Therefore, the corresponding cyclic graph is not in the form shown in Fig. 2, but the one in Fig. 3.

Fig. 3 Activity graph for repeating production, “not-start-after-end” type



We can see that the graph contains activity arrows, dummy arrows and a new type of arrows called backward arrows. They represent the redeployment activities of working groups from the house n to the house $n + 1$. In Fig. 3 there are three such arrows corresponding to three groups:

- G1 accomplishing activities a, c, f ,
- G2 accomplishing e ,
- G3 accomplishing b, d, g .

Removal arrows introduce a new relation “ $\rightarrow r \rightarrow$ ” between couples of activities in the set A . In Fig. 3, we have e.g. $f \rightarrow r_1 \rightarrow a$.

Calculation of Cycle Length

We shall suppose that the activities a, b, \dots have deterministic lengths t_a, t_b, \dots in each cycle. In the first cycle, if they have starting times s_a, s_b, \dots , their ending times will be exactly $s_a + t_a, s_b + t_b, \dots$. Moreover, the dummy activities have a length of 0.

In this cyclic case, instead of the total length of construction, we shall be interested in the (minimum) cycle length τ , so that for each n the activities a, b, \dots will have the starting times $s_a + n\tau, s_b + n\tau, \dots$ and the ending times $s_a + t_a + n\tau, s_b + t_b + n\tau, \dots$ in the $(n + 1)$ -th cycle. Therefore, for the given values t_a, t_b, \dots and the variables s_a, s_b, \dots, τ the following inequalities must hold:

$$i \in A \Rightarrow s_i \geq 0 \quad (1)$$

$$i, j \in A, i \rightarrow j \Rightarrow s_i + t_i \leq s_j \quad (2)$$

$$i, j \in A, i \rightarrow r \rightarrow j \Rightarrow s_i + t_i - \tau \leq s_j \quad (3)$$

Usually, the time lengths t_a, t_b, \dots are positive integers.

Our goal is to find the starting times s_a, s_b, \dots and the minimum value of the cycle length τ . We can use linear programming while solving the following problem:

Minimize τ subject to (1), (2), (3).

Total Time Minimization, Critical Activities and Slack Times

The constraints (1) – (3) enable us to find the minimum cycle length. Nevertheless, that is not the only thing we need. The number of repetitions is certainly limited by a natural number m , e.g., we intend to construct m houses. Thus, we want to minimize the ending time of the last activity in the last cycle. It is obvious that this can be reached in two steps:

1. Finding the minimum cycle length τ .
2. Finding the minimum possible values of all starting times s_a, s_b, \dots

However, it is possible to combine these steps into one by modifying the objective function. Suppose that q is a “big number”, e.g. $q = m(s_a + s_b + \dots)$. Let us define the following **Problem P1**: to minimize the objective function.

$$z = \sum_{i \in A} s_i + q \cdot \tau \quad (4)$$

subject to (1), (2), (3).

It is obvious that the solution to Problem P1 represents the **minimum possible value** τ of the cycle length and the **minimum possible value** of each s_a, s_b, \dots . Putting

$$t_{\text{end}} = (m - 1)\tau + \max \{s_a + t_a, s_b + t_b, \dots\}, \quad (5)$$

we obtain the **minimum possible time for ending the whole set of cycles**.

Another managerial interest is to find the **critical activities**, i.e., the ones in which delays are undesirable. If a critical activity has a delay d and other critical activities keep the duration times t_i , then the whole project will have the same delay d .

In order to find them, we shall find the **last possible starting times in the first cycle** p_a, p_b, \dots of all activities in A .

Let us solve Problem P1. It is obvious that the length of at least one cycle in the activity graph equals τ . Let us choose an arbitrary activity k belonging to that cycle.

Then we can formulate **Problem P2**: to maximize the objective function.

$$z = \sum_{i \in A} p_i \quad (6)$$

subject to

$$i \in A \Rightarrow p_i \geq s_i \quad (7)$$

$$i, j \in A, i \rightarrow j \Rightarrow p_i + t_i \leq p_j \quad (8)$$

$$i, j \in A, i \rightarrow r \rightarrow j \Rightarrow p_i + t_i - \tau \leq p_j \quad (9)$$

$$p_k = s_k \quad (10)$$

Suppose that Problem P2 is already solved. Then each activity $i \in A$ with the property $p_i = s_i$ is called a **critical activity**. If the activity $j \in A$ is not critical then the value $p_j - s_j$ is called a **slack** in the activity j .

The sense of the last notion is that if each other activity $i \in A$ starts at the time $s_i + n\tau$ in the $(n + 1)$ -th cycle, then in the same cycle the activity j can have the delay $p_j - s_j$.

Note: Similarly to the non-cyclic case, it is possible to examine mutual delays of several activities in the same cycle.

Example: Let us suppose that the activities a, b, \dots, g sketched in Fig. 3 have the following durations: $t_a = 3, t_b = 1, t_c = 2, t_e = 7, t_f = 2, t_g = 3$. With respect to Fig. 3, we have the following constraints of the type (2) and (3):

$$s_a + 3 \leq s_c \quad (2.1)$$

$$s_a + 3 \leq s_d \quad (\text{because of the dummy arrow}) \quad (2.2)$$

$$s_a + 3 \leq s_e \quad (2.3)$$

$$s_b + 1 \leq s_d \quad (2.4)$$

$$s_c + 2 \leq s_f \quad (2.5)$$

$$s_d + 4 \leq s_f \quad (\text{dummy}) \quad (2.6)$$

$$s_d + 4 \leq s_g \quad (2.7)$$

$$s_e + 7 - \tau \leq s_e \quad (3.1)$$

$$s_g + 3 - \tau \leq s_b \quad (3.2)$$

$$s_f + 2 - \tau \leq s_a \quad (3.3)$$

Through calculation we can obtain the minimal possible value $\tau = 9$ and the first possible beginning times: $s_a = 0, s_b = 1, s_c = 3, s_d = 3, s_e = 3, s_f = 7, s_g = 7$. We can observe that if only one such house is constructed, the duration of the construction will be 10, i.e., more than the cycle length.

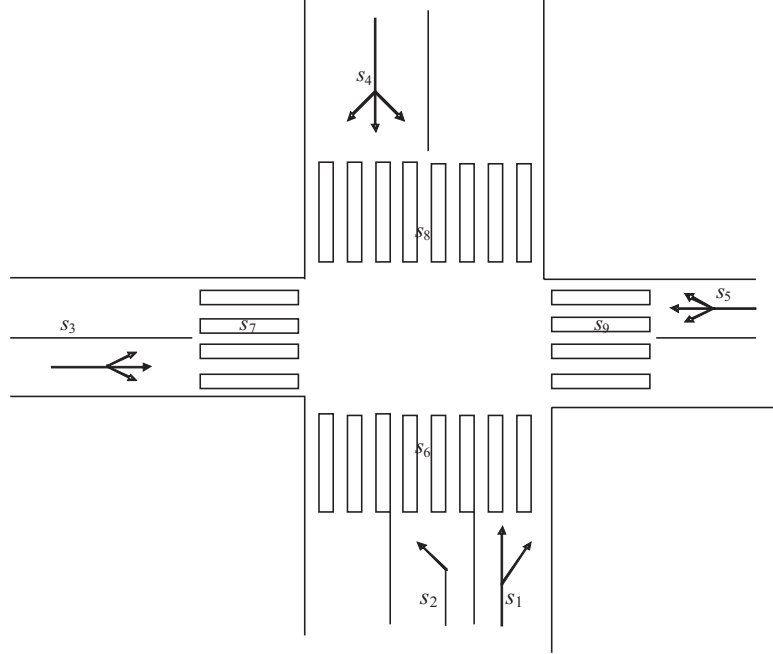
Furthermore, we obtain the last possible beginning times: $p_a = 0, p_b = 2, p_c = 5, p_d = 3, p_e = 5, p_f = 7, p_g = 8$. We see that a, d and f are critical, and the slacks in b, c, e and g are subsequently 1, 2, 2, 1.

Applications

Cyclic activity networks can be applied outside the building industry as well. For instance, the use of this optimization can be anticipated in the manufacturing of large machines (airplanes, cranes, ships, etc.) of equal types not using an assembly line.

An interesting application can be found in road junction signal settings. Let us examine an example (Fig. 4).

Fig. 4 Example of a road junction



We can see traffic streams $s_1 - s_5$ (vehicles) and $s_6 - s_9$ (pedestrians). Each of them is controlled by its own traffic signal. Vehicle signals are three-color (green, yellow, red), but pedestrian ones are two-color (green, red). The signal for s_1 is a combined "straight and right" double arrow, s_2 has a single "left" arrow, and $s_3 - s_9$ are full circles. The arrows guarantee passing through the junction without any necessity to give right of way, whereas the circles allow entrance to the junction only. Therefore, some pairs of them are not permitted to have green at the same time; such pairs are said to be "conflicting". The conflicting pairs represent a subset C of the set $S \times S$ where $S = \{s_1, \dots, s_9\}$.

If s_i and s_j are conflicting, the green for s_j can begin no sooner than t_{ij} seconds after the end of the green of s_i . The value t_{ij} is called the "clearing time", i.e. the time devoted to the "clearing activity" c_{ij} , when the junction is cleared of the traffic participants s_i in favor of the s_j .

In Table 1 we can see the values t_{ij} , e.g., $t_{27} = 5$. The empty cell represents non-conflicting pairs from the set $(S \times S) - C$, e.g., s_4 and s_7 can have the green light at the same time.

Tab. 1 Clearing times for the junction in Fig. 4

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9
s_1			1		4	2		4	4
s_2			2	3	2	1	5		
s_3	5	4		2			2		6
s_4		2	3		1	4		2	
s_5	1	2		5			6		2
s_6	12	10		10					
s_7		6	8		6				
s_8	10			12					
s_9	7		6		8				

Let us suppose that traffic engineers break down the set $S = \{s_1, \dots, s_9\}$ into three disjoint subsets called phases: $F_1 = \{s_1, s_2\}$, $F_2 = \{s_4, s_7, s_9\}$ and $F_3 = \{s_3, s_5, s_6, s_8\}$. We see that for each F_k and each $s_i, s_j \in F_k$ the pair s_i, s_j is non-conflicting, i.e., the whole phase F_k can have the green signal at the same time.

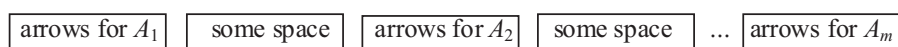
Moreover, suppose that the desired value of the cycle length τ is at most 60 seconds. From that, the engineers will identify the following minimum length of the green light t_i for the signal for s_i , $i = 1, \dots, 9$ (in seconds):

$$t_1 = 25, t_2 = 12, t_3 = 12, t_4 = 25, t_5 = 8, t_6 = t_7 = t_8 = t_9 = 5$$

If we intend to express the situation by a cyclic activity graph, we have to summarize it. The set of activities A consists of:

- “green-light activities” a_i with the length t_i for each $s_i \in S$ representing the process when the stream s_i enters and passes the junction. These activities belong to the disjoint “phase subsets” A_1, \dots, A_m (in our example $m = 3$), corresponding to phases F_1, \dots, F_m ,
- each “clearing activity” $c_{ij} \in C$ with the length t_{ij} .

Of course, if we draw the activity graph, we have to position it from the left:



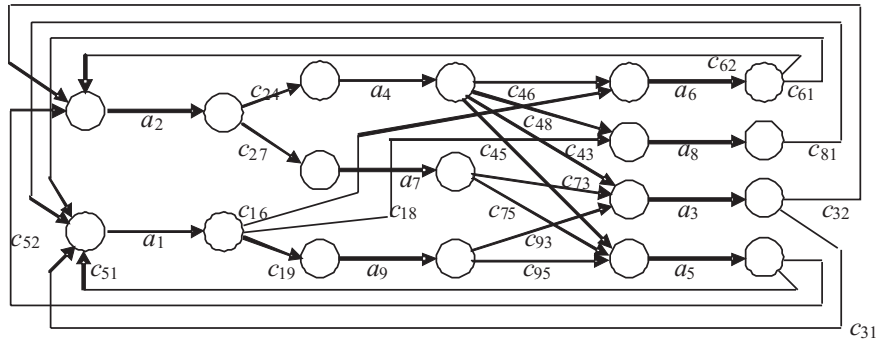
(the spaces are reserved for clearing activity arrows).

Then we have to draw “forward” arrows for such $c_{ij} \in C$ where a_i is located left of a_j and “backward” arrows for such $c_{ij} \in C$ where a_i is located right of a_j .

It is possible to omit “clearing activities” c_{ij} when a_i is located left of a_j and there exists a_k such that it is located between a_i and a_j and both $c_{ik} \in C$ and $c_{kj} \in C$.

The resulting cyclic activity graph (without any labeling of vertices) is drawn in Fig. 5.

Fig. 5 Cyclic activity graph for the road junction example



Hereafter, the linear programming optimization can be used like above:

1. To minimize the objective function (4) subject to (1) – (3). If the resulting cycle length τ exceeds the desired limit (in our example 60 sec) the traffic engineers have to decide what to do next. If it does not, we proceed to 2.
2. To minimize the objective function (6) subject to (7) – (10).
3. To shift the ends of the green lights to later values, not violating the values p_i .

Note: It may happen that a requirement for coordination in a “green wave” strictly determines the cycle length λ (say 60 sec). If the resulting τ is smaller than λ , then step 2 above can be modified taking the limit λ instead of the resulting τ from step 1.

Conclusion

We have defined and studied “not-start-after-end” cyclic activity graphs. Compared to acyclic ones (and the ones of the “start-after-end” type), our case leads to a more complicated model and new features such as the possible difference between the cycle length and the duration of one item. The authors were able to propose a linear programming model but not a comprehensive graph theory; maybe it will be found in future.

As we have shown, the cyclic activity graph can be used as a good optimization tool in managerial decision-making. The authors hope that the application is much more widespread than we have mentioned in the building industry, manufacturing of airplanes, ships and cranes, and in traffic signal settings.

References

- [1] BEKKE, J. H.; BAKKER, J. A. 2003. *Fast Recursive Data Processing in Graphs Using Reduction*. Proc. of the 21st IASTED International Conference Applied Informatics, 2003, February 10–13, Innsbruck, Austria.
- [2] BIANCHINI, M.; GORI, M.; SARTI, L.; SCARSELLI, F. 2006. *Recursive Neural Networks and Graphs: Dealing with Cycles*. Lecture notes in Computer Science, Springer Berlin / Heidelberg, 2006, 38–43.
- [3] HEIZER, J.; RENDER, B. 1999. *Operations Management*. 5th ed., Prentice Hall, Inc., Upper Saddle River NJ, 1999.
- [4] KARAYIANNIS, A.; LOIZOU, G. 1978. *Cycle Detection in Critical Path Networks*. Information processing letters, 1978, Vol. 7, No. 1, 15–19.

- [5] SANDNES, F. E.; SINNEN, O. 2005. *A New Strategy for Multiprocessor Scheduling of Cyclic Task Graphs*. Int. J. High Performance Computing and Networking, 2005, Vol.
- [6] SCHONBERGER, R. J.; KNOD, E. M. 1998. *Operations Management*. 5th ed., Business Publications, Inc., Plano Tex, 1998.
- [7] STEVENSON, W. J. 1989. *Introduction to Management Science*. Boston MA : Irwin, 1989.
- [8] WEINBLATT, H. 1972. *A New Algorithm for Finding the Simple Cycles of a Finite Directed Graph*. Journal of the ACM, 1972, Vol. 19, 45–56, No. 1., 62–71.

Notes on Cycles in Activity-on-Arc Networks

Abstract

A new type of cyclic activity graph is introduced. In contrast to the well-known “start-after-end” type with only one backward arrow, the new one admits several backward arrows representing the removal of working groups from the n -th to the $(n + 1)$ -th cycle (e.g., from the previous family house being constructed to the next one). It is shown how to calculate minimum cycle length, first and last possible starting times of activities and slacks. Brief examples of applications (e.g., in building industry, manufacturing of airplanes, ships and cranes) are mentioned. Moreover, an example of traffic signal settings is presented in detail.

Keywords: Network; Activity; Cycle; Optimization; Application; Traffic Lights.

JEL classification: C6.