Journal Pre-proof

Project Scheduling with Generalized Precedence Relations: a New Method to Analyze Criticalities and Flexibilities

Lucio Bianco, Massimiliano Caramia, Stefano Giordani

PII: S0377-2217(21)00617-2
DOI: https://doi.org/10.1016/j.ejor.2021.07.022
Reference: EOR 17351

To appear in: *European Journal of Operational Research*

Received date: 14 December 2020
Accepted date: 12 July 2021

Please cite this article as: Lucio Bianco, Massimiliano Caramia, Stefano Giordani, Project Scheduling with Generalized Precedence Relations: a New Method to Analyze Criticalities and Flexibilities, *European Journal of Operational Research* (2021), doi: https://doi.org/10.1016/j.ejor.2021.07.022

Highlights

- Project scheduling with GPRs and unlimited resources
- GPRs with both minimum and maximum time lags
- Evidence of failures and limits of the current activity criticality theory
- New definitions of the activity criticalities
- New method to overcome failures and limits of the current theory

Project Scheduling with Generalized Precedence Relations: a New Method to Analyze Criticalities and Flexibilities

Lucio Bianco * Massimiliano Caramia † Stefano Giordani ‡

Abstract

In this paper, we illustrate a new method to overcome the failures of the theory proposed by Elmaghraby and Kamburowski (1992) and De Reyck (1998) for the analysis of activity criticalities and flexibilities in non-preemptive project scheduling with generalized precedence relations under unlimited resources. These failures, discussed in detail in this paper, call for a new approach to study this problem. We provide new definitions of criticalities and consequently new tools for their identification within a more general framework without ambiguities.

Keywords: Project Scheduling; Critical Path; Flexibility Analysis; Generalized Precedence Relations.

1 Introduction

This paper concerns the method of determining the activity criticalities and flexibilities in non-preemptive project scheduling with Generalized Precedence Relations (GPRs), in the presence of minimum and maximum times lags, under unlimited resources.

It is well known that a critical path on a project network is the longest path from the source (dummy) node to the sink (dummy) node. If the project network comprises only finish-to-start precedence relations, i.e., the Standard Precedence Relations (SPRs) assumed in the traditional Critical Path Method (CPM), a critical path is always *elementary*, i.e., it visits a node at most once, since the network cannot contain cycles. When a project network comes with GPRs with minimum and maximum times lags, the network may contain cycles and then a critical path may not be elementary, that is, traversing a node more than once and, hence, containing a cycle (see, e.g., De Reyck, 1998, Demeulemeester and Herroelen, 2002, p. 123).

*Dipartimento di Ingegneria dell'Impresa, University of Rome "Tor Vergata", Via del Politecnico, 1 - 00133 Rome, Italy email: bianco@dii.uniroma2.it

†Corresponding author, Dipartimento di Ingegneria dell'Impresa, University of Rome "Tor Vergata", Via del Politecnico, 1 - 00133 Rome, Italy email: caramia@dii.uniroma2.it

‡Dipartimento di Ingegneria dell'Impresa, University of Rome "Tor Vergata", Via del Politecnico, 1 - 00133 Rome, Italy email: stefano.giordani@uniroma2.it

The first systematic formalism of GPRs, based on a previous work of Roy (1962), was introduced by Kerbosch and Shell (1975). Successively, GPRs have been discussed by Elmaghraby (1977) and in the book by Moder *et al.* (1983), where the results of Crandall (1973) and Wiest (1981) are also summarized. In the more general case with GPRs, differently from project networks with SPRs where an increase of the duration of a critical activity always results in an increase of the project duration of the same amount, it may happen that the project duration increases when the duration of a critical activity is shortened and viceversa. This phenomenon, apparently anomalous, was firstly studied by Wiest (1981), who proposed a first classification of critical activities in four types of criticalities (*normal*, *neutral*, *reverse*, *perverse*). Analog classification appears in Moder *et al.* (1983), Moder and Crandall (1985), and Hajdu (1997), where the term *bicritical* is adopted in place of perverse.

In another seminal paper, Elmaghraby and Kamburowski (1992) further studied the anomaly occurring under GPRs also in the presence of maximun time lags, and described two phenomena: one refers to the reduction (increase) of project duration as a consequence of prolonging (shortening) an activity, and the other one occurs when diminishing (increasing) the duration of an activity results in a time-infeasibility for the project, since the length of a cycle becomes positive. To formalize these scenarios they introduced five distinct criticality types (*start-critical*, *finish-critical*, *backward-critical*, *forward-critical*, and *bi-critical*), different from those introduced by Wiest (1981), and the new concept of *flexibility*. All these concepts were introduced on an Event On Nodes (EON) project network representation and were defined looking at the network topology and considering networks admitting only elementary critical paths. Later, De Reyck (1998) in his Ph.D. thesis revisited these concepts, adapting them for Activity On Nodes (AON) project networks. De Reyck associated these definitions to durations, start times, and finish times of the critical activities, and extended their validity also to non-elementary critical paths, i.e., to critical paths containing also cycles. The same author proposed a method for recognizing the criticalities and flexibilities of a critical activity, based on the types of its ingoing and outgoing precedence relations. Since De Reyck's Ph.D. thesis appears not to be available in the open literature, the reader may find the above mentioned analysis in the book by Demeulemeester and Herroelen (2002).

After 1998, many authors have studied project scheduling with GPRs under different scenarios (e.g., type of time lags, maximum activity durations, imprecision or lack of information on future events, and so on). Here, we only recall the papers that mainly focus on the problem of identifying the types of criticalities and flexibilities of critical activities. Valls and Lino (2001) present a systematic study of the critical nature of the activities. They show that the classification of Wiest (1981) is not complete and propose an extension from four to six types of criticalities (adding *increasing-normal* and *decreasing-reverse* types), along with a related characterization valid only for GPRs project network with minimum time lags. Hajdu (2015) and Hajdu *et al.* (2016) propose to study the criticalities extending the modeling capabilities of the Precedence Diagramming Method (PDM) with minimum time lags by introducing different types of precedence relations (point-to-point relations, continuous relations, and non-linear activity (production-time) functions). Nisar and Halim (2018) start from the criticality definitions given by Wiest (1981) and extend his classification by introducing further types of criticalities. Surprisingly, they ignore the previous work by Valls and Lino (2001), who have firstly proposed the same classification. More recently, Kong and Li (2020) propose a network model where a combination of two types of GPRs, called hybrid precedence relations and maximum activity duration, are considered.

However, to the best of our knowledge, the analysis made by De Reyck (1998) is the only one that considers the most general case of GPRs with minimum and maximum time lags. It is widely accepted and not revised from 1998 to nowadays.

Since the method proposed by De Reyck (1998), as well as that one by Elmaghraby and Kamburowski (1992), fails in the general context of GPRs, as shown further on, in this work we give new definitions of activity criticalities and develop a new general methodology able to determine the correct activity criticalities and flexibilities in project scheduling with GPRs with minimum and maximum time lags under unlimited resources.

The remainder of the paper is organized as follows. Section 2 provides some definitions and notations. In Section 3, by means of some examples, we show potential failures of the method proposed by De Reyck (1998). In Section 4, after having redefined and discussed the different types of criticalities, we provide our new method. In Section 5, we show that its application on the same examples of Section 3 overcomes the failures of the methods so far utilized. Finally, we close the paper with some conclusions in Section 6. A *supplemental document* containing four appendices is also available, where the proofs of some results and additional examples show the correctness of the proposed method. Moreover, potential failures of the method proposed by Elmaghraby and Kamburowski (1992) are also discussed. In addition, it is shown that our criticality characterization also includes and extends that of Valls and Lino (2001) to the more general case of GPRs with both minimum and maximum time lags.

2 Definitions and notations

Accordingly to De Reyck (1998) and to Demeulemeester and Herroelen (2002), hereafter we assume that a project is modeled by means of an AON network $N = (V, A; d, \delta)$. Node set V, with $V = V^r \cup \{1, n\}$, represents the set $V^r = \{2, \ldots, n-1\}$ of $n-2$ real activities, that are to be performed without preemption, and two additional dummy activities 1 and n, with duration equal to zero, representing project beginning and completion, respectively. Vector d represents the set of activity durations d_i, with $i \in V^r$, and δ is the generic time lag of the precedence relation between a pair of activities. Without loss of generality, we assume that the real activity durations are positive integers and the time lags are integers. Arc set A represents GPRs between pairs of activities. An arc may model a start-to-start (SS), a start-to-finish (SF), a finish-to-start (FS) and a finish-to-finish (FF) precedence relation with minimum or maximum time lags. A GPR with minimum time lag ($SS_{ij}^{min}(\delta)$, $SF_{ij}^{min}(\delta)$, $FS_{ij}^{min}(\delta)$, $FF_{ij}^{min}(\delta)$) specifies that activity j can start (finish) only if its predecessor i has started (finished) at least δ time units before. Analogously, a GPR with maximum time lag ($SS_{ij}^{max}(\delta)$, $SF_{ij}^{max}(\delta)$, $FS_{ij}^{max}(\delta)$, $FF_{ij}^{max}(\delta)$) imposes that activity j can be started (finished) at most δ time units beyond the start (finish) time of activity i. It is well known (see, e.g., Demeulemeester and Herroelen, 2002) that a GPR with maximum time lag is equivalent to a GPR with minimum time lag with opposite direction and opposite time lag. Hence, we can always model the project activities and their relationships with a GPRs AON network with only minimum time lags, but the resulting network may contain cycles. Dummy nodes 1 and n, representing dummy activities 1 and n, respectively, are added in order to have a project network with exactly one source (i.e., node 1) and one sink (i.e., node n). Therefore, arc set A includes all arcs $SS_{1i}^{min}(0)$ and $FS_{in}^{min}(0)$ necessary to obtain this result.

It is also well known that with the transformations of Bartush *et al.* (1988) we can represent the project network in a so called *standardized* form where only one type of GPRs is considered. In particular, we consider $SS_{ij}^{min}(\ell)$, with time lags $\ell = \delta$, $\ell = \delta - d_j$, $\ell = \delta + d_i$, $\ell = \delta + d_i - d_j$, in place of the GPRs of type $SS_{ij}^{min}(\delta)$, $SF_{ij}^{min}(\delta)$, $FS_{ij}^{min}(\delta)$, $FF_{ij}^{min}(\delta)$, respectively. This standardized network represents the time constraints among the activity starting events due to the GPRs. Therefore, assuming the project starting at time zero, i.e., the initial dummy activity earliest start time $ES_1 = 0$, it is well known that, in the case of unlimited resources, the earliest start time ES_i of activity i is equal to the length of the longest path from source node 1 to node i in the standardized network, and the (minimum) project duration T is equal to the length of the longest path from source node 1 to sink node n, i.e., $T = ES_n$. It is also well known that the latest start time LS_i of activity i is equal to the difference between T and the length of the longest path from node i to sink node n. Finally, the earliest finish time EF_i of activity i is equal to $ES_i + d_i$, and the latest finish time LF_i of activity i is equal to $LS_i + d_i$. Activities having equal earliest and latest start (finish) times are *critical*, and longest paths from source node 1 to sink node n traverse only critical activities and are called *critical* paths. Since the network may contain cycles, critical paths may be non-elementary, that is, they may contain *critical* cycles whose lengths are clearly equal to zero.

To assure that dummy activities 1 and n actually represent the project start and end, respectively, it is assumed that in the standardized network there is at least a path from node 1 to node i of non-negative length, and there is at least a path from node i to node n of length at least equal to d_i, for each real activity i. If these paths do not exist, De Reyck (1998) suggests to add to the standardized network arc $(1, i)$ of length $\ell_{1i} = 0$, i.e., precedence relation $SS_{1i}^{min}(0)$ between activities 1 and i on the original GPRs network, and to add arc (i, n) of length $\ell_{in} = d_i$, i.e., precedence relation $FS_{in}^{min}(0)$ between activities i

and n on the original GPRs network, respectively.

3 Potential failures of De Reyck's method

In this section, we analyze through two project network examples some potential failures of the method proposed by De Reyck (1998) in determining activity criticalities. Some analog failures can also be shown on the approach proposed by Elmaghraby and Kamburowski (1992) and are discussed in Appendix B of the paper *supplemental document*, where also additional examples can be found.

For ease of presentation, we firstly recall the critical definitions adopted by De Reyck (1998) (as reported at page 124 of the book of Demeulemeester and Herroelen, 2002). An activity is classified:

- *critical*, if it lays on a critical path of the standardized network;
- *start-critical* (*finish-critical*), if it is critical and the project duration increases when the activity start (finish) time is delayed;
- *forward-critical* (*backward-critical*), if it is start-critical (finish-critical) and the project duration increases when the activity duration is increased (decreased);
- *bi-critical*, if it is start- and finish-critical and the project duration increases when the activity duration is either increased or decreased.

In particular, De Reyck identifies the type of criticalities of a critical activity i on the basis of the types of critical precedence relations preceding and succeeding the activity, according to the following scheme (see Table 6 at page 124 of the book of Demeulemeester and Herroelen, 2002), where X is indifferently S or F:

- *start-critical* (*finish-critical*), if there is a critical precedence relation of type XS_{hi}^{min} (XF_{hi}^{min}) preceding the activity and a critical precedence relation of type SX_{ij}^{min} (FX_{ij}^{min}) succeeding the activity;
- *forward-critical* (*backward-critical*), if there is a critical precedence relation of type XS_{hi}^{min} (XF_{hi}^{min}) preceding the activity and a critical precedence relation of type FX_{ij}^{min} (SX_{ij}^{min}) succeeding the activity;
- *bi-critical*, if there are critical precedence relations of type XS_{hi}^{min} and $XF_{h'i}^{min}$ preceding the activity, and critical precedence relations of type FX_{ij}^{min} and $SX_{ij'}^{min}$ succeeding the activity.

As for activity flexibilities, De Reyck (1998) adopts the definitions proposed by Elmaghraby and Kamburowski (1992), that is, an activity is said *forward-inflexible* (*backward-inflexible*) if an increase (decrease) of its duration by 1 increases the project duration by the same amount or makes the project time-infeasible (with respect to precedence relations). It is useful to recall that project time-infeasibility follows from the existence of positive length cycles on the standardized network, i.e., directed cycles whose total arc length is positive. An activity is said *bi-inflexible* if it is both forward- and backward-inflexible; finally, it is said *bi-flexible* if it is not *forward-inflexible* and not *backward-inflexible*. Consequently, a forward-critical (backward-critical) activity is forward-inflexible (backward-inflexible), a bi-critical activity is bi-inflexible, and a start-critical (finish-critical) activity is bi-flexible.

3.1 The case of more than one criticality associated with an activity

Example 1. Let us consider the GPRs project network, with minimum time lags, shown in Figure 1, where arc attributes represent GPRs and node weights are activity durations.

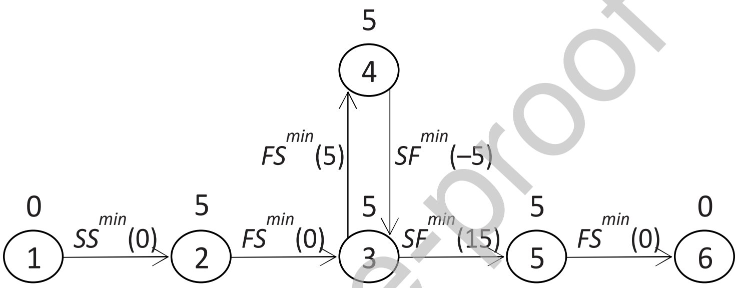

Figure 1: The project network with GPRs of Example 1

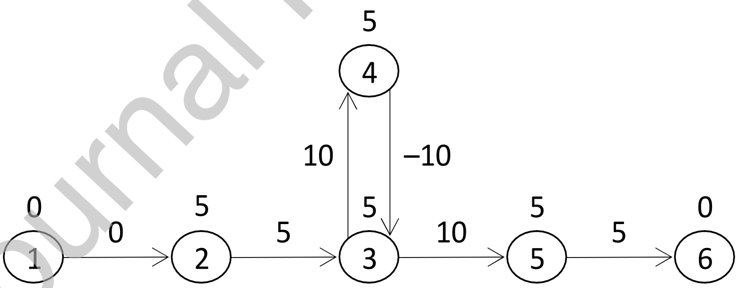

Figure 2: The standardized network associated with the project network in Figure 1

The standardized network is shown in Figure 2, where arc weights are time lags ℓ and node weights are activity durations. On this network, let us consider the critical path (1 — 2 — 3 — 4 — 3 — 5 — 6) of length 20: this path is non-elementary and contains the critical cycle (3 — 4 — 3) whose length is obviously equal to zero. All the activities belong to the critical path, and, hence, are critical.

Let us examine first the criticality type of activity 3 following the scheme proposed by De Reyck (1998). It is easy to verify that according to the critical relations involving activity 3, this activity should be bi-critical and, therefore, bi-inflexible. Unfortunately, the flexibility analysis does not confirm this result. In fact, if the activity duration was decreased by 1, i.e., it was $d_3 = 4$, the arcs lengths of cycle $(3 — 4 — 3)$ would change, but the cycle length would remain equal to zero and the critical path length would not change. Therefore, activity 3 is backward-flexible. If, on the contrary, the duration of activity 3 was increased by 1, i.e., if it was $d_3 = 6$, the lengths of the arcs of that cycle would change, although the cycle length would remain equal to zero. However, the longest path from node 1 to node 4 would increase by 1, while the longest path from node 4 to node 6 would decrease by the same amount and would have length $4 < d_4 = 5$. Consequently, the finish time of activity 4 should be 21, while the length of the critical path, i.e., the project completion time, would remain unchanged and equal to 20. Since this cannot happen, in order to overcome this anomaly, we have to correct the standardized network by adding arc $(4, 6)$ of length $\ell_{46} = d_4$, as shown in Figure 3. In doing so, the length of the longest path from node 4 to node 6 continues to be (at least) equal to d_4. Note that this correction would imply the correction of the original GPRs project network too, with the addition of precedence relation $FS_{46}^{min}(0)$ from activity 4 to dummy final activity 6. The critical path on the standardized network would be $(1 — 2 — 3 — 4 — 6)$ of length

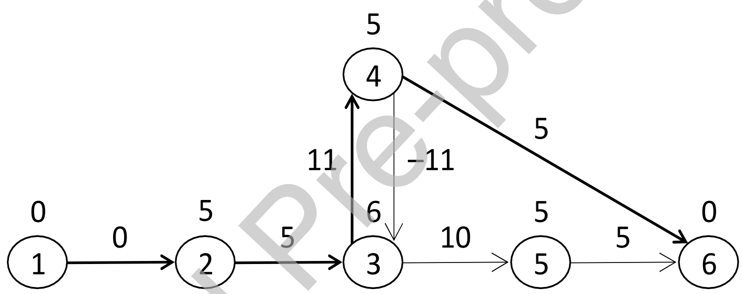

Figure 3: The standardized network of the project network in Figure 1, with $d_3 = 6$, after its correction with the addition of arc $(4, 6)$ of length $\ell_{46} = d_4$ (in bold the critical path)

21. Therefore, activity 3 (with the original duration $d_3 = 5$) is only forward-inflexible. Hence, the conclusion is that activity 3 cannot be bi-critical. In addition, it is simple to verify that also a delay on the starting time of activity 3 implies an increase on the project duration even if we decrease the activity duration; therefore, activity 3, according to De Reyck's definition, should be forward-critical.

3.2 The case of a critical path from a critical activity with length equal to the activity duration

Let us continue the analysis of Example 1 by examining the criticality type of activity 4, again following the scheme proposed by De Reyck (1998). The critical relations involving activity 4 would imply that this activity is start-critical. Also in this case the conclusion is not true, because activity 4 is not bi-flexible as it should be if it was only start-critical. In fact, if we decrease by 1 the activity duration and, hence, assume $d_4 = 4$, nothing will change on the standardized network and then the critical path will remain the same with the same length equal to 20. If, on the contrary, we increase by 1 the duration of activity 4 and, hence, consider $d_4 = 6$, the length of the longest path from node 4 to node 6 will be $5 < d_4 = 6$. Consequently, the finish time of activity 4 should be 21, while the length of the critical path would remain unchanged and equal to 20. Since this cannot happen, in order to overcome this contradiction arc $(4, 6)$ of length $d_4 = 6$ must be added to exactly compute the completion time of the project (see Figure 4). In doing so, the length of the longest path from node 4 to node 6 continues to be equal to d_4, even if the duration of activity 4 increases. Note that this corresponds to add precedence relation $FS_{46}^{min}(0)$ from activity 4 to dummy final activity 6. The project completion time would increase since

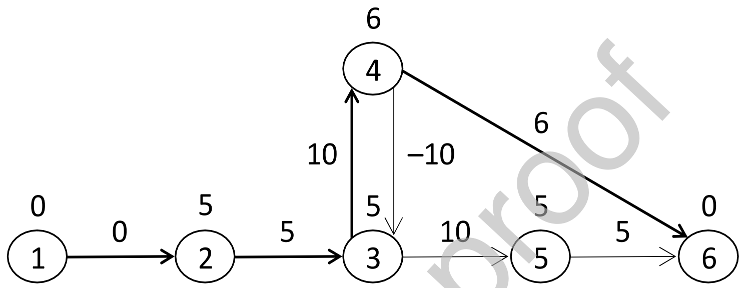

Figure 4: The standardized network of the project network in Figure 1, with $d_4 = 6$, and the addition of arc $(4, 6)$ of length $\ell_{46} = d_4$ for its correction (in bold the critical path)

the critical path would be (1 — 2 — 3 — 4 — 6) of length 21. This leads to observe that activity 4 cannot be only start-critical since it is backward-flexible and forward-inflexible. Therefore, the length of the critical path increases if both the starting time of activity 4 and/or its duration are increased. Consequently, according to De Reyck's definition, activity 4 should be forward-critical.

Finally, according to the analysis of De Reyck, activities 2 and 5 are forward-critical and finish-critical, respectively, and the flexibility analysis confirms these criticalities.

3.3 The case of a zero length critical path from node 1 to a critical node

Example 2. Let us consider now the network of Figure 5. The corresponding standardized network is shown in Figure 6.

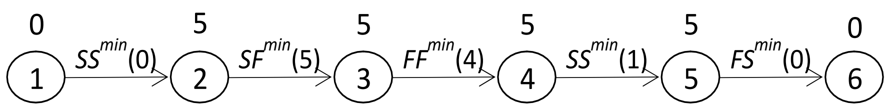

Figure 5: The GPRs project network of Example 2

Clearly, all the activities are critical and the critical path is (1 — 2 — 3 — 4 — 5 — 6) of length 10. The earliest start ES_3 of activity 3 is equal to 0, because the length of the

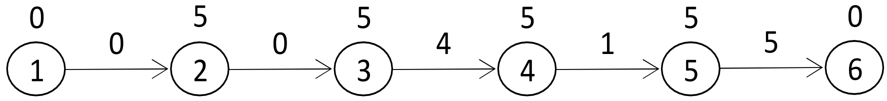

Figure 6: The standardized network associated with the project network of Figure 5

longest path from node 1 to node 3 is equal to 0. On the basis of De Reyck's scheme, activity 3 should be finish-critical, because the ingoing critical precedence of activity 3 is of type *SF* and its outgoing critical precedence is of type *FS*. Consequently, activity 3 should be bi-flexible. However, this is not true. In fact, if the duration of activity 3 increases by 1, then time lag ℓ_{23} becomes equal to -1. This would imply that ES_3 becomes equal to -1, that is, less than the project start time (assumed to be equal to 0), but this clearly cannot occur. To avoid this anomaly and force activity 3 to start not before the beginning of the project, we have to add arc $(1, 3)$ with length $\ell_{13} = 0$ to the standardized network (see Figure 7). In doing so, the length of the longest path from node 1 to node 3 continues to be equal to 0, even if the duration of activity 3 increases. The new arc $(1, 3)$ corresponds to consider the additional precedence relation $SS^{min}_{13}(0)$ from the dummy initial activity 1 to activity 3 on the project network with GPRs of Figure 5. The consequence is that we have two critical paths (1 — 2 — 3 — 4 — 5 — 6) and (1 — 3 — 4 — 5 — 6). According to the first path, activity 3 is finish-critical and according to the second path is forward-critical, and, hence, forward-inflexible and not bi-flexible. It is easy to verify also that activity 2 is start-critical and activity 4 is backward-critical. In fact, if we modify the value of d_2 the project duration will not change, and, hence, activity 2 is bi-flexible. If we reduce by 1 the duration of activity 4, and, hence, assuming $d_4 = 4$, we will get $\ell_{34} = 5$ and an increase of the project duration to 11; on the contrary, if the duration of activity 4 is increased by 1 the critical path length will decrease by the same amount. Therefore, activity 4 is only backward-inflexible. The conclusion is that, also in this case, the method proposed by De Reyck fails, with reference to activity 3, and it is necessary to correct the network to find the actual criticality.

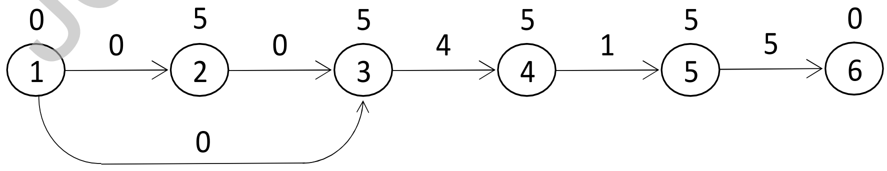

Figure 7: The correction of the standardized network of Figure 6

3.4 List of failures

Examples 1 and 2 of this section, together with other examples discussed in Appendix B of the paper *supplemental document*, show that it may be necessary to add further corrections to both standardized and GPRs networks, in addition to those already proposed by De Reyck (1998). Moreover, Example 1 shows that the method proposed by De Reyck for analyzing activity criticalities may fail also when the critical paths are not elementary. In the case when the method fails, e.g., when there are critical cycles involving a critical activity with many different precedence relations, nothing is said about how to establish which are the dominating criticalities of that activity.

Summing up, we may notice that:

1. The definitions of forward-critical and backward-critical present ambiguities with respect to the definitions of start- and finish-critical, respectively (see Section 4.2).
2. Both the standardized network and the original network (with minimum time lags) may need further appropriate corrections with respect to those suggested by De Reyck.
3. The rules proposed by De Reyck for the identification of the type of criticality work correctly when on the network (modified as prescribed in the previous point) there are only elementary critical paths. When there are non-elementary critical paths, the proposed rules may fail (see previous examples).
4. When a critical cycle is included in a (non-elementary) critical path, we should resort to the flexibility analysis in order to identify the correct activity dominating criticalities (see previous examples).

4 Our proposal

First of all, we start by giving additional corrections to the standardized network, and, consequently, on the GPRs network with minimum time lag, which may be required as shown, for example, by the analysis of Examples 1 and 2 of Section 3.

4.1 Additional network corrections

The same corrections reported at the end of Section 2 are also required when the conditions of the following propositions occur (see Appendix A of the paper *supplemental document* for their proofs).

Proposition 4.1 *Given a critical activity i such that the longest path in the standardized network from node 1 to node i is equal to 0, if the ingoing critical precedence relations of activity i are only of type XF_{hi}^{min} then we need to add (critical) precedence relation $SS_{1i}^{min}(0)$ between dummy activity 1 and activity i in order to correctly analyze what happens when the duration of i increases.*

Proposition 4.2 *Given a critical activity i whose duration is equal to the longest path from node i to the sink node n in the standardized network, if the outgoing critical precedence relations of activity i are only of type SX_{ij}^{min} then we need to add (critical) precedence relation $FS_{in}^{min}(0)$ between activity i and dummy activity n in order to correctly analyze what happens when the duration of i increases.*

4.2 Redefinition of criticalities

Before illustrating a new method for analyzing criticalities and flexibilities of the activities, we redefine, in a quite different way, the activity criticalities, in comparison with those given by De Reyck (1998), and reported in Demeulemeester and Herroelen (2002), and those given by Elmaghraby and Kamburowski (1992). Moreover, we propose additional types of criticalities. In particular, all our definitions are abstract and, therefore, independent from the project network representation. As for activity flexibilities, we adopt the same definitions introduced by Elmaghraby and Kamburowski (1992) and also adopted by De Reyck (1998).

With GPRs, we have to distinguish among different types of criticality, because, differently from the project networks with SPRs, the phenomenon firstly studied by Wiest (1981) and recalled in Section 1 may occur (see, e.g., activity 4 of Example 2). Moreover, it may happen that varying the duration of an activity does not imply any increase on the project duration, while delaying only its start (finish) time does (see, e.g., activity 2 of Example 2). In addition, the presence of cycles may have an effect on the project time-feasibility, as said in Section 3. We underline that the concept of activity criticality below defined is strictly related to a possible finite project duration increase. Therefore, in the following, we characterize the different types of criticality of a real activity, apart from the project time-infeasibility.

Definition 4.3 *An activity is* start-critical (finish-critical) *if it is critical and the project duration increases only if we delay the activity start (finish) time.*

Apart from the project time-infeasibility, the previous definition implies that, given a start-critical (finish-critical) activity, if we fix its start (finish) time and increase or decrease its duration, and, hence, either increase or decrease its finish (start) time, the project duration does not change. Therefore, the finish (start) time of a start-critical (finish-critical) activity is not constrained.

Since, by definition, we have in both cases the freedom to increase or decrease the activity duration without incurring a project duration increase, we conclude that start-critical activities and finish-critical activities are bi-flexible (apart from the project time-infeasibility).

Definition 4.4 *An activity is* forward-critical (backward-critical) *if it is critical and the project duration increases whether we delay its start (finish) time, while maintaining fixed* *its duration, or we increase (decrease) its duration while maintaining fixed its start (finish) time (apart from the project time-infeasibility).*

The consequence is that, in anyone of the above two cases, also the activity finish (start) time increases, meaning that the forward-criticality (backward-criticality) *dominates* the finish-criticality (start-criticality).

In general, given a forward-critical (backward-critical) activity, we have the opportunity to decrease (increase) its duration without increasing the project duration. Note that if we decrease (increase) the duration of a forward-critical (backward-critical) activity we have the chance of delaying the start (finish) time of the activity, without increasing its finish (start) time, or to finish (start) the activity earlier. In the first case, we have an increase of the activity start (finish) time without causing an increase of the project duration; therefore, in general, there is no dominance relation between forward-criticality and start-criticality (backward-criticality and finish-criticality). Note that this is in contrast with the definition of forward-critical given by De Reyck (1998), and reported at page 124 of the book of Demeulemeester and Herroelen (2002), where a forward-critical (backward-critical) activity is said to be start-critical (finish-critical) and such that an increase (decrease) of its duration implies an increase of the project duration.

Moreover, apart from the project time-infeasibility, since, by definition, an increase (decrease) of the duration of a forward-critical (backward-critical) activity increases the project duration, a forward-critical (backward-critical) activity is forward-inflexible (backward-inflexible).

Definition 4.5 *An activity is* bi-critical *if it is both forward-critical and backward-critical.*

Therefore, by Definition 4.4 and apart from the project time-infeasibility, both lengthening and shortening such an activity increases the project duration, meaning that a bi-critical activity is bi-inflexible. Clearly, also by delaying the start time or the finish time of such an activity we get an increase of the project duration.

The dominances established on the basis of the previous definitions call for additional new types of criticalities.

Definition 4.6 *An activity is* start-&-forward-critical (finish-&-backward-critical) *if it is critical and the project duration increases whether we increase (decrease) the activity duration or we delay its start (finish) time despite decreasing (increasing) its duration.*

Therefore, although we can decrease (increase) the duration of a start-&-forward-critical (finish-&-backward-critical) activity without lengthening the project (assuming that the activity is not also backward-critical (forward-critical)), we can accomplish this only by finishing (starting) the activity earlier and not by delaying its start (finish) time. Moreover, we cannot delay the finish (start) time of a start-&-forward-critical (finish-&-backward-critical) activity because the forward-criticality (backward-criticality) dominates the finish-criticality (start-criticality). Of course, apart from the project time-infeasibility, a start-&-forward-critical (finish-&-backward-critical) activity is forward-inflexible (backward-inflexible) due to the forward-criticality (backward-criticality).

Obviously, if an activity is start-&-forward-critical or forward-critical, and also finish-&-backward-critical or backward-critical, then it is bi-critical and, hence, bi-inflexible.

Definition 4.7 *An activity is* start-&-finish-critical *if it is critical and the project duration increases when we delay the activity start time or its finish time, even if the activity duration changes.*

The criticality analysis provided in Section 5 for Example 3 shows that such an activity may exist. Moreover, in Appendix A of the paper *supplemental document*, we show that this activity cannot be bi-flexible because either an increase or a decrease of its duration will make the project time-infeasible.

Finally, at the end of Appendix A, a comparison with the activity criticality classification by Valls and Lino (2001) shows that our criticality definitions include that of these authors, and allow a more detailed identification of the activity parameters responsible for the criticality.

4.3 Establishing the criticality and flexibility of a critical activity

The proposed method for identifying the criticality and flexibility of a critical activity i is based on the analysis of the structure of critical paths and cycles traversing node i.

4.3.1 Critical paths and critical cycles related to a critical activity

On the GPRs project network (with minimum time lags), let us denote with $P^s_{1i} = (1 \cdots \underline{XS}\, i)$ and with $P^f_{1i} = (1 \cdots \underline{XF}\, i)$ two generic possible longest paths from source node 1 to node i, with intermediate nodes distinct from the path extreme nodes: in the first path the ingoing arc of node i is of type XS, while it is of type XF in the latter. Analogously, let $P^s_{in} = (i\, \underline{SX} \cdots n)$ and $P^f_{in} = (i\, \underline{FX} \cdots n)$ be two generic possible longest paths from node i to sink node n, with intermediate nodes distinct from the path extreme nodes, and where the outgoing arcs of node i are of types SX and FX, respectively.

According to the transformations of Bartush *et al.* (1988), the length of P^s_{1i} is independent from duration d_i of activity i, while the length of P^f_{1i} is a decreasing function of d_i that decreases as much as d_i increases. Likewise, the length of P^s_{in} is independent from d_i, while the length of P^f_{in} is an increasing function of d_i that increases as much as d_i does.

Moreover, since we are interested in analyzing only the criticality and the flexibility of activity i, we may assume, without loss of generality, that paths P^s_{1i}, P^s_{in}, P^f_{1i}, and P^f_{in} are elementary. In fact, if they contain a cycle, the latter will pass only through nodes distinct from i and of course the cycle would be of length zero. Therefore, we can always neglect such a cycle in our analysis.

By definition, it follows that a real activity i is critical if and only if it lays on a critical path of the standardized network, that is, there is a longest (i.e., critical) path $P^{yz}_{1n;i}$ from node 1 to node n passing through node i, formed by the concatenation of paths P^{y}_{1i} and P^{z}_{in}, i.e., $P^{yz}_{1n;i} = P^{y}_{1i} \bigoplus P^{z}_{in}$, with $y, z \in \{s, f\}$. For a critical activity i, besides the elementary critical paths of type P^{y}_{1i} and P^{z}_{in}, node i may belong to additional p_i zero length (i.e., critical) cycles $C^k_i = (i \cdots j_k \cdots i)$, with $k = 1, \ldots, p_i$. Analogously, without loss of generality, we may assume that also these cycles are elementary (i.e., no node of the cycle but the first and last appears twice). Therefore, denoting with $\mathcal{C}_i$ the concatenation of these cycles (with the exception of the cycle already contained in path $P^{yz}_{1n;i}$ if this is not elementary), let $\hat{P}^{yz}_{1n;i} = P^{y}_{1i} \bigoplus \mathcal{C}_i \bigoplus P^{z}_{in}$ be the generic structure of a critical path passing through node i.

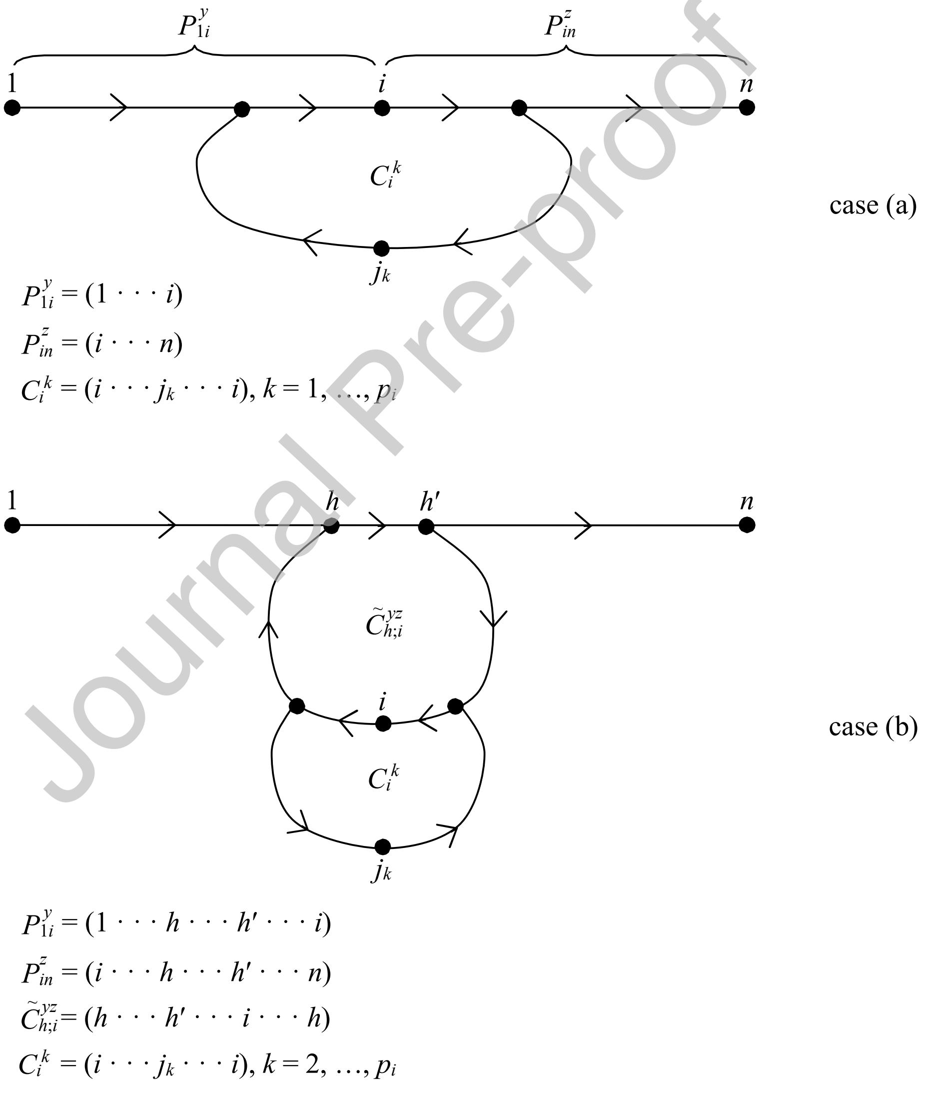

Figure 8: The two possible structures of a critical path $\hat{P}^{yz}_{1n;i} = P^{y}_{1i} \bigoplus \mathcal{C}_i \bigoplus P^{z}_{in}$ traversing critical real activity i

We have to distinguish two alternative cases, shown in Figure 8:

(a) There is no other non-dummy node $h \neq i$ traversed by both P^y_{1i} and P^z_{in}, and, hence, critical path $P^{yz}_{1n;i} = P^y_{1i} \bigoplus P^z_{in} = (1 \cdots i \cdots n)$ is elementary.

(b) There is at least a common intermediate non-dummy node $h \neq i$ of P^y_{1i} and of P^z_{in}. Hence, critical path $P^{yz}_{1n;i} = P^y_{1i} \bigoplus P^z_{in} = (1 \cdots h \cdots i \cdots h \cdots n)$ is not elementary and contains the zero length elementary cycle $\tilde{C}^{yz}_{h;i} = (h \cdots i \cdots h)$, where, without loss of generality, h is the first common intermediate node traversed by P^y_{1i} if there were more than one.

As for case (a), critical path $\hat{P}^{yz}_{1n;i}$ is formed by elementary path $P^{yz}_{1n;i} = (1 \cdots i \cdots n)$ and $p_i \geq 0$ elementary cycles $C^k_i = (i \cdots j_k \cdots i)$, with $k = 1, \ldots, p_i$. As for case (b), critical path $\hat{P}^{yz}_{1n;i}$ is formed by non-elementary path $P^{yz}_{1n;i} = (1 \cdots h \cdots i \cdots h \cdots n)$, containing exactly one elementary cycle $\tilde{C}^{yz}_{h;i} = (h \cdots i \cdots h)$ corresponding, e.g., to C^1_i, and $p_i - 1 \geq 0$ elementary cycles $C^k_i = (i \cdots j_k \cdots i)$, with $k = 2, \ldots, p_i$.

The following example clarifies the previous formalism and the two different cases.

Example 3. Let us consider the project network with GPRs of Figure 9 with $n = 6$.

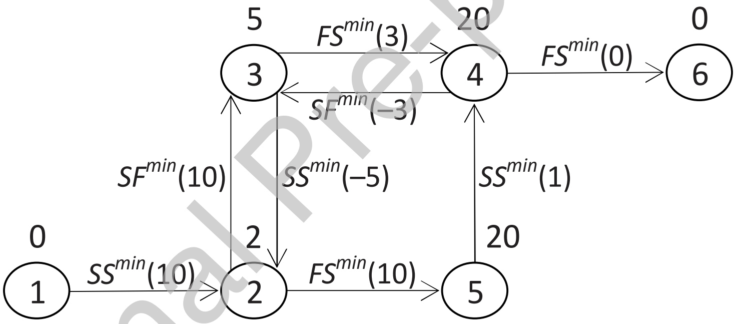

Figure 9: The project network with GPRs of Example 3

It is easy to verify that all the activities are critical and that the critical network coincides with the given network which contains some distinct cycles. Let us consider critical activity 3. We have two critical elementary paths from source node 1 to node 3, i.e., paths $P^f_{13}(1) = (1 \text{ — } 2 \overset{\boldsymbol{SF}}{\text{—}} \mathbf{3})$ and $P^f_{13}(2) = (1 \text{ — } 2 \text{ — } 5 \text{ — } 4 \overset{\boldsymbol{SF}}{\text{—}} \mathbf{3})$. We have two critical elementary paths from node 3 to sink node 6, i.e., paths $P^s_{36}(1) = (\mathbf{3} \overset{\boldsymbol{SS}}{\text{—}} 2 \text{ — } 5 \text{ — } 4 \text{ — } 6)$ and $P^f_{36}(2) = (\mathbf{3} \overset{\boldsymbol{FS}}{\text{—}} 4 \text{ — } 6)$. Moreover, we have $p_3 = 3$ critical elementary cycles passing through node 3, i.e., cycles $C^1_3 = (\mathbf{3} \overset{\boldsymbol{SS}}{\text{—}} 2 \overset{\boldsymbol{SF}}{\text{—}} \mathbf{3})$, $C^2_3 = (\mathbf{3} \overset{\boldsymbol{SS}}{\text{—}} 2 \text{ — } 5 \text{ — } 4 \overset{\boldsymbol{SF}}{\text{—}} \mathbf{3})$, and $C^3_3 = (\mathbf{3} \overset{\boldsymbol{FS}}{\text{—}} 4 \overset{\boldsymbol{SF}}{\text{—}} \mathbf{3})$.

Therefore, we have four critical paths of type $P^{yz}_{16;3} = P^y_{13} \bigoplus P^z_{36}$, i.e., paths $P^{fs}_{16;3}(1) = (1 \text{ — } [2 \overset{\boldsymbol{SF}}{\text{—}} \mathbf{3} \overset{\boldsymbol{SS}}{\text{—}} 2] \text{ — } 5 \text{ — } 4 \text{ — } 6)$, $P^{ff}_{16;3}(2) = (1 \text{ — } 2 \overset{\boldsymbol{SF}}{\text{—}} \mathbf{3} \overset{\boldsymbol{FS}}{\text{—}} 4 \text{ — } 6)$, $P^{fs}_{16;3}(3) = (1 \text{ — } [2 \text{ — } 5 \text{ — } 4 \overset{\boldsymbol{SF}}{\text{—}} \mathbf{3} \overset{\boldsymbol{SS}}{\text{—}} 2] \text{ — } 5 \text{ — } 4 \text{ — } 6)$, and $P^{ff}_{16;3}(4) = (1 \text{ — } 2 \text{ — } 5 \text{ — } [4 \overset{\boldsymbol{SF}}{\text{—}} \mathbf{3} \overset{\boldsymbol{FS}}{\text{—}} 4] \text{ — } 6)$. Path $P^{ff}_{16;3}(2)$ is elementary, while paths $P^{fs}_{16;3}(1)$, $P^{fs}_{16;3}(3)$, and $P^{ff}_{16;3}(4)$ are not elementary and contain cycles $\tilde{C}^{fs}_{2;3}(1) = (2 \overset{\boldsymbol{SF}}{\text{—}} \mathbf{3} \overset{\boldsymbol{SS}}{\text{—}} 2)$, $\tilde{C}^{fs}_{2;3}(3) = (2 \text{ — } 5 \text{ — } 4 \overset{\boldsymbol{SF}}{\text{—}} \mathbf{3} \overset{\boldsymbol{SS}}{\text{—}} 2)$, and

$\tilde{C}^{ff}_{4;3}(4) = (4 \; \underline{\boldsymbol{SF}} \; \mathbf{3} \; \underline{\boldsymbol{FS}} \; 4)$, respectively; in particular, $\tilde{C}^{fs}_{2;3}(1) \equiv C^1_3$, $\tilde{C}^{fs}_{2;3}(3) \equiv C^2_3$, and $\tilde{C}^{ff}_{4;3}(4) \equiv C^3_3$.

Finally, we have four critical paths $\hat{P}^{fs}_{16;3}(1), \hat{P}^{ff}_{16;3}(2), \hat{P}^{fs}_{16;3}(3), \hat{P}^{ff}_{16;3}(4)$ of type $\hat{P}^{yz}_{16;3} = P^y_{13} \bigoplus \mathcal{C}_3 \bigoplus P^z_{36}$, each one obtained by inserting the concatenation $\mathcal{C}_3$ of the critical elementary cycles C^k_3, $k = 1, 2, 3$ (with the exception of the cycle already contained in $P^{yz}_{16;3}$ if non-elementary), between P^y_{13} and P^z_{36}. An analog analysis can be done for the other (critical) activities.

4.3.2 Identifying criticalities

In Appendix A of the paper *supplemental document*, we show that for identifying the criticality of a given critical activity i, we may restrict the analysis to the critical paths $P^{yz}_{1n;i}$ only, while the critical elementary cycles $C^k_i = (i \cdots j_k \cdots i)$, with $k = 1, \ldots, p_i$, are responsible for any project time-infeasibility due to a variation of the activity duration. In addition, the proofs of the following results are provided.

Let us denote with $P^{*s}_{1n;i} = (1 \cdots h \cdots \underline{XX}\; i \; \underline{SX} \cdots h \cdots n)$ ($P^{*f}_{1n;i} = (1 \cdots h \cdots \underline{XX}\; i \; \underline{FX} \cdots h \cdots n)$) a generic critical non-elementary path passing through node i, containing elementary cycle $\tilde{C}^{*s}_{i;h} = (h \cdots \underline{XX}\; i \; \underline{SX} \cdots h)$ ($\tilde{C}^{*f}_{i;h} = (h \cdots \underline{XX}\; i \; \underline{FX} \cdots h)$).

Proposition 4.8 *Activity i is* start-critical *if it belongs only to critical elementary paths of type $P^{ss}_{1n;i} = (1 \cdots \underline{XS}\; i \; \underline{SX} \cdots n)$, or only to critical non-elementary paths of type $P^{*s}_{1n;i}$ containing the elementary cycle of type $\tilde{C}^{*s}_{i;h} = (h \cdots \underline{XX}\; i \; \underline{SX} \cdots h)$, or only to paths of both types.*

Proposition 4.9 *Activity i is* finish-critical *if it belongs only to critical elementary paths of type $P^{ff}_{1n;i} = (1 \cdots \underline{XF}\; i \; \underline{FX} \cdots n)$, or only to critical non-elementary paths of type $P^{*f}_{1n;i}$ containing the elementary cycle of type $\tilde{C}^{*f}_{i;h} = (h \cdots \underline{XX}\; i \; \underline{FX} \cdots h)$, or only to paths of both types.*

Proposition 4.10 *Activity i is* forward-critical *if it belongs to a critical elementary path of type $P^{sf}_{1n;i} = (1 \cdots \underline{XS}\; i \; \underline{FX} \cdots n)$ and does not belong to any critical elementary path of type $P^{ss}_{1n;i} = (1 \cdots \underline{XS}\; i \; \underline{SX} \cdots n)$ or of type $P^{fs}_{1n;i} = (1 \cdots \underline{XF}\; i \; \underline{SX} \cdots n)$, nor to any critical non-elementary path of type $P^{*s}_{1n;i}$ containing the elementary cycle of type $\tilde{C}^{*s}_{i;h} = (h \cdots \underline{XX}\; i \; \underline{SX} \cdots h)$.*

Proposition 4.11 *Activity i is* backward-critical *if it belongs to a critical elementary path of type $P^{fs}_{1n;i} = (1 \cdots \underline{XF}\; i \; \underline{SX} \cdots n)$ and does not belong to any critical elementary path of type $P^{ff}_{1n;i} = (1 \cdots \underline{XF}\; i \; \underline{FX} \cdots n)$ or of type $P^{sf}_{1n;i} = (1 \cdots \underline{XS}\; i \; \underline{FX} \cdots n)$, nor to any critical non-elementary path of type $P^{*f}_{1n;i}$ containing the elementary cycle of type $\tilde{C}^{*f}_{i;h} = (h \cdots \underline{XX}\; i \; \underline{FX} \cdots h)$.*

Proposition 4.12 *Activity i is* bi-critical *if it belongs to a critical elementary path of type $P^{sf}_{1n;i} = (1 \cdots \underline{XS}\; i\; \underline{FX} \cdots n)$ and to a critical elementary path of type $P^{fs}_{1n;i} = (1 \cdots \underline{XF}\; i\; \underline{SX} \cdots n)$.*

Proposition 4.13 *Activity i is* start-&forward-critical *if it belongs to a critical elementary path of type $P^{sf}_{1n;i} = (1 \cdots \underline{XS}\; i\; \underline{FX} \cdots n)$ and, in addition, to a critical elementary path of type $P^{ss}_{1n;i} = (1 \cdots \underline{XS}\; i\; \underline{SX} \cdots n)$ and/or to a critical non-elementary path of type $P^{*s}_{1n;i}$ containing the elementary cycle of type $\tilde{C}^{*s}_{i;h} = (h \cdots \underline{XX}\; i\; \underline{SX} \cdots h)$, but not to any critical elementary path of type $P^{fs}_{1n;i} = (1 \cdots \underline{XF}\; i\; \underline{SX} \cdots n)$.*

Proposition 4.14 *Activity i is* finish-&-backward-critical *if it belongs to a critical elementary path of type $P^{fs}_{1n;i} = (1 \cdots \underline{XF}\; i\; \underline{SX} \cdots n)$ and, in addition, to a critical elementary path of type $P^{ff}_{1n;i} = (1 \cdots \underline{XF}\; i\; \underline{FX} \cdots n)$ and/or to a critical non-elementary path of type $P^{*f}_{1n;i}$ containing the elementary cycle of type $\tilde{C}^{*f}_{i;h} = (h \cdots \underline{XX}\; i\; \underline{FX} \cdots h)$, but not to any critical elementary path of type $P^{sf}_{1n;i} = (1 \cdots \underline{XS}\; i\; \underline{FX} \cdots n)$.*

Proposition 4.15 *Activity i is* start-&-finish-critical *if all the following three conditions hold:*

1. *Activity i belongs to a critical elementary path of type $P^{ss}_{1n;i} = (1 \cdots \underline{XS}\; i\; \underline{SX} \cdots n)$ and/or to a critical non-elementary path of type $P^{*s}_{1n;i}$ containing the elementary cycle of type $\tilde{C}^{*s}_{i;h_1} = (h_1 \cdots \underline{XX}\; i\; \underline{SX} \cdots h_1)$;*
2. *Activity i belongs to a critical elementary path of type $P^{ff}_{1n;i} = (1 \cdots \underline{XF}\; i\; \underline{FX} \cdots n)$ and/or to a critical non-elementary path of type $P^{*f}_{1n;i}$ containing the elementary cycle of type $\tilde{C}^{*f}_{i;h_2} = (h_2 \cdots \underline{XX}\; i\; \underline{FX} \cdots h_2)$;*
3. *Activity i does not belong to any critical elementary path of type $P^{sf}_{1n;i} = (1 \cdots \underline{XS}\; i\; \underline{FX} \cdots n)$ or $P^{fs}_{1n;i} = (1 \cdots \underline{XF}\; i\; \underline{SX} \cdots n)$.*

4.3.3 Identifying inflexibilities in terms of project time-infeasibility due to variations of a critical activity duration

We say that an activity is *forward-time-infeasible* (*backward-time-infeasible*) if an increase (decrease) of its duration causes a project time-infeasibility; moreover, if it is both forward-time-infeasible and backward-time-infeasible, we say that it is *bi-time-infeasible*; finally, if neither an increase nor a decrease of the activity duration cause a project time-infeasibility, we say that the activity is *bi-time-feasible*.

Let us consider the GPRs critical sub-network (with minimum time lags), and let us show how to identify, from this network, a possible project time-infeasibility due to a variation of the duration of a critical real activity i.

It is clear that the forward-time infeasibility (backward-time infeasibility) for a critical activity i is related to the existence of a (zero length) critical elementary cycle containing node i whose length becomes positive if we increase (decrease) the activity duration. Therefore, we may restrict the analysis by examining the set of $p_i \geq 0$ critical elementary cycles $C_i^k = (i \xrightarrow{Y_k X} \cdots j_k \cdots \xrightarrow{X Z_k} i)$, with $1 \leq k \leq p_i$ and $Y_k, Z_k \in \{S, F\}$.

It is straightforward to show that the length of cycle $C_i^k = (i \xrightarrow{Y_k X} \cdots j_k \cdots \xrightarrow{X Z_k} i)$ does not depend on the duration d_i of activity i if $Y_k = Z_k \in \{S, F\}$, while it is an increasing (decreasing) function of d_i if $Y_k = F$ and $Z_k = S$ ($Y_k = S$ and $Z_k = F$). Hence,

Proposition 4.16 *Critical activity i is* forward-time-infeasible *if it belongs at least to one critical cycle C_i^k of type $(i \xrightarrow{FX} \cdots j_k \cdots \xrightarrow{XS} i)$.*

Proposition 4.17 *Critical activity i is* backward-time-infeasible *if it belongs at least to one critical cycle C_i^k of type $(i \xrightarrow{SX} \cdots j_k \cdots \xrightarrow{XF} i)$.*

Clearly, if activity i is forward-time-infeasible (backward-time-infeasible) then it is forward-inflexible (backward-inflexible), and if it both forward-time-infeasible and backward-time-infeasible, and, hence, bi-time-infeasible, then it is bi-inflexible.

4.4 A new method for activity criticality and flexibility analysis

On the basis of what previously discussed, we propose the following approach to identify activity criticalities and flexibilities:

1. Adopt the AON project network representation with minimum time lags, after having converted the precedences with maximum time lags into the corresponding ones with minimum time lags.

2. Convert the given network into the standardized network using the transformations of Bartush *et al.* (1988), with only GPRs of type $SS_{ij}^{min}(\ell)$, and, eventually, the corrections suggested by De Reyck.

3. Find the project length on the standardized network, along with the earliest and latest start (finish) times of the activities, which are useful to determine the *critical subnetwork* composed by all the critical activities and all the critical arcs (i.e., critical precedences among critical activities) on the standardized network.

4. Correct the GPRs network (and hence the standardized network), if necessary, with the addition of new (critical) arcs outgoing from source node 1 and/or ingoing to sink node n, according to Propositions 4.1 and 4.2.

5. Trace back the critical activities and the critical arcs on the (corrected) GPRs project network (with minimum time lags) in order to consider only its critical subnetwork.

6. For each critical real activity i,

   (a) Determine the criticality type of activity i, according to Propositions 4.8–4.15.

(b) Determine possible project time-infeasibility of activity i, according to Propositions 4.16 and 4.17.

7. Analyze the flexibility of non-critical activities in order to detect possible project time-infeasibility due to duration changing for these activities.

In the following, we describe in details some of the above steps.

4.4.1 Determining the critical subnetwork on the standardized network

As recalled before, finding the subcritical network of the standardized network requires to determine the earliest start time ES_i and the latest start time LS_i of each activity i. The ES_i can be computed in $O(nm)$ time, with m being the number of arcs of the standardized network, with the Bellman-Ford algorithm (see, e.g., Bellman, 1958, Moore, 1959, and Ford and Fulkerson, 1962) for the single origin-multiple destination longest path problem (assuming the single origin in node 1). Let $T = ES_n$ be the project duration. Analogously, LS_i can be computed in $O(nm)$ time by computing the length λ_i of the longest path from node i to node n, with the reverse version of the Bellman-Ford algorithm for the multiple origin-single destination longest path problem (assuming the single destination in node n), and by calculating $LS_i = T - \lambda_i$. The critical subnetwork is the subnetwork of the standardized network containing all the (critical) nodes i (i.e., for which $LS_i = ES_i$) and the (critical) arcs (i, j) of the network among critical nodes, such that the arc length $\ell_{ij} = ES_j - ES_i$.

4.4.2 Determining the criticality type of a critical activity

Given the critical sub-network $\hat{N} = (\hat{V}, \hat{A})$ with GPRs (with only minimum time lags), next we show how to determine the criticality type of a (real) critical activity i, according to Propositions 4.8–4.15. Even if, on the basis of these propositions, finding the criticalities of i requires to analyze the structures of the paths of type $P^{yz}_{1n;i} = P^y_{1i} \bigoplus P^z_{in}$, with $y, z \in \{s, f\}$, it is not necessary to list all such paths. In fact, as proved in Appendix A of the paper *supplemental document*, the criticalities of i depend on precedence type of the last arc of any elementary path of type P^y_{1i} and the precedence type of the first arc of any elementary path of P^z_{in}, composing a critical path of type $P^{yz}_{1n;i} = P^y_{1i} \bigoplus P^z_{in}$, and upon the latter being elementary or not.

Therefore, the analysis can be done by considering ordered couples $[(h, i), (i, j)]$ of arcs incident to node i with arcs (h, i) and (i, j) belonging to the set of ingoing arcs $\Gamma^-_{\hat{N}}(i)$ and outgoing arcs $\Gamma^+_{\hat{N}}(i)$ of node i, respectively, on the critical sub-network $\hat{N}$.

Indeed, we can exclude from the analysis the ordered arc couples with first arc $(h, i) \in \Gamma^-_{\hat{N}}(i)$ such that there is no elementary path of type P^y_{1i} ending with arc (h, i), and analogously we can exclude the analysis of ordered arc couples with the second arc $(i, j) \in \Gamma^+_{\hat{N}}(i)$ such that there is no elementary path of type P^z_{in} starting with arc (i, j). Ingoing arcs

$(h,i) \in \Gamma^-_{\hat{N}}(i)$ to be excluded can be easily recognized by checking if there is no elementary path from node 1 to node h without i as intermediate node (see, e.g., arc $(4,3)$ in Figure 1, assuming $h = 4$ and $i = 3$). The latter can be done by checking the reachability of node h from node 1 on the subnetwork $\hat{N} - i$ (obtained by removing node i and all its ingoing and outgoing arcs from $\hat{N}$), by performing a reverse visit on this subnetwork from node h to node 1. Similarly, outgoing arcs $(i,j) \in \Gamma^+_{\hat{N}}(i)$ to be excluded can be easily recognized by checking if there is no elementary path from node j to node n without i as intermediate node (see, e.g., arc $(3,4)$ in Figure 1, assuming $i = 3$ and $j = 4$). The latter can be done by checking the reachability of node n from node j on the subnetwork $\hat{N} - i$, by performing a forward visit on this subnetwork from node j to node n. In both cases, these network visits can be done in linear time with respect to the number of arcs of $\hat{N} - i$.

Therefore, given network $\hat{N}$, let $\tilde{\Gamma}^-_{\hat{N}}(i)$ and $\tilde{\Gamma}^+_{\hat{N}}(i)$ be the subsets of ingoing and outgoing arcs of node i, respectively, such that for each $(h,i) \in \tilde{\Gamma}^-_{\hat{N}}(i)$ there is an elementary path from 1 to i ending with arc (h,i), and for each $(i,j) \in \tilde{\Gamma}^+_{\hat{N}}(i)$ there is an elementary path from i to n starting with arc (i,j).

Table 1 shows the types of ordered couples $[(h,i),(i,j)]$ of ingoing and outgoing arcs of a given critical activity i, based on the related precedence types and the existence or not of an elementary critical path $P^{yz}_{1n;i} = (1 \cdots h — i — j \cdots n)$ passing through these arcs.

Table 1: Types of ordered couples $[(h,i),(i,j)]$ of ingoing and outgoing arcs of a given critical activity i.

Type	$(h,i) \in \tilde{\Gamma}^-_{\hat{N}}(i)$	$(i,j) \in \tilde{\Gamma}^+_{\hat{N}}(i)$	$P^{yz}_{1n;i} = (1 \cdots h — i — j \cdots n)$
1	$(h \overset{XS}{—} i)$	$(i \overset{SX}{—} j)$	whatever
2	$(h \overset{XF}{—} i)$	$(i \overset{SX}{—} j)$	all non-elementary paths
3	$(h \overset{XF}{—} i)$	$(i \overset{SX}{—} j)$	there exists an elementary path
4	$(h \overset{XF}{—} i)$	$(i \overset{FX}{—} j)$	whatever
5	$(h \overset{XS}{—} i)$	$(i \overset{FX}{—} j)$	all non-elementary paths
6	$(h \overset{XS}{—} i)$	$(i \overset{FX}{—} j)$	there exists an elementary path

In order to distinguish ordered couples $[(h,i),(i,j)]$ of type 2 (5) from those of type 3 (6), we need to check if there exists at least one elementary path $P^{yz}_{1n;i} = (1 \cdots h — i — j \cdots n)$ in the critical subnetwork $\hat{N}$. Clearly, this is unnecessary when $h \equiv 1$ or $j \equiv n$, because in this case path $P^{yz}_{1n;i}$ is always elementary. For the other cases, the complexity of this check and a method for doing it are shown in Appendix D of the paper *supplemental document.*

According to Propositions 4.8–4.15, the combinations of types of ordered couples of ingoing and outgoing arcs of activity i, according to the typology (elementary or not) of critical paths traversing the couple of arcs, determine the type of criticality of i. Table 2 reports the criticality identification scheme for a generic critical activity i.

Table 2: Criticality identification scheme for a given critical activity i.

Criticality	Requirements	Type	$(h,i) \in \tilde{\Gamma}^-_{\hat{N}}(i)$	$(i,j) \in \tilde{\Gamma}^+_{\hat{N}}(i)$	$P^{yz}_{1n;i} = (1 \cdots h — i — j \cdots n)$
start-critical	only at least one	1	$(h \underline{XS}\, i)$	$(i \underline{SX}\, j)$	whatever
	of these two types	2	$(h \underline{XF}\, i)$	$(i \underline{SX}\, j)$	all non-elementary paths
finish-critical	only at least one	4	$(h \underline{XF}\, i)$	$(i \underline{FX}\, j)$	whatever
	of these two types	5	$(h \underline{XS}\, i)$	$(i \underline{FX}\, j)$	all non-elementary paths
forward-critical	at least one of this type and none of types 1, 2, and 3	6	$(h \underline{XS}\, i)$	$(i \underline{FX}\, j)$	there exists an elementary path
backward-critical	at least one of this type and none of types 4, 5, and 6	3	$(h \underline{XF}\, i)$	$(i \underline{SX}\, j)$	there exists an elementary path
bi-critical	at least one for each	6	$(h \underline{XS}\, i)$	$(i \underline{FX}\, j)$	there exists an elementary path
	of these two types	3	$(h \underline{XF}\, i)$	$(i \underline{SX}\, j)$	there exists an elementary path
start-&-forward-critical	at least one of type 6	6	$(h \underline{XS}\, i)$	$(i \underline{FX}\, j)$	there exists an elementary path
	and one of type 1 or 2	1	$(h \underline{XS}\, i)$	$(i \underline{SX}\, j)$	whatever
	but none of type 3	2	$(h \underline{XF}\, i)$	$(i \underline{SX}\, j)$	all non-elementary paths
finish-&-backward-critical	at least one of type 3	3	$(h \underline{XF}\, i)$	$(i \underline{SX}\, j)$	there exists an elementary path
	and one of type 4 or 5	4	$(h \underline{XF}\, i)$	$(i \underline{FX}\, j)$	whatever
	but none of type 6	5	$(h \underline{XS}\, i)$	$(i \underline{FX}\, j)$	all non-elementary paths
start-&-finish-critical	at least one of type 1 or 2	1	$(h \underline{XS}\, i)$	$(i \underline{SX}\, j)$	whatever
	and one of type 4 or 5	2	$(h \underline{XF}\, i)$	$(i \underline{SX}\, j)$	all non-elementary paths
	but none of types 3 and 6	4	$(h \underline{XF}\, i)$	$(i \underline{FX}\, j)$	whatever
		5	$(h \underline{XS}\, i)$	$(i \underline{FX}\, j)$	all non-elementary paths

4.4.3 Determining possible project time-infeasibility of a critical activity

Given the critical sub-network $\hat{N}$ with GPRs (with only minimum time lags), let us show how to determine possible project time-infeasibility of a (real) critical activity i, according to Propositions 4.16 and 4.17. Although, on the basis of such propositions, this requires the analysis of the structure of the critical elementary cycles $C_i^k = (i \cdots j_k \cdots i)$, with $k = 1, \ldots p_i$, it is not necessary to list all such cycles. In fact, according to Proposition 4.16 (Proposition 4.17), we only need to check if there is an elementary cycle $(i \xrightarrow{FX} j \cdots h \xrightarrow{XS} i)$ (resp., $(i \xrightarrow{SX} j \cdots h \xrightarrow{XF} i)$) in the critical network $\hat{N}$. In particular, this check is required for each ordered couple $[(h \xrightarrow{XS} i), (i \xrightarrow{FX} j)]$ (resp. $[(h \xrightarrow{XF} i), (i \xrightarrow{SX} j)]$) of ingoing and outgoing critical arcs of node i, and it can be done in linear time with respect to the number of arcs of $\hat{N}$, by searching for an elementary path from node j to node h in $\hat{N}$ with a network visit.

4.4.4 Analyzing flexibility of non-critical activities

As for determining the flexibility (with respect to project time-infeasibility) of a non-critical real activity i, we first consider the duration of i equal to $d_i + 1$ and then equal to $d_i - 1$, and for each one of these two cases we find the longest path from node 1 to i with the Bellman-Ford algorithm with the positive length cycle halting condition. If the algorithm halts it means that the network contains a positive length cycle involving activity i which is therefore *forward-time-infeasible* and, hence, forward-inflexible (*backward-time-infeasible* and, hence, backward-inflexible) if the test has been made assuming the duration of i equal to $d_i + 1$ ($d_i - 1$). If this happens in both the two cases, the activity is bi-inflexible.

5 Examples of criticality and flexibility analysis with the proposed approach

We reconsider here the examples of Section 3 and Section 4.3, adopting the proposed criticality and flexibility analysis. Additional examples can be found in Appendix C of the paper *supplemental document.*

Example 1. Let us consider the project network with GPRs (with minimum time lags) of Figure 1, where $n = 6$, and its standardization shown in Figure 2.

The time analysis on the standardized network reveals that we need to correct the original (and the standardized) network adding precedence $FS_{46}^{min}(0)$, because in the standardized network the length of the longest path from the node representing critical activity 4 to dummy node 6 has length equal to the activity duration and activity 4 has only outgoing critical precedences of type SX (see Proposition 4.2).

From the time analysis on the corrected standardized network, we find that the critical subnetwork is the network itself. By tracing back the critical activities and the critical arcs on the corrected original GPRs project network, we get that also the critical subnetwork $\hat{N}$ of this network coincides with the latter, with all the (real) activities being critical. Figure 10 shows the critical subnetwork $\hat{N}$, where the number associated to node i represents the duration d_i of the activity represented by that node.

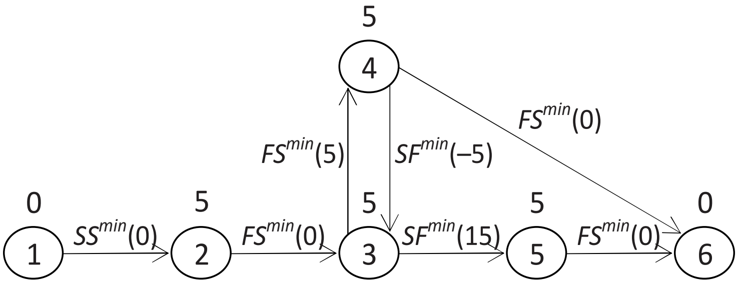

Figure 10: The critical network of Example 1

Node 2 has only one couple of ingoing and outgoing arcs, that is $[(1 \xrightarrow{SS} \mathbf{2}), (\mathbf{2} \xrightarrow{FS} 3)]$ which is of type 3 because there exists an elementary path of type $(1 \xrightarrow{SS} \mathbf{2} \xrightarrow{FS} 3 \cdots 6)$. Therefore (see Table 2), activity 2 results to be *forward-critical* (and, hence, forward-inflexible). Moreover, it is bi-time-feasible because there is no critical elementary cycle containing node 2. Hence, activity 2 turns out to be *forward-inflexible*.

Node 3 has two couples of ingoing and outgoing arcs to be considered, i.e., $[(2 \xrightarrow{FS} \mathbf{3}), (\mathbf{3} \xrightarrow{FS} 4)]$ and $[(2 \xrightarrow{FS} \mathbf{3}), (\mathbf{3} \xrightarrow{SF} 5)]$. Note that ingoing arc $(4, 3)$ should not be considered when looking at couples of ingoing and outgoing arcs of node 3 because there is no elementary path of type $(1 \cdots 4 — 3)$. Arc couple $[(2 \xrightarrow{FS} \mathbf{3}), (\mathbf{3} \xrightarrow{FS} 4)]$ is of type 6 because there exists an elementary path of type $(1 \cdots 2 \xrightarrow{FS} \mathbf{3} \xrightarrow{FS} 4 \cdots 6)$, while arc couple $[(2 \xrightarrow{FS} \mathbf{3}), (\mathbf{3} \xrightarrow{SF} 5)]$ is of type 1. Therefore, activity 3 results to be *start-&-forward-critical* (and, hence, forward-inflexible). Moreover, on the basis of the critical elementary cycle $(\mathbf{3} \xrightarrow{FS} 4 \xrightarrow{SF} \mathbf{3})$, it results to be bi-time-feasible. Hence, activity 3 turns out to be *forward-inflexible.*

Node 4 has two couples of ingoing and outgoing arcs, i.e., $[(3 \xrightarrow{FS} \mathbf{4}), (\mathbf{4} \xrightarrow{SF} 3)]$ and $[(3 \xrightarrow{FS} \mathbf{4}), (\mathbf{4} \xrightarrow{FS} 6)]$. Arc couple $[(3 \xrightarrow{FS} \mathbf{4}), (\mathbf{4} \xrightarrow{SF} 3)]$ is of type 1; arc couple $[(3 \xrightarrow{FS} \mathbf{4}), (\mathbf{4} \xrightarrow{FS} 6)]$ is of type 6 because there exists an elementary path of type $(1 \cdots 3 \xrightarrow{FS} \mathbf{4} \xrightarrow{FS} 6)$. Therefore, activity 4 results to be *start-&-forward-critical* (and, hence, forward-inflexible). Moreover, on the basis of the critical elementary cycle $(\mathbf{4} \xrightarrow{SF} 3 \xrightarrow{FS} \mathbf{4})$, it results to be bi-time-feasible. Hence, activity 4 turns out to be *forward-inflexible.*

Finally, node 5 has only one couple of ingoing and outgoing arcs, that is $[(3 \xrightarrow{SF} \mathbf{5}), (\mathbf{5} \xrightarrow{FS} 6)]$ which is of type 4. Therefore, activity 5 results to be *finish-critical.* Moreover, it is bi-time-feasible because there is no critical elementary cycle containing node 5. Hence, activity 5 turns out to be *bi-flexible.*

To better understand how a start-forward-critical activity works, let us analyze what happens for activity 4. From the (corrected) standardized network, we have that $ES_4 = 15$

and then $EF_4 = 20$, which is also equal to the project (minimum) duration T, since $T = ES_6 = 20$. If we delay by 1 the start time of activity 4 (with respect to its earliest start time), assuming therefore that it starts at time 16, and decrease its duration to 4, we have that its finish time remains equal to 20, but the earliest start time of activity 3 becomes equal to 6 and, consequently, the earliest finish time of activity 5 becomes equal to 21, as well as the project duration. Note that the latter will therefore increase by 1 even if the duration of 4 is decreased by the same amount.

Example 2. Let us consider again the project network of Figure 5, where $n = 6$.

The time analysis on the standardized network shown in Figure 6 reveals that we need to correct the original (and the standardized) network adding the GPR $SS_{13}^{min}(0)$, because in the standardized network the length of the longest path from dummy node 1 to the node representing critical activity 3 has length equal to 0 and the activity has only ingoing critical precedences of type XF (see Proposition 4.1).

From the time analysis on the corrected standardized network shown in Figure 7, we get that the critical subnetwork coincides with the network itself. By tracing back the critical activities and the critical arcs on the original GPRs project network, we get that also the critical subnetwork of the original project network coincides with the latter, with all the (real) activities being critical. Figure 11 shows the critical subnetwork, where the number associated to node i represents the duration d_i of the activity represented by that node.

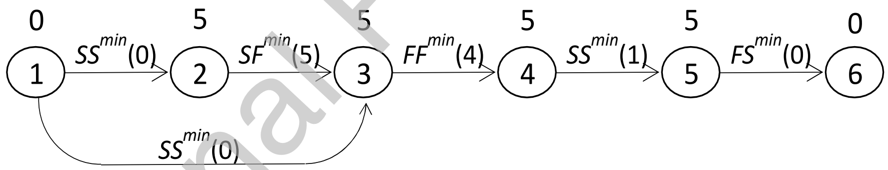

Figure 11: The critical network of Example 2

Node 2 has only one couple of ingoing and outgoing arcs, i.e., $[(1 \xrightarrow{\boldsymbol{SS}} \mathbf{2}), (\mathbf{2} \xrightarrow{\boldsymbol{SF}} 3)]$, which is of type 1. Therefore, activity 2 results to be *start-critical*. Moreover, it is bi-time-feasible because there is no critical elementary cycle containing node 2. Hence, activity 2 turns out to be also *bi-flexible*.

Node 3 has two couples of ingoing and outgoing arcs, i.e., $[(1 \xrightarrow{\boldsymbol{SS}} \mathbf{3}), (\mathbf{3} \xrightarrow{\boldsymbol{FS}} 4)]$ and $[(2 \xrightarrow{\boldsymbol{SF}} \mathbf{3}), (\mathbf{3} \xrightarrow{\boldsymbol{FS}} 4)]$. Arc couple $[(1 \xrightarrow{\boldsymbol{SS}} \mathbf{3}), (\mathbf{3} \xrightarrow{\boldsymbol{FS}} 4)]$ is of type 6 because there exists an elementary path of type $(1 \xrightarrow{\boldsymbol{SS}} \mathbf{3} \xrightarrow{\boldsymbol{FS}} 4 \cdots 5)$, while arc couple $[(2 \xrightarrow{\boldsymbol{SF}} \mathbf{3}), (\mathbf{3} \xrightarrow{\boldsymbol{FS}} 4)]$ is of type 4. Therefore (see Table 2), activity 3 results to be *forward-critical* (and, hence, forward-inflexible). Moreover, it is bi-time-feasible because there is no critical elementary cycle containing node 3. Hence, activity 3 turns out to be also *forward-inflexible*.

Node 4 has only one couple of ingoing and outgoing arcs, i.e., $[(3 \xrightarrow{\boldsymbol{FF}} \mathbf{4}), (\mathbf{4} \xrightarrow{\boldsymbol{SS}} 5)]$, which is of type 3. Therefore, activity 4 results to be *backward-critical* (and, hence, backward-inflexible). Moreover, it is bi-time-feasible because there is no critical elementary cycle containing node 4. Hence, activity 4 turns out to be also *backward-inflexible.*

Finally, node 5 has only one couple of ingoing and outgoing arcs, i.e., $[(4 \underline{\boldsymbol{SS}}\, \mathbf{5}), (\mathbf{5} \underline{\boldsymbol{FS}}\, 6)]$, which is of type 6 because there exists an elementary path of type $(1 \cdots 4 \underline{\boldsymbol{SS}}\, \mathbf{5} \underline{\boldsymbol{FS}}\, 6)$. Therefore, activity 5 results to be *forward-critical* (and, hence, forward-inflexible). Moreover, it is bi-time-feasible because there is no critical elementary cycle containing node 5. Hence, activity 5 turns out to be also *forward-inflexible.*

Example 3. Let us consider again the project network of Figure 9, where $n = 6$.

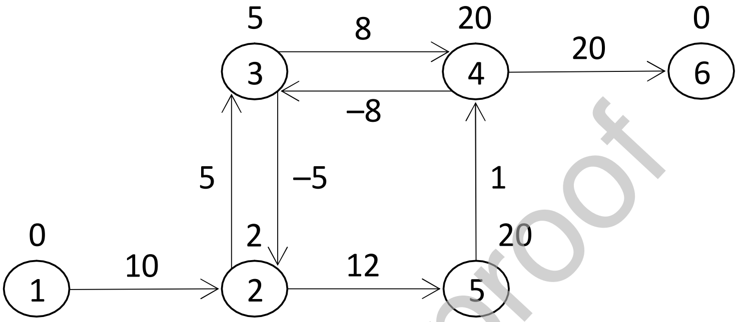

Figure 12: The standardized network of Example 3

Also in this case, the critical subnetwork is the network itself as revealed by the time analysis on the standardized network shown in Figure 12, where the number associated to node i represents the duration d_i of the corresponding activity.

Referring to the critical subnetwork (see Figure 9), we note that node 2 has two couples of ingoing and outgoing arcs to be considered, i.e., $[(1 \underline{\boldsymbol{SS}}\, \mathbf{2}), (\mathbf{2} \underline{\boldsymbol{SF}}\, 3)]$ and $[(1 \underline{\boldsymbol{SS}}\, \mathbf{2}), (\mathbf{2} \underline{\boldsymbol{FS}}\, 5)]$. In fact, even though node 2 has two ingoing arcs, arc $(3, 2)$ should not be considered when looking at couples of ingoing and outgoing arcs of node 2 because there is no elementary path of type $(1 \cdots 3 — 2)$. Arc couple $[(1 \underline{\boldsymbol{SS}}\, \mathbf{2}), (\mathbf{2} \underline{\boldsymbol{SF}}\, 3)]$ is of type 1, while arc couple $[(1 \underline{\boldsymbol{SS}}\, \mathbf{2}), (\mathbf{2} \underline{\boldsymbol{FS}}\, 5)]$ is of type 6 because there exists an elementary path of type $(1 \underline{\boldsymbol{SS}}\, \mathbf{2} \underline{\boldsymbol{FS}}\, 5 \cdots 6)$. Therefore (see Table 2), activity 2 results to be *start-&-forward-critical* (and, hence, forward-inflexible). Moreover, on the basis of the two critical elementary cycles $(\mathbf{2} \underline{\boldsymbol{SF}}\, 3 \underline{\boldsymbol{SS}}\, \mathbf{2})$ and $(\mathbf{2} \underline{\boldsymbol{FS}}\, 5 — 4 — 3 \underline{\boldsymbol{SS}}\, \mathbf{2})$, with the length of the former being independent from the duration of acitvity 2 and the length of the latter that increases by lengthening the activity, it results that activity 2 is forward-time-infeasible. Hence, activity 2 turns out to be also *forward-inflexible.*

Node 3 has four couples of ingoing and outgoing arcs, i.e., $[(2 \underline{\boldsymbol{SF}}\, \mathbf{3}), (\mathbf{3} \underline{\boldsymbol{SS}}\, 2)]$, $[(2 \underline{\boldsymbol{SF}}\, \mathbf{3}), (\mathbf{3} \underline{\boldsymbol{FS}}\, 4)]$, $[(4 \underline{\boldsymbol{SF}}\, \mathbf{3}), (\mathbf{3} \underline{\boldsymbol{SS}}\, 2)]$, and $[(4 \underline{\boldsymbol{SF}}\, \mathbf{3}), (\mathbf{3} \underline{\boldsymbol{FS}}\, 4)]$; the first couple of arcs is of type 2 since they form a cycle and, hence, there is no critical elementary path traversing them; the second and fourth ones are of type 4; finally, the third one is of type 2 since there is no critical elementary path of type $(1 \cdots 4 \underline{\boldsymbol{SF}}\, \mathbf{3} \underline{\boldsymbol{SS}}\, 2 \cdots 6)$. Therefore, activity 3 results to be *start-&-finish-critical.* However, on the basis of the critical elementary cycles

$(\mathbf{3} \xrightarrow{SS} 2 \xrightarrow{\boldsymbol{SF}} \mathbf{3})$ and $(\mathbf{3} \xrightarrow{SS} 2 — 5 — 4 \xrightarrow{SF} \mathbf{3})$, whose lengths increase over shortening activity 3, it results to be backward-time-infeasible; on the contrary, the other remaining critical elementary cycle $(\mathbf{3} \xrightarrow{FS} 4 \xrightarrow{\boldsymbol{SF}} \mathbf{3})$ does not add further time-infeasibility for activity 3 since its length does not depend on the activity duration. Hence, activity 3 turns out to be also *backward-inflexible*.

Node 4 has two couples of ingoing and outgoing arcs to be considered, i.e., $[(3 \xrightarrow{\boldsymbol{FS}} \mathbf{4}), (\mathbf{4} \xrightarrow{\boldsymbol{FS}} 6)]$ and $[(5 \xrightarrow{\boldsymbol{SS}} \mathbf{4}), (\mathbf{4} \xrightarrow{FS} 6)]$. Note that outgoing arc $(4, 3)$ should not be considered when looking at couples of ingoing and outgoing arcs of node 4 because there is no elementary path of type $(4 — 3 \cdots 6)$. Both the two arc couples to be considered are of type 6 because there exist an elementary path of type $(1 \cdots 3 \xrightarrow{\boldsymbol{FS}} \mathbf{4} \xrightarrow{\boldsymbol{FS}} 6)$ and an elementary path of type $(1 \cdots 5 \xrightarrow{SS} \mathbf{4} \xrightarrow{\boldsymbol{FS}} 6)$. Therefore, activity 4 results to be *forward-critical* (and, hence, forward-inflexible). Moreover, on the basis of the two critical elementary cycles $(\mathbf{4} \xrightarrow{SF} 3 \xrightarrow{FS} \mathbf{4})$ and $(\mathbf{4} \xrightarrow{SF} 3 — 2 — 5 \xrightarrow{SS} \mathbf{4})$ whose lengths do not depend on the duration of activity 4, it results that the latter is bi-time-feasible. Hence, activity 4 turns out to be also *forward-inflexible*.

Finally, node 5 has only one couple of ingoing and outgoing arcs, that is $[(2 \xrightarrow{\boldsymbol{FS}} \mathbf{5}), (\mathbf{5} \xrightarrow{SS} 4)]$ which is of type 1. Therefore, activity 5 results to be *start-critical*. Moreover, it is bi-time-feasible because the unique elementary cycle involving node 5 is $(\mathbf{5} \xrightarrow{SS} 4 — 3 — 2 \xrightarrow{FS} \mathbf{5})$ and its length that does not depend on the duration of activity 5. Hence, activity 5 turns out to be also *bi-flexible*.

6 Conclusions

In this work, we showed that the definitions of activity criticalities and the method proposed by De Reyck (1998) for their identification fail in the general context of project networks with GPRs. Analog issues and limitations are also present in the method of Elmaghraby and Kamburowski (1992). To fix these shortcomings, new definitions of the criticality types and a new representation of the latter on GPRs networks were proposed. In particular, we provided a new method for analyzing, without ambiguities, activity criticalities and flexibilities, based on the analysis of the critical paths of the AON project network representation. The correctness of this method has also been shown by means of some examples.

References

[1] Bartusch, M., Möhring R.H., Radermacher F.J., Scheduling project networks with resource constraints and time windows, *Annals of Operations Research*, 16(1) (1988) 201–240.

[2] Bellman, R., On a routing problem, *Quarterly of Applied Mathematics*, 16 (1958) 87–90.

[3] Crandall, K.C., Project planning with precedence lead/lag factors, *Project Management Quarterly*, 4 (1973) 18–27.

[4] De Reyck, B., Scheduling Project with Generalized Precedence Constraints - Exacts and Heuristics Approaches, Ph.D. Thesis, Department of Applied Economics, Katholiecke Universiteie Leuven, Leuven, 1998.

[5] Demeulemeester, E., Herroelen, W., *Project scheduling: a research handbook*, Kluwer, Boston, 2002.

[6] Elmaghraby, S.E., *Activity Networks: Project Planning and Control by Networks*, John Wiley & Sons, New York, 1977.

[7] Elmaghraby, S.E., Kamburowski, J., The analysis of activity networks under generalized precedence relations, *Management Science*, 38(9) (1992) 1245–1263.

[8] Ford, L.R.Jr., Fulkerson, D.R., *Flows in Networks*, Princeton University Press, 1962.

[9] Hajdu, M., *Network Scheduling Techniques for Construction Project Management*, Kluwer Academic, 1997.

[10] Hajdu, M., Continuous precedence relations for better modelling overlapping activities, *Proc. of the 2015 4th Creative Construction Conference (CCC 2015)*, *Procedia Engineering*, 123 (2015) 2016–2023.

[11] Hajdu, M., Skibniewski, M.J., Vanhoucke, M., Horvath, A., Brilakis, I., How many types of critical activities exist? A conjecture in need of proof, *Proc. of the 2016 5th Creative Construction Conference (CCC 2016)*, 25–28 June 2016, *Procedia Engineering*, 164 (2016) 3–11.

[12] Kerbosch, J.A.G.M., Shell, H., "Network planning by the extended METRA potential method", Report KS-1.1, Department of Industrial Engineering, Group Operational Research, University of Technology Eindhoven, The Netherlands, 1975.

[13] Kong, F., Li, J., New network model of the project scheduling problem with hybrid precedence relations and maximum activity durations, *Proc. of the 2020 4th International Seminar on Education, Management and Social Sciences (ISEMSS 2020)*, *Advances in Social Science, Education and Humanities Research*, 466 (2020) 1086–1089.

[14] Moder, J.J, Phillips, C.R., Davis, E.W., *Project Management with CPM, PERT and Precedence Diagramming*, Van NostrandReinhold, 1983.

[15] Moder, J.J, Crandall, K.C., Precedence diagramming: time computations, anomalies and remedies, *Project Management: Methods and Studies*, (1985) pp. 95110.

[16] Moore, E.F., The shortest path through a maze, *Proc. Internat. Sympos. Switching Theory 1957*, Part II, Harvard Univ. Press, (1959) pp. 285–292.

[17] Nisar, J., and Halim, S., The effect of critical activity on critical path and project duration in Precedence Diagram Method, *International Scholarly and Scientific Research & Innovation*, 12(9) (2018) 837–843.

[18] Roy, G.B., Graphes et ordonnancement, *Revue Franais de Recherché Opérationnelle*, 25(6) (1962) 323–333.

[19] Valls, V., Lino, P., Criticality analysis in activity-on-node networks with minimal time lags, *Annals of Operations Research*, 102 (2001) 17-37.

[20] Wiest, J.D., Precedence diagramming methods: some unusual characteristics and their implications for project managers, *Journal of Operations Management*, 1(3) (1981) 121–130.