CSCE Annual Conference
Growing with youth – Croître avec les jeunes

Laval (Greater Montreal)
June 12 - 15, 2019

CONCURRENT DELAYS: COMPARISON AMONG FORENSIC ANALYSIS RECOMMENDED PRACTICES

Moneer Bhih [1,2,3], and Tarek Hegazy[1]
[1] University of Waterloo, Canada.
[2] University of Benghazi, Libya. (Formerly)
[3] mbhih@uwaterloo.ca

Abstract: Construction projects usually operate in dynamic multi-party constrained environments that frequently cause schedule delays, cost overruns, and disputes among parties. To mitigate construction disputes and fairly apportion delay responsibility among parties, forensic delay analysis becomes necessary, particularly to assess the difficult case of concurrent delays. Delay analysis techniques, however, vary in their use of a single versus multiple windows (schedule periods that represent the project) of analysis. Moreover, the adoption of either the literal or functional views on concurrent delays, may affect the analysis results. Literal concurrency is a strict perspective when the delay events are happening simultaneously, while functional concurrency represents the more flexible perspective that delays are concurrent even if the causing events occur in different, but close, time periods. This paper, thus, examines the varying assumptions among delay analysis techniques and recommended practices, and shows their varying delay analysis results. Single window analysis is found to be unable to account for critical path fluctuations, nor identify true concurrency. The paper clarifies the misconceptions in the use of the various delay analysis methods, compares among the common forensic analysis standards, and proposes the use of event chronology in a multi-window analysis to improve concurrency assessment. An example application is used to demonstrate the benefits of the proposed approach in providing more accurate assessment of concurrency.

1 INTRODUCTION

Delays are inevitable on the majority of construction projects. Delay events (activity-level interruptions) can be caused by one or more project party, in different times during the construction period. Having two or more simultaneous critical events overlapped, in whole or in part, lead to a combined effect on the project completion date called concurrent delays. Delay events occurring at the same time and either of them alone would cause a project delay (i.e., both are critical delays) are considered for concurrancy (Stumpf 2000). Apportioning responsibility of concurrent delays is a challenging task since it involves the effect of multiple events (Ibbs, et. al, 2011). Delay analysts must take into account concurrent delays to allocate project delays equitably and to avoid multiple counting of delay days (Lee 2007). In the past decades, several efforts addressed delay analysis issues including concurrent delays. Recently, several professional societies issued guidance documents for forensic schedule analysis. However, the assessment of concurrency remains one of the most confusing forensic analysis issues. This paper thus discusses the technical issues of concurrency assessment using CPM-based analysis techniques. It starts by introducing the three forensic analysis guidelines followed by a description of the controversial definitions and assumptions. The paper then discusses the misconceptions in the analysis and presents guidelines to overcome these misconceptions.

2 RECOMMENDED PRACTICES AND STANDARDS FOR FORENSIC SCHEDULE ANALYSIS

Several professional bodies have shown their interest in forensic schedule analysis, in the last few decades, by establishing protocols, recommended practices, and standards, aiming at promoting clearer contract conditions regarding float ownership, concurrent delays, and methods to determine compensation and time extension. Currently, there are three main guidance documents for schedule delay analysis in the world (Livengood 2017); briefly introduced as follows:

Delay and Disruption Protocol, Society of Construction Law (SCL), UK: The Society of Construction Law was the first to publish the delay and disruption protocol in October 2002 (SCL-DDP 2002). In February 2017, the 2nd edition of the Protocol was published. The protocol helps in determining the extensions of time and the compensations for delay and disruption. It distinguishes between delay and disruption. Delay signifies that activities taking longer than planned and causing delay to the completion of the project. Disruption, on the other hand, is concerned with disturbance, hindrance or interruption to the contractor’s working methods causing lower productivity, regardless of whether the activity is on the critical path (SCL-DDP 2017). This document thus is intended to be a balanced document that equally represent the interests of all parties and not a statement of the law (SCL-DDP 2017). The protocol concepts are not geared towards the use of any specific method of delay analysis.

Recommended Practice 29R-03, AACEI, USA: The Association of the Advancement of Cost Engineering International (“AACEI”) published its Recommended Practice No. 29R-03 (RP) in June 2007. This RP was revised twice, with the latest revision in April 2011 (AACE RP 29R-03 2011). The objective is to “provide a unifying technical reference for the forensic application of critical path method (CPM) of scheduling and to reduce the degree of subjectivity involved in the current state of the art. The RP is an advisory document to be used with professional judgment based on working experience and knowledge”. The RP recommends best practices for forensic schedule analysis by introducing and implementing nine methods of analysis. It also discusses the factors affecting the selection of a particular method.

Schedule Delay Analysis (ANSI/ASCE/CI 67-17), ASCE, USA: In 2017, the American Society of Civil Engineers ASCE published their new proposed standard entitled “Schedule Delay Analysis” under the designation (ANSI/ASCE/CI 67-17) in accordance with the American National Standards Institute ANSI. The document introduces 35 guidelines of schedule delay analysis principles which enable apportioning of responsibility for delay, to project milestones and the project completion date, and calculating delay or liquidated damages. The guidelines are categorized in eight categories: critical path, float, early completion, chronology of delay, concurrent delay, responsibility for delay, changing schedules after the fact, and acceleration (ASCE 67-17 2017). The standard can be used in consistency with practitioner’s experience, along with project facts and contract requirements. Its guidelines do not emphasize any specific delay analysis method, rather are applicable in conjunction with any methodology.

3 VARYING DEFINITIONS OF CONCURRENT DELAYS

Although the above three professional guidance documents have defined concurrent delays; those definitions reflect the differing viewpoints that originated from the inconsistent application of concurrent delay in practice (Livengood 2017).

(SCL-DDP 2017) defines true concurrent delays as "the occurrence of two or more delay events at the same time, one an Employer Risk Event, the other a Contractor Risk Event, and the effects of which are felt at the same time". It requires that both events be critical and affect the critical path of the project. Based on this definition, concurrent delays exist only if both causing events are happening at the same time and their effects (project delays) are felt at the same time.

(AACE RP 29R-03 2011) defines the concurrent delays as "two or more delays that take place or overlap during the same period, either of which occurring alone would have affected the ultimate completion date". AACE recommended practice also defines true concurrency as defined in (SCL-DDP 2017), however it has a more flexible position regarding the time period between the delay-causing events, as opposed to the literal viewpoint which requires true concurrency. As such, AACE recommended practice accepts both the literal and functional concurrency definitions, reflecting the American practice of concurrent delays.

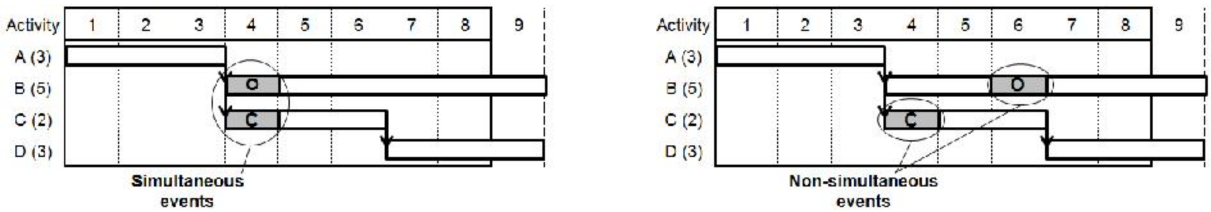
(ASCE 67-17 2017) defines concurrent delays as "delay to the project critical path caused concurrently by multiple events not exclusively within the control of one party". In another definition, the standard described the concurrent delay situation as "a situation where two or more critical delays are occurring at the same time". The commentary on this definition explained that both events should not be starting or ending at the same time. The standard has no clear position regarding the time interval between delay events, but it could be understood that it requires both events literally "occurring at the same time".

3.1 Literal Concurrency versus Functional Concurrency

The AACE recommended practice highlighted two theories of concurrency; literal versus functional concurrency. The literal theory is a strict perspective when delay events are happening at the same time, while the functional theory represents a more flexible perspective that delays are concurrent even if the causing events occur in different, but close, time periods. Other researchers have also contributed to the debate about concurrency. Dale and D'Onofrio (2010) presented a definition of concurrent delays that combines the literal and functional viewpoints, as follows: "concurrent delay generally refers to both delays occurring at the same time as well as delays that occur at different times, but with a common effect". Later, Dale and D'Onofrio (2017) concluded that depending on the schedule accuracy, critical events that start less than a week apart can be considered to cause concurrent project delays. However, the debate among practitioners still exists about the time interval between events of any candidate concurrent delays. This debate is a result of the conflict of interest among project parties. Concurrency can be used to excuse one party from compensation or liquidated damages. As such, the party trying to prove concurrency prefers a wider time range to consider concurrency while the other party prefers a narrower range to defend against it.

3.2 True Concurrency and Concurrent Effects

The (SCL-DDP 2017) protocol adopts the literal concurrency viewpoint and calls it "true concurrency", and calls any two or more events that happen at different times but its effects are felt at the same time as "concurrent effects" . To explain the difference between the two situations, the simple progress scenarios in Figure 1 is used. In Figure 1a, there are two simultaneous delay events happening on the same day (day4), one of them is owner event and the other is contractor event. Both events are on parallel critical paths so their effects (project delays) happen concurrently on day 9, in this case the delays are concurrent. On the other hand, in the scenario given in Figure 1b the contractor event is on day 4 while the owner event is on day 6. Both of the events still critical so their effects (project delay) are felt on day 9; since the causing events are not simultaneous, the situation is considered as concurrent effects and the delay is attributed to only the contractor.



a) True concurrent delays b) Concurrent effects

Figure 1: True concurrency versus concurrent effects

3.3 Pacing Delay and Offsetting Delay

Pacing delay and offsetting delay are project delays that have some flavor of concurrent delays (Dale and D'Onofrio 2017). Pacing delay is the situation when the owner causes delay to the critical path (parent delay), and the contractor opt to relax its performance of work to the extent that it does not affect project completion (pacing delay). "Why hurry up and end up waiting" is normally used to justify pacing. Concurrent delays by definition are involuntary delays by both parties(AACE RP 29R-03 2011), however, pacing delay is voluntary (Livengood 2017). In the pacing situation, the contractor has to prove pacing to defend against concurrency by demonstrating the existence of parent delay; the ability to continue with the normal pace; and an evidence of intent to pace (AACE RP 29R-03 2011). Pacing delay, however, was not discussed by (SCL-DDP 2017) and (ASCE 67-17 2017) guidelines.

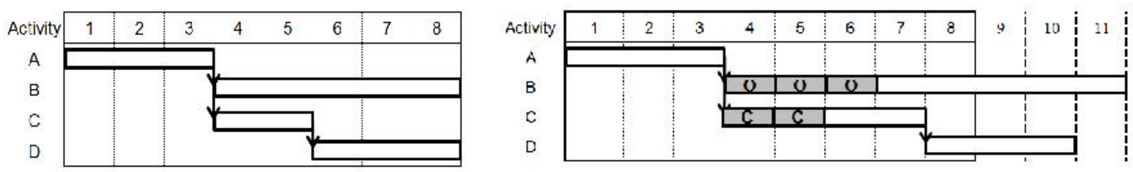
The offsetting delays is defined by (ASCE 67-17 2017) as "a delay that may occur when a contractor is behind schedule and the owner later causes a delay to the contract completion date". Based on this definition, when the project completion date is passed and the contractor is behind schedule, all of the remaining activities are critical and have negative float values. In such situation, any owner-caused delay may offset the contractor liquidated damages, even if it doesn't cause further delay to the project. The concept created a big debate among the practitioners since it was introduced; it requires the owner to grant the contractor time extension for a non-critical delay and to waive its liquidated damages. At the same time, it does not apply equally for the owner and contractor (Nagata 2018). The offsetting delay is considered as a sort of concurrent delays as it is caused by two different parties (Contractor and Owner) and the effects of both events are concurrent, moreover it follows the same compensation rule of concurrent delays.

4 MTHODOLOGY

This paper clarifies the misconception in delay analysis practice by first explaining the source of problem, then suggesting ways to resolve this misconception and arrive at accurate assessment of delay concurrency. The paper deals with two main problems, namely: (1) resolving the misconceptions in delay analysis practice; and (2) Improving the assessment of concurrent delays. These aspects are discussed in the following two sections, respectively.

5 RESOLVING MISCONCEPTIONS IN DELAY ANALYSIS PRACTICE

Some delay analysis techniques involve misconceptions and challenges in assessing concurrent delays, particularly when the analysis is performed from different parties' viewpoints. The common application of the But-For (collapsed as-built) and What-If (impacted as-planned) techniques involves such misconceptions, which renders both of them unable to identify concurrent delays, as highlighted in literature (SCL-DDP 2002; Dale and D'Onofrio 2017; El-adaway et al. 2014; Fawzy and El-Adaway 2012). The same issue is shown in the analysis results of case studies presented by (Dale and Onofrio 2010; Braimah 2013; Mbabazi et al. 2005) in comparison with other techniques. The (AACE RP 29R-03 2011) methods of implementation MIP 3.8 and MIP 3.6 are similar to But-For and What-If analysis, respectively, when applied as a single window analysis. However, both implementations allow stepped analysis, i.e., accommodate multiple windows, and thus it descriptively attempt to address some of the shortcomings in the common practice of the But-For and What-If analyses. To explain the misconceptions, the small example of four activities in Figure 2 is used.



a) As-planned schedule b) As-built schedule

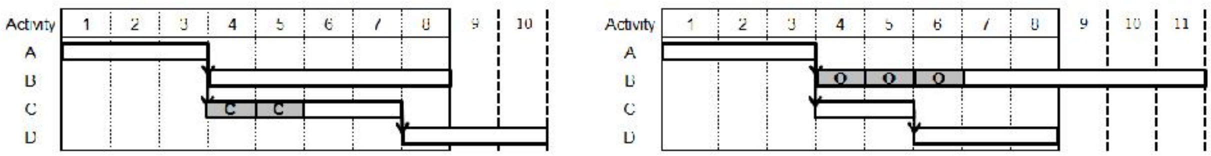
Figure 2: Case study schedules

5.1 Problems with Single-Window Analysis

But-For and What-if techniques consider the whole project period as a single analysis window. In But-For analysis, the delay events are grouped, based on the causing party, and subtracted one at a time from the as-built schedule to investigate its impact on the project completion date. On the other hand, in What-If analysis, the events are added to the as-planned schedule to investigate the same effect. Only But-For analysis issues are shown in this paper due to space limitation.

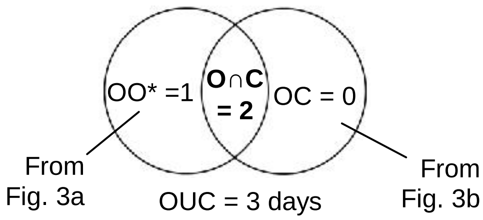
To determine the delay reasonability using the common But-For analysis, let's first adopt one of the project parties' viewpoints. From the owner's perspective, by removing the owner-caused events from the as-built schedule, the schedule of Figure 3a is obtained. The as-built schedule was collapsed 1 day after removing the owner-caused events; so the owner accepts responsibility for only 1 day project delay, and claims that the remaining balance of the project delay (2 days) is attributed to the contractor (OO = 1 day, OC = 2 days, O∩C = 0 days). The same analysis from the contractor's point of view is presented in Figure 3b, which produces different responsibility of 3 days due to owner and zero days due to contractor (OO = 3 day, OC = 0 days, O∩C = 0 days). This disparity in results is because of the analysis misconception involved in the results interpretation of the common But-For implantation.

To easily explain the common misinterpretation of results, the Venn representation proposed by (Mbabazi et al. 2005) is utilized. Figure 4 shows the graphic representation of project delay caused by "only owner OO", "only contractor OC", and "both concurrently O∩C". OO responsibility is obtained from Figure 3a by removing owner-caused delays (1 day); the misinterpretation is now clear that the owner assumes the rest of the delay is due to only contractor OC, while it's in fact a combination of OC and O∩C. Similarly, OC reasonability is obtained from Figure 3b by removing contractor-caused delays (0 days). Since both parties ignore the O∩C component, it is possible to calculate it easily from the results (i.e., total Venn area of OUC = OO + OC + O∩C). In the present case, 3 = 1 + 0 + O∩C, thus O∩C = 2 days, as indicated on Figure 4. Thus, in the common interpretation of But-For results, each party ignores its contribution to concurrent O∩C delays. The correct responsibility in this case is (OO = 1 day, OC = 0 days, O∩C = 2 days). For this reason the technique was criticized in literature for the inability to identify concurrent delays. The summary of the common But-For analysis and the correct analysis are presented in Table 1.



a) Collapsed as-built (without owner events)   b) Collapsed as-built (without contractor events)

Figure 3: Case study collapsed-as-built schedules



* OO = Only Owner; OC = Only Contractor; O∩C = Both Concurrently

Figure 4: But-For analysis correct interpretation of results

Table 1: Delay responsibility using But-For Analysis

Point of view	Responsibility		
	Owner (days)	Contractor (days)	Both concurrently (days)
Owner	1	2	0
Contractor	3	0	0
Correct Results	1	0	2

5.2 Single window Implementations of AACE MIP 3.8 and MIP 3.6

Considering the similarities between (AACE RP 29R-03 2011) single window implementation of MIP 3.8 and MIP 3.6 on one hand and the common application of But-For and What-IF analysis on the other hand, the AACE methods have some improvements to address the inability of the traditional techniques to identify concurrent delays. MIP 3.8 in its minimum implementation requires concurrency analysis. The method identifies that after subtracting any sub-set of delay events from the as-built schedule, the collapsed as-built schedule still contains the impact of the delay events left including any potential concurrent impact of those events. In other words, the method recognizes that the variance between the as-built and the collapsed as-built schedule is the sole responsibility of the removed events and the remaining balance of the delay is due to the remaining events including any concurrent delays. In contrast, concurrency cannot be evaluated using the minimum implementation of AACE method MIP 3.6 (AACE RP 29R-03 2011). The recommended practice listed several enhancements to the minimum implementation that would help in evaluating the concurrency, out of which is to compare the impacted as-planned schedule with the as-built schedule. This could be interpreted as an attempt to assess the sole responsibility of the other party as the variance between the impacted as planned and the as-built schedule if only two parties' events are involved. Both methods of implementation have no clear step by step analysis procedures to quantify concurrent delays.

6 IMPROVING CONCURRENCY ASSESSMENT CONSIDERING EVENT CHRONOLOGY

Delay events chronology is the order in which delays have evolved over the course of project. Using the chronological order of events helps to segregate its effect on the project completion date correctly and allow allocating it among project parties. (ASCE 67-17 2017) and (AACE RP 29R-03 2011) requires that delays should be evaluated chronologically and cumulatively in the same order in which they occurred. In other words, project delays should be assessed by conducting delay analysis on properly updated schedule that consider the evolution of events. Most delay analysis techniques investigate the impact of the events related to each party separately within the same analysis window to allocate project delays and to assess concurrency, if any. Having several delay events caused by different parties, on the same critical path or on different paths, may cause concurrency miss-assessment if the effect of each party events was investigated in an order differing from the actual progress. This kind of practice affects not only the concurrency assessment, but could lead to change in events criticality to manipulate the responsibility. Considering event chronology in the concurrency assessment, however, needs to consider either the literal or the functional concepts of concurrency.

6.1 Guidelines for Single Window Analysis

The single window analysis techniques (But-For and What-If) are able to identify concurrent delays if the correct interpretation of results, as explained in the previous section, is used. However, it is still lacking the ability to distinguish between the true concurrency and the concurrent effects; i.e., any two critical delays on parallel critical paths of the as-built schedule, caused by different parties, will be assigned as concurrent delays regardless of the timing of the causing events. This is because of the single window implementation which does not respect the event chronology by nature. However, such non-simultaneous events can cause concurrent delays under the functional concurrency if the events are close enough in time. To explain this issue using the But-For analysis, the as-built schedule of the case study has been slightly altered as shown in progress scenario 2 in Figure 5. The difference in this scenario is that the owner and contractor events are not occurring simultaneously.

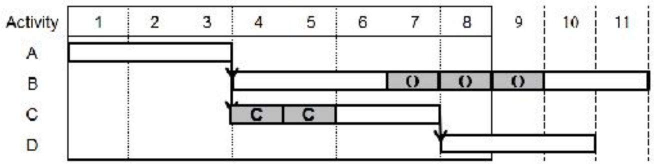


Figure 5: As-built schedule (progress scenario 2)

Results using correct But-For Analysis: as explained earlier in this paper, OO responsibility is obtained by removing owner-caused delays which result in reducing the as-built duration by (1 day). Similarly, OC reasonability is obtained by removing contractor-caused delays with no reduction in the as-built duration (0 days). Total project delay is 3 days, thus O∩C = 2 days. The delay responsibility is identical to those of original progress scenario (OO = 1 day, O∩C = 2 days), despite the difference in the timing of events.

Results considering event chronology: The as-planned schedule of the project has two critical paths (AB and ACD) (Figure 2a). If the events was considered separately, as commonly conducted in the But-For and What-If analysis, the 3 days owner events on AB critical path caused 3 days extension of project duration (days 9,10 and 11), while the 2 days contractor event on ACD critical path caused 2 days extension (days 9 and 10). The delay on days 9 and 10 caused by the owner and contractor are mistakenly captured by the analysis as concurrent delays as they are felt concurrently. Nevertheless, the contractor delays on days 9 and 10 are caused by events on days 4 and 5, while the owner delays on days 9 and 10 are caused by events on days 7 and 8; as the causing events are not simultaneous; thus it is just "concurrent effects". To consider the chronological order of events, the contractor events on days 4 and 5 caused an extension of 2 days to the project duration (days 9 and 10), this extension created 2 days float for the other critical path (AB). The analysis now consider the subsequent 3 days owner events, which will first consume the 2 days float created earlier by the contractor delay and then will cause an extra 1 day project delay. Based on this analysis the correct responsibility considering event chronology is (OO = 1 day, OC = 2 days, O∩C = 0 days).

Results under literal concurrency: The 2 days of O∩C concurrent delay obtained by the But-For analysis is clearly wrong where no simultaneity exists between the two events as shown on the as-built schedule. But-For technique, in this case, identified concurrent effect as true concurrent delays; single window implementation makes the technique insensitive to the timing of events. A known drawback of the traditional single window analysis techniques is its inability to respect the chronological order of the events (Dale and D'Onofrio 2017). To overcome this drawback, analysis considering event chronology as explained in the previous paragraph needs to be conducted (in this case the correct responsibility is OO = 1 day, OC = 2 days, O∩C = 0 days). A more accurate delay analysis method becomes necessary for large and complex projects as the analysis considering event chronology is not a simple task for such projects.

Results under functional concurrency: If the analyst assumes that the functional concept is to be used, then the acceptable time interval between the events should be assumed for the concurrency assessment. In this case, from the as-built schedule in Figure 5, the difference between the start of contractor event and the start of owner event is 3 days. Based on the schedule update accuracy and the uncertainty of project schedules, such events might be close enough to be considered simultaneous events. If this argument is justified by enough evidences, then the analysis results obtained by the correct But-For analysis are correct (in this case the responsibility is OO = 1 day, OC = 0 days, O∩C = 2 days). If the events are not close enough; the correct responsibility in such case will be OO = 1 day, OC = 2 days, O∩C = 0 days. Summary of analysis results is shown in Table 2 for comparison. For larger projects with multiple critical paths and delay events, the use of multiple windows analysis is necessary, as the analysis considering event chronology using single window becomes complex. The multiple analysis windows in such situations should be arranged considering the project characteristics (critical path(s) and delay events) and the adopted concurrency concept, as explained in the next subsection.

Table 2: Delay analysis results of scenario 2

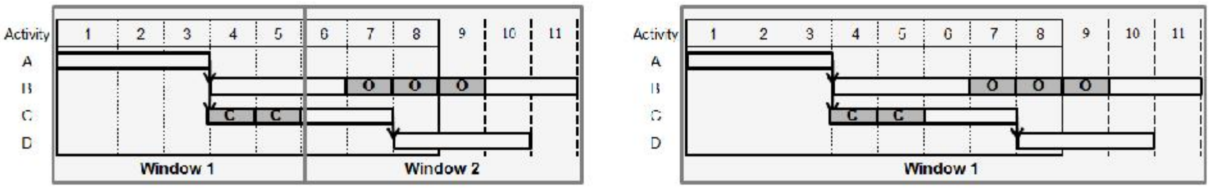
Analysis type	Responsibility (days)		
	Owner	Contractor	Concurrent
Correct single window analysis (But-For)	1	0	2
Analysis considering event chronology	1	2	0
Analysis under literal concurrency	1	2	0
Analysis under functional concurrency	1	0	2

6.2 Guidelines for Multiple Windows Analysis

In the additive techniques, such as windows analysis, the events are inserted into as-planned schedule from the project start towards its end on windows intervals in sequence, while in the subtractive techniques, such as multiple windows But-For analysis, the events are removed from the as-built schedule starting from the project end towards its start (AACE RP 29R-03 2011). Both types of techniques respects the event chronology only on a window bases from one to another, (i.e., the chronology is violated within the analysis windows), unless the events are investigated individually and sequentially within each window. Window size has an impact on the analysis results, the smaller the window sizes the more accurate analysis results (higher analysis resolution). The smaller window sizes provide better respect of event chronology. Analysis window size should be selected carefully to enable the technique to distinguish between true concurrency and concurrent effects. This also depends on the concurrency definition adopted in the analysis; literal or functional.

Analysis windows arrangement under literal concurrency: Since chronology is not respected within the analysis windows, any two project delays on parallel critical paths within the same window will be seen as concurrent delays even if the causing events are not literally simultaneous. To avoid such wrong assessment the window sizes need to be selected so that critical party's events are on separate windows, unless the events are simultaneous. For example, in progress scenario 2 of Figure 6, using two windows of analysis (from day 1 to day 5) and (from day 6 to day 11) (Figure 6a), will ensure that the delay events of the two parties are on separate windows. This window arrangement ensures correct concurrency assessment for both the additive and subtractive techniques under the literal concept. Daily windows analysis proposed by (Hegazy and Zhang 2005) has the highest analysis resolution which can track critical path fluctuation on a daily basses, and fully respect the event chronology. Thus daily windows technique provides consistent results under the literal concept.

Analysis windows arrangement under functional concurrency: Under the functional concept, critical events of different parties close enough in time are treated as simultaneous and its effects are allocated as concurrent delays. To obtain the correct results under functional concept, the window should be selected in such a manner that it contains the events that fall within the acceptable time range from each other. For example, in the progress scenario 2 of Figure 6, if both the owner and contractor events are considered close enough to be treated as simultaneous, then the window size should be selected so that both of them are in the same window. In this case, only the single window analysis (one window from day 1 to day 11 (Figure 6b)) will produce the correct results using the additive or subtractive techniques. Also in this case, if a smaller window size is used so that both events are in separate windows, the analysis will not account for concurrency among both delays, and will produce wrong results under the functional concept.



a) Considering literal concurrency b) Considering functional concurrency

Figure 6: Window arrangement for concurrency assessment – progress scenario 2

7 DISCUSSION AND ONGOING WORK

Acceleration events can cause multiple-day impact on the project completion date. The same impact could also be caused by delay events that shift an activity to a time period that has lower productivity rate or less available resources. The allocation of responsibility in the case of two or more simultaneous events that have multiple days of project delays/accelerations can be complex, particularly when acceleration is involved. Even with the most accurate delay analysis techniques such as daily windows analysis (Hegazy and Zhang 2005) calculation errors can happen. In such case, the remaining part of the as-planned schedule will be changed, leading to a possible change in the parallel paths, which requires consideration of both critical and near-critical paths in the analysis. Under the functional concept the analysis becomes even more complex as events on different days are candidate for concurrency; thus, events on near critical paths are also considered for concurrency. The use of a more detailed analysis becomes necessary to untangle the intertwined effects of events. Ongoing research work is being conducted by the authors using set theorem to allocate responsibility of events that have multiple days of delays/accelerations.

8 CONCLUSION

This paper provided guidelines towards a more accurate assessment of concurrent delays using several single and multiple-window analysis techniques. Concurrent delay definitions by the professional guides were also introduced and discussed to show the different viewpoints and application assumptions. Some of the single window analysis techniques in practice are unable to identify concurrent delays due to the misleading interpretation of results involved in its traditional analysis procedures. Moreover, disregarding delay-event chronology in delay analysis is a major cause of inaccurate concurrency assessment as event chronology is partially or totally violated by the analysis techniques, such as the common But-For method. In this paper, the miss-interpretation of results in the common application of single window delay analysis (e.g., But-For Analysis) was explained. Guidelines for considering event chronology to improve concurrency assessment were introduced using both single and multiple-windows techniques, under both literal and functional concurrency concepts. Using multiple window analysis considering the explained guidelines presents a more accurate method for a better assessment of concurrent delays.

References

AACE RP 29R-03. 2011. *Forensic Schedule Analysis*. American Association of Cost Engineers.

ASCE 67-17. 2017. *Schedule Delay Analysis*. *American Society of Civil Engineers*.

Braimah, Nuhu. 2013. "Construction Delay Analysis Techniques—A Review of Application Issues and Improvement Needs." *Buildings* 3 (3): 506–31. doi:10.3390/buildings3030506.

Dale, W. Stephen, and Robert M. D'Onofrio. 2010. "Reconciling Concurrency in Schedule Delay and Constructive Acceleration." *Public Contract Law Journal*, no. 39: 161–230.

Dale, W. Stephen, and Robert M. D'Onofrio. 2017. *Construction Schedule Delays*. 2017 ed. THOMSON REUTERS.

El-adaway, Islam H., Salwa a. Fawzy, Robert Bingham, Patrick Clark, and Taylor Tidwell. 2014. “Different Delay Analysis Techniques Applied to the American Institute of Architects A201-2007 Standard Form of Contract.” *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction* 6 (3): 2514001. doi:10.1061/(ASCE)LA.1943-4170.0000145.

Fawzy, Salwa a., and Islam H. El-Adaway. 2012. “Contract Administration Guidelines for Effectively and Efficiently Applying Different Delay Analysis Techniques under World Bank Funded Projects.” *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction* 5 (February): 120915032631000. doi:10.1061/(ASCE)LA.1943-4170.0000104.

Hegazy, Tarek, and Kehui Zhang. 2005. “Daily Windows Delay Analysis.” *Journal of Construction Engineering and Management* 131 (5): 505–12. doi:10.1061/(ASCE)0733-9364(2005)131:5(505).

Ibbs, William, Long D. Nguyen, and Lonny Simonian. 2011. “Concurrent Delays and Apportionment of Damages.” *Journal of Construction Engineering and Management* 137 (2): 119–26. doi:10.1061/(ASCE)CO.1943-7862.0000259.

Lee, Jae-Seob. 2007. “Delay Analysis Using Linear Schedule in Construction.” *AACE International Transactions*, 1–6.

Livengood, John. 2017. “Knowns and Unknowns of Concurrent Delay” 9 (3): 1–11. doi:10.1061/(ASCE)LA.1943-4170.0000224.

Mbabazi, Anania, Tarek Hegazy, and Frank Saccomanno. 2005. “Modified But-For Method for Delay Analysis.” *Journal of Construction Engineering and Management* 131 (10): 1142–44. doi:10.1061/(ASCE)0733-9364(2005)131:10(1142).

Nagata, Mark. 2018. “Criticism of the ASCE Schedule Delay Analysis Offsetting Delay Concept,” 1–20.

SCL-DDP. 2002. *Delay and Disruption Protocol*. *Society of Construction Law*.

SCL-DDP. 2017. *Delay and Disruption Protocol*. Society of Construction Law.