Integrated Daily Windows and But-For Analyses for Accurate Schedule Forensics in Construction Projects

by
Moneer Bhih

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Civil Engineering

Waterloo, Ontario, Canada, 2021

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	**NAME:**	Amin Hammad
	TITLE:	Professor at Concordia University, Concordia Institute for Information Engineering (CIISE)
Supervisor	**NAME:**	Tarek Hegazy
	TITLE:	Professor at University of Waterloo, Civil and Environmental Engineering Department
Internal Member	**NAME:**	Carl T. Haas
	TITLE:	Professor at University of Waterloo, Civil and Environmental Engineering Department
	NAME:	Scott Walbridge
	TITLE:	Professor at University of Waterloo, Civil and Environmental Engineering Department
Internal-external Member	**NAME:**	Ramadan El Shatshat
		Professor at University of Waterloo, Electrical and Computer Engineering Department

Author’s Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Construction projects usually operate in dynamic multi-party constrained environments that frequently cause schedule delays and cost overruns and lead to disputes among parties. To mitigate such disputes and fairly apportion delay responsibility among parties, forensic schedule analysis becomes necessary. Each schedule analysis technique has its unique methodology, advantages, and drawbacks, thus, no single method provides a comprehensive schedule analysis tool that suits all viewpoints and progress situations, and produces consistent results in all cases. In addition to its drawbacks, some delay analysis techniques have misconceptions associated with its common application, which render it unable to identify concurrent delays and produce different results when used by different project parties.

This research, thus, introduces an integrated delay analysis framework that aims at enhancing the current state of forensic analysis to overcome the persistent drawbacks of the techniques. The proposed framework targets the most used and accepted techniques, which are But-For and Windows Analyses, aiming at integrating them to benefit from their advantages and complement each other. As the first step, the framework targeted enhancing the traditional But-For analysis to match the industry standards requirements. The introduced But-For enhancements start with clarifying the misleading interpretations of But-For results and introducing an explicit implementation procedure. As a more accurate approach to consider event chronology and baseline updates, a detailed procedure for applying But-For with multiple analysis windows is introduced. Then, the But-For analysis is extended to the general case of three project parties.

Further enhancements to But-For analysis have been introduced by improving the Modified But-For method (MBF). In the literature, the MBF resolved some of the But-For analysis drawbacks, however, its methodology was developed only to allocate schedule delays overlooking common schedule accelerations. In the second step, the MBF methodology has been extended to allocate both delays and accelerations, then, the Multi-Window MBF analysis has been introduced to address its insensitivity to event chronology and critical path(s) fluctuations, and to consider multiple baseline updates. With those enhancements, the But-For analysis was enabled to allocate both concurrent and non-concurrent delays and accelerations with a clear and structured methodology that considers event chronology and baseline updates.

The final step, in this research, introduces an Enhanced Daily Windows Analysis (EDWA) that integrates the ability of the Daily Windows analysis, as the most accurate version of windows analysis, and the ability of the enhanced But-For analysis to analyze concurrent delays and accelerations. The EDWA follows the day-by-day analysis of Daily Windows and apportions delays/accelerations responsibilities using the enhanced But-For methodology. To facilitate the application of EDWA, a macro program was written on MS Project software and case studies were used to prove its ability to resolve the shortcomings of parent techniques. EDWA method contributes to developing a readily usable and comprehensive method for schedule analysis that produces accurate and repeatable results and helps to avoid disputes among project parties.

Acknowledgements

In the Name of Allah, the Most Beneficent, the Most Merciful. First and foremost, I thank Allah (God) for giving me health, patience, and strength, to complete this work.

I wish to express my profound and sincere gratitude to my supervisor Prof. Tarek Hegazy for his continuous support, encouragement, and guidance. You offered me the time and advice I would ever need to conduct my research, and you inspired me with your knowledge, wisdom, and dedication to academic research. May Allah reward you and bless you.

I am very grateful to my home university, University of Benghazi, for giving me the chance to pursue my education journey through the fanatical support provided by the Ministry of Higher Education of Libya.

I would love also to thank my committee members Prof. Amin Hammad, Prof. Carl Haas, Prof. Scott Walbridge, and Prof. Ramadan El Shatshat for their insightful comments.

My thanks to my colleague Dr. Wail Menesi for his support and advice. My thanks goes also to my colleagues in the research group for their friendship, for all the great times, and for providing a friendly environment to work in.

My deep gratitude is upon my small family in Canada for their support and patience, and my big family in Libya for their encouragement and support. Without you all, I couldn't do it

Dedication

To your soles; My Parents

Table of Contents

List of Figures

List of Tables

List of Abbreviations

O	Owner
C	Contractor
N	Third party (neither owner nor contractor)
OO	Only Owner responsibility
OC	Only Contractor responsibility
ON	Only third party responsibility
$O \cap C$	Owner and contractor concurrent responsibility
$O \cap N$	Owner and third party concurrent responsibility
$C \cap N$	Contractor and third party concurrent responsibility
$O \cap C \cap N$	Owner, contractor, and third party concurrent responsibility
BF	But-For
MBF	Modified But-For
EMBF	Enhanced Modified But-For
DWA	Daily Windows Analysis
EDWA	Enhanced Daily Windows Analysis
CP	Critical Path
CPM	Critical Path Method
T_0	As-built duration
T_1	As-planned duration
T_2	As-built without owner events duration
T_4	As-built without contractor events duration
T_3	As-built without third party events duration
T_5	As-built without owner and contractor events duration
T_6	As-built without owner and third party events duration
T_7	As-built without contractor and third party events duration

Chapter 1
Introduction

1.1 Background

The construction sector is an important contributor to the economy for many countries. In Canada, the construction sector represents about 6.0 % of GDP, with a yearly volume of about CA$75 billion (Figure 1.1). This sector employs more than 1.2 million men and women, which is 7.1% of all employed Canadians (Canada year book 2011). In 2020, the Canadian construction sector employed 1.4 million people and generated CA$141 billion to the economy, accounting for 7.5 % of Canada's GDP (Canadian Construction Association, 2021). In the USA, the construction industry contributes about US$500 billion (3.5%) to the total GDP (Fig 1.2).

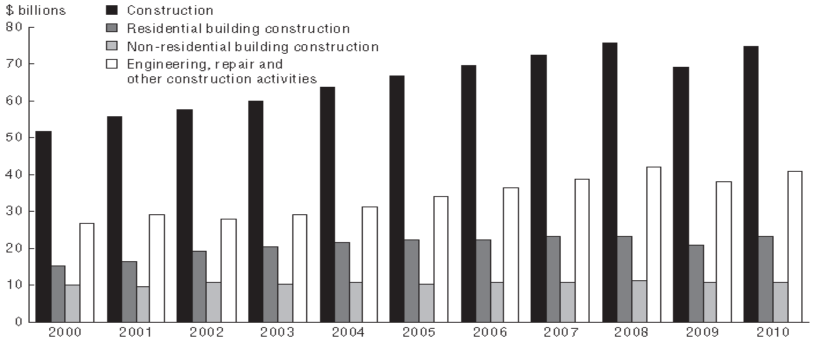


Figure 1.1 Canadian construction sector value (2000-2010)
(Canada year book 2011)

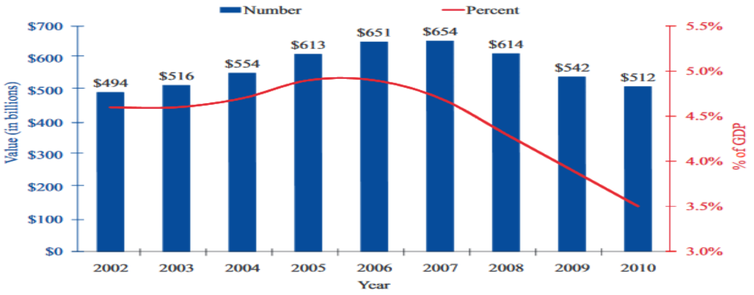


Figure 1.2 Value added by the construction industry and as % of GDP, US
(2010 Bureau of Economic Analysis report)

The construction industry often operates in a dynamic multi-party constrained environment that frequently leads to schedule and cost overruns, and disputes among parties, which results in reduced confidence in the efficiency of the construction industry (Arif & Morad, 2014). According to a report by Ernst & Young "Canadian infrastructure megaprojects run 39% (US$2.2b) over budget and behind schedule by 12 months on average. However, Canadian megaprojects perform better than those in the US, where the average project delay is a little more than three years"(Ernst & Young LLP, 2017). In a more recent report, Arcadis (2020) reported that North America's construction dispute costs US$ 18.8 million and lasts 17.6 months on average. In the US, by the end of the year 2013, more than 500 large complex construction cases were administered by the vice presidents and directors (American Arbitration Association (AAA), 2013). In the past few years, the number of claims submitted to the American Arbitration Association (AAA) reached almost 25% of the 1.7 million claims submitted over the past 74 years (Hegazy et al., 2011). A report by the National Audit Office of the UK, entitled "Modernizing Construction 2001" revealed that 70% of the projects undertaken by government departments and agencies were delivered late (Vasilyeva-lyulina, 2013). In the United Arab Emirates (UAE), a survey by Faridi & El-Sayegh (2006) revealed that 50% of construction projects encounter delays. Such project delays and cost overruns typically escalate into disputes among parties.

To mitigate construction disputes and fairly apportion project delays among the parties, forensic schedule analysis becomes necessary. In general, Forensic Analysis has been defined as "the study of how actual events interacted in the context of a complex model for the purpose of understanding the significance of a specific deviation or series of deviations from some baseline model and their role in determining the sequence of tasks within the complex network" (AACE RP 29R-03, 2011). In construction, delay analysis techniques have been recognized as the main forensic analysis tools to settle disputes. Based on the delay analysis results, the contractor may/may not be entitled to a time extension and/or compensation, or be liable for liquidated damages. In many cases, disputes escalate and lead to undesirable effects such as lawsuits between the parties or even contract termination (Menesi, 2007).

While some delay analysis techniques (But-For and Windows Analysis) are considered by arbitration boards and courts as acceptable tools to quantify responsibility, such techniques still have persistent issues, particularly when dealing with concurrent delays (Bhih & Hegazy, 2019, 2020b; Kao & Yang, 2009; Stumpf, 2000). Efforts in the literature introduced significant improvements to both techniques to enable it to identify concurrent delays and provide better concurrency assessment (e.g., Hegazy & Zhang (2005); Mbabazi et al. (2005)), some of the existing efforts are still unable to capture the dynamic fluctuation in the project critical paths and respect event chronology (Hegazy & Zhang, 2005), and thus, cannot adequately address concurrent delays. In addition, none of the existing efforts introduced a detailed methodology to allocate schedule accelerations, especially for the complex situations when delay/acceleration events simultaneously contribute to multi-day project delay or acceleration.

Recently, recommended practices and guidelines for delay analysis have been introduced by international organizations. For example, the Delay and Disruption Protocol (SCL-DDP, 2002) was introduced in the UK by the Society of Construction Law. In the USA, the American Association of Cost Engineering (AACE) also introduced its Recommended Practice for forensic delay analysis (AACE RP 29R-03, 2011). Subsequent revisions of the mentioned documents were issued by both organizations. Recently, in 2017, the American Society of Civil Engineers (ASCE) issued the schedule delay analysis standard (ASCE 67-17, 2017). The main purpose of these documents is to categorize and evaluate the varying delay analysis methods and to provide guidelines for fair delay analysis, as such any newly proposed improvements to delay analysis techniques should be matched with the guidelines provided by the mentioned documents.

1.2 Research Motivation

Due to the persistent drawbacks of current delay analysis techniques which are widely used in practice, this research targets to introduce an integrated daily windows with enhanced But-For delay analysis technique capable to accurately allocate both schedule delays and accelerations. The targeted technique has a clear and simple step-by-step methodology prepared in a computer-oriented fashion to minimize the analysis effort through automation. This research has been motivated by the following:

1.2.1 Need to correct the existing misinterpretation of delay analysis results

Some delay analysis techniques involve misconceptions in interpreting analysis results, particularly when the analysis is performed from different parties' viewpoints. The common application of the single window analysis techniques such as But-For (collapsed as-built) analysis involves such misconceptions, which render it unable to identify concurrent delays and produce conflicting results when adopting different viewpoints; this is highlighted in the literature (Dale & D'Onofrio, 2017; El-Adaway, Fawzy, Bingham, Clark, & Tidwell, 2014; Fawzy & El-Adaway, 2012; SCL-DDP, 2002). Addressing this misconception will improve the accuracy of the analysis of one of the widely used techniques by enabling it to identify concurrent delays and produce consistent results irrespective of the point of view.

1.2.2 Need for a clear methodology to allocate schedule accelerations

Schedule accelerations are common in construction projects as measures to expedite the work execution, recover past delays, or accommodate additional work. Acceleration is defined as carrying out the same amount of work in a shorter time or carrying out more work within the same time; in other words, acceleration means completing the work in less time than planned. Accelerating an activity may lead to a consequent time-saving in project duration if the activity being expedited was critical activity. In forensic analysis, acceleration is commonly treated as a negative delay. In the literature, delay analysis techniques have been developed considering schedule delays overlooking the common acceleration events. Having an analysis technique with a clear methodology able to allocate both schedule delays and accelerations will help to introduce a more fair and accurate analysis.

1.2.3 Need for accurate assessment of concurrent delays

Delay concurrency was addressed in different ways by many researchers in the literature (Hegazy & Zhang, 2005; Kim et al., 2005; Mbabazi et al., 2005), however, the issue need a better assessment of simultaneous daily events, particularly those causing more than one day impact, either delay or acceleration. To improve the current state of delay analysis techniques, the most challenging concurrent situations should be considered. Also, better consideration of the concurrent effect of sequential events will help to apportion concurrent delays accurately.

1.2.4 Need for an integrated delay analysis mechanism

Although several forensic delay analysis techniques have been used and widely accepted over the past few decades, particularly the But-For (BF) and the Windows analysis methods, yet both suffer from serious drawbacks. The But-For analysis although could be enhanced to accurately identify concurrent delays and accelerations, yet cannot capture the critical-path dynamics or the chronological order of all delay events of all parties due to its single window analysis. On the other hand, the windows delay analysis technique and its advanced version, Daily Windows Analysis (DWA) introduced by Hegazy & Zhang (2005), is capable of respecting all event chronology, yet does not assess concurrent delays, particularly in more complex delay and acceleration situations. As such both the BF and the DWA have their unique advantages and it is very beneficial to address their shortcomings and integrate their capabilities into a generic analysis technique to promote fairness, resolve claims quickly, and avoid disputes among project parties.

1.3 Research Objectives and scope

The research aims to develop a comprehensive delay analysis framework that can overcome the common delay analysis drawbacks, accurately assess concurrent delays and accelerations, and produce repeatable results considering the chronology of all parties' events on the schedule. Detailed objectives are as follows:

1. Investigate the varying definitions and methods of assessment of concurrent delays introduced by practitioners and the available industry guidance documents;
2. Introduce enhancements to the traditional But-For (BF) delay analysis technique and resolve existing misconceptions about BF results. Accordingly, introducing a Modified But-For (MBF) methodology to assess both concurrent delays and accelerations;
3. Fully integrate the MBF and the daily windows analysis and combine their benefits to establish the Enhanced Daily Windows Analysis (EDWA) as a more accurate forensic analysis framework that provides consistent results even in the most complex concurrent delay situations; and
4. Develop a computerized decision support system based on the EDWA framework and validate its benefits on several case studies.

This research covers delay analysis of projects within the construction industry considering CPM-based delay analysis techniques. This research attempts to introduce a better forensic delay analysis mechanism able to consistently integrate current analysis techniques to apportion concurrent and non-concurrent delays/accelerations and resolve issues not addressed before. The introduced model is expected to benefit both researchers and practitioners.

1.4 Research Methodology

The proposed research methodology is shown in Figure 1.3. The methodology is represented in sequential steps which follows the logical order of activities, however, the real application of it passed through several iterations that reflect the nature of research work which requires repeating and refinement. It can be described as follows:

1. Conduct an extensive literature review on forensic delay analysis aspects, techniques, and standards for construction projects
2. Study the varying concurrent delays definitions and situations.
3. Identify delay analysis techniques’ shortcomings.
4. Investigate the misconceptions involved in the traditional techniques.
5. Introduce improvements to the traditional But-For technique to resolve its misconceptions and improve its concurrency assessment.
6. Study the difference in assessing schedule delays versus accelerations.
7. Improve the Modified But-For method to allocate both schedule delays and accelerations.
8. Introduce integrated delay analysis technique (But-For and Window Analysis) to overcome their drawbacks. The hybrid technique aims at addressing complex situations of concurrent delays and considers events chronology and critical path(s) fluctuations.
9. Develop computerized forensic analysis MS-project macro to apply the integrated technique and validate it using case studies.

1.5 Thesis Organization

The remainder of the thesis is organized as follows:

Chapter 2: This is a literature review of schedule delay analysis. It introduces delay events versus project delay definitions and reviews the varying definitions of concurrent delays and accelerations and the recent efforts on progress documentation for the purpose of forensic analysis. The common delay analysis techniques are reviewed with a special focus on the But-For and Windows analysis techniques and their recent improvements. The existing industry guidance documents are reviewed; conclusions and knowledge gaps are presented at the end of the chapter.

Chapter 3: In this chapter, significant improvements are introduced to the traditional But-For technique. The common misinterpretation of results of the But-For technique (owner and contractor viewpoints) is explained by introducing the correct But-For results interpretation. Afterwards the But-For concurrency assessment is improved by introducing multi-window But-For analysis. The proposed improvements are then extended to the general case of three project parties.

Chapter 4: This chapter introduces the Enhanced Modified But-For Method (EMBF). Presenting explanations on how the MBF has addressed the But-For shortcomings of producing different results when adopting different party’s viewpoint and its inability to identify concurrent delays. Extending the MBF calculations to allocate both delays and accelerations and introducing multi-window MBF implementation to enable the technique to allocate simultaneous delay/acceleration events and to consider multiple baseline updates. Limitations and potential improvements are also explained.

Chapter 5: The Enhanced Daily Windows Analysis (EDWA) is introduced in this chapter. The EDWA is produced by integrating the recent improvements of the But-For and Windows analysis (Enhanced Modified But-For method (EMBF) and Daily Windows Analysis (DWA)). The EDWA methodology is introduced and explained on case study different scenarios to show the ability of the new technique to handle the most complex situations. Computer implementation is presented using MS-project VBA macro with a validation case study.

Chapter 6: summarizes the research and its contributions, and lists suggestions for future work.

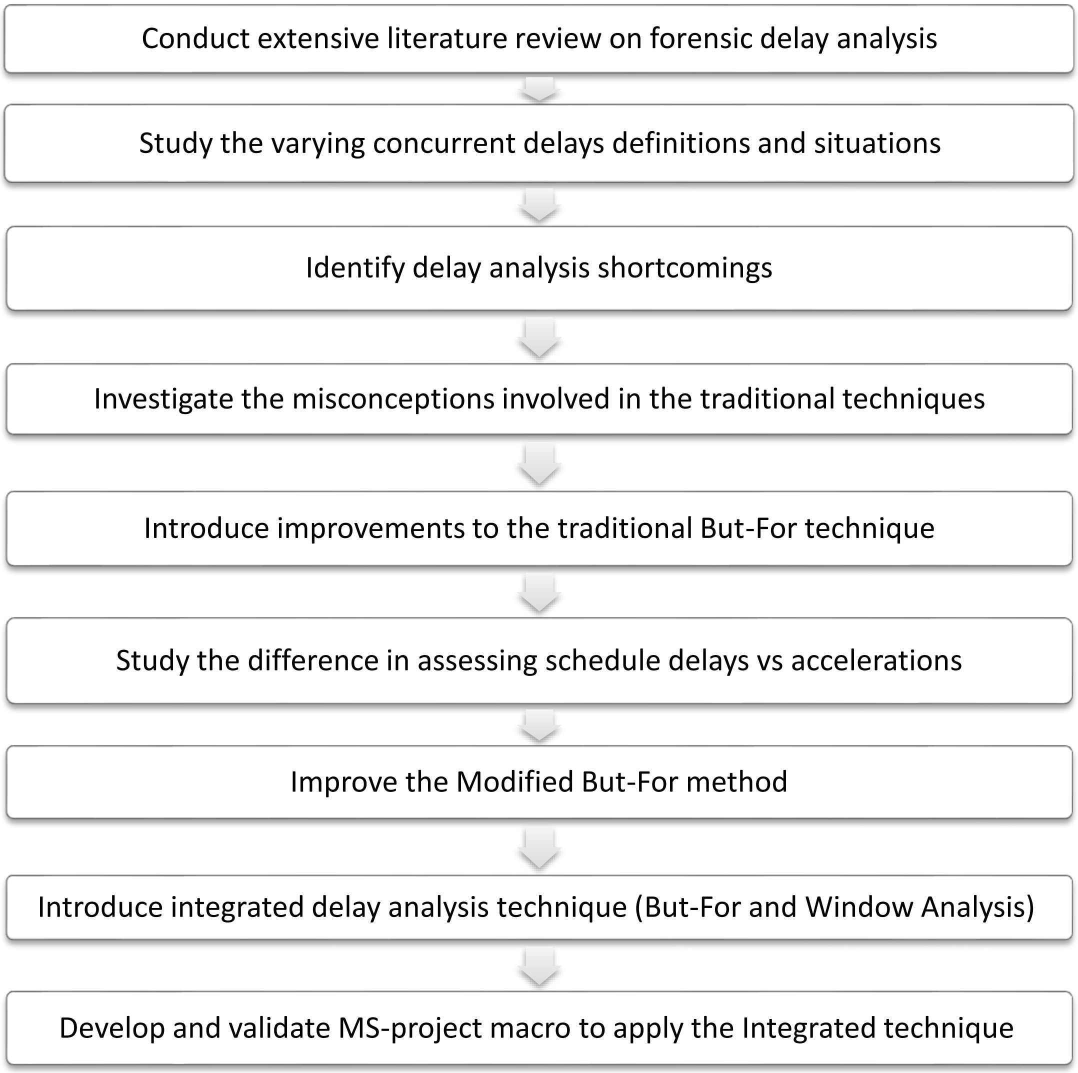

Conduct extensive literature review on forensic delay analysis
Study the varying concurrent delays definitions and situations
Identify delay analysis shortcomings
Investigate the misconceptions involved in the traditional techniques
Introduce improvements to the traditional But-For technique
Study the difference in assessing schedule delays vs accelerations
Improve the Modified But-For method
Introduce integrated delay analysis technique (But-For and Window Analysis)
Develop and validate MS-project macro to apply the Integrated technique

Figure 1.3 Research Methodology

Chapter 2
Literature Review

2.1 Introduction

Forensic schedule analysis is the main tool to investigate schedule-delay causes and allocate responsibilities among project parties (owner, contractor, and third party). Over the past few decades, the area of delay analysis received increasing interest from academic researchers, construction practitioners, and professional bodies. This chapter is dedicated to reviewing the literature on schedule delay analysis including causes of delays, advances in delay analysis techniques, and the guidelines published by professional societies. The conclusion of this chapter highlights research gaps and potential areas of improvement.

2.2 Claim Management Framework

Claim management is a bigger framework that starts from the project initiation phase and develops throughout the project and after the project completion (closing phase). The claim management framework includes claim prevention, mitigation, quantification, and resolution. In the initiation and the contracting phase, the contract documents are formulated properly to prevent claims throughout the balanced contract conditions and equitable risk-sharing among parties. In the construction phase, claim mitigation is an essential part of the framework; this can be achieved by punctual execution of activities, timely communication of project information; and taking the necessary corrective actions at the right time. When conflict arises, claim identification and quantification are needed. In the final part of the framework, the escalated claims are resolved. Claim resolution has different levels that are: negotiation among parties, mediation, arbitration, and litigation. Forensic analysis in its broader concept may involve all of the mentioned components of the framework. However, schedule delay analysis techniques, the topic of this research, are used during the construction and after the project completion to determine the responsibility and support the claim argument.

2.3 Delays in Construction Projects

Risk and uncertainty associated with construction projects often lead to project delays. On the other hand, schedule accelerations are commonly conducted voluntarily by the contractor or directed by the owner to recover delays, expedite the work, or accommodate more work within the same project duration. Schedule delay and acceleration responsibilities should be allocated to the causing party to determine the entitlement for time extensions and compensations.

2.3.1 Delay events

Delay event, is an event on the activity level that alters the as-planned schedule to reflect the actual progress of the activity; this event may or may not impact the project completion date based on the event criticality. For example, Figure 2.1 shows work suspension on Day 4 of activity C due to the Contractor (C); this is activity level delay caused by the contractor and called a “contractor delay event”. In this case, the contractor delay event affected an activity on one of the critical paths, thus, it is critical delay event and it caused project delay of 1 day. The Delay and Disruption Protocol (SCL-DDP, 2002) defines the delay event as an activity level event “which may be either Employer Risk Event or a Contractor Risk Event”; nevertheless, those events could be caused by a third party (such as acts of GOD) which represent all events beyond of the control of the owner and the contractor. The delay events are often caused by the action or inaction of one of the project parties (owner, contractor) and to be attributed to the causing party; if the event was due to a reason out of the owner and the contractor’s responsibility then it is called a third-party event.

If a particular activity affected by delay event was on the critical path, i.e. critical activity, then the event is critical and consequently will cause an impact on the project completion date. Critical events can also be on the near-critical paths if the event duration was greater than the available total float, so it consumes the available float and then causes further project delay. Events that do not affect the project duration are Non-critical events; however, such events may disrupt the contractor's work and cause cost impact (Figure 2.2). Disruption, as defined by Delay and Disruption Protocol (SCL-DDP, 2017), is " disturbance, hindrance or interruption to a Contractor's normal working methods, resulting in lower productivity or efficiency in the execution of particular work activities". Disruption analysis is different from delay analysis as it aims to demonstrate loss of productivity and incurring additional expenses more than what would have been incurred without disruption. Disruption analysis is beyond the scope of this study.

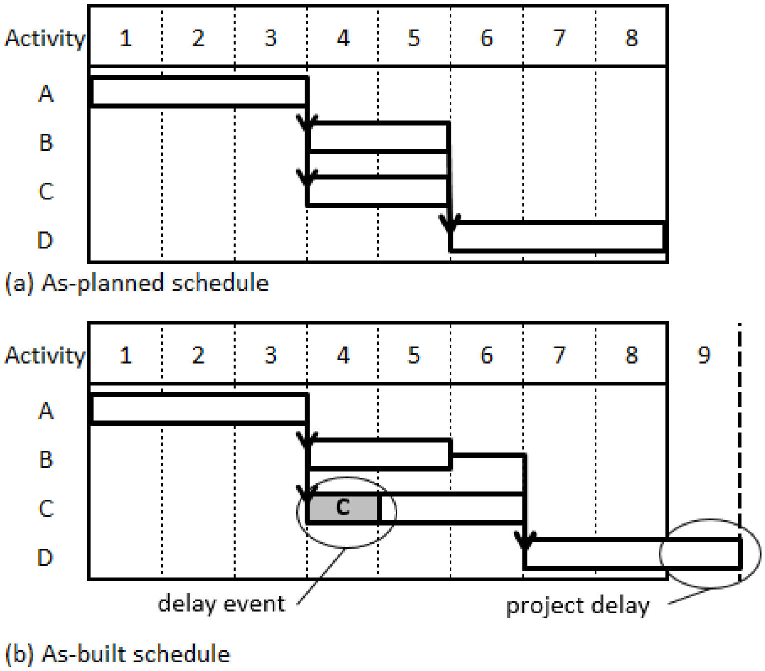


Figure 2.1 delay events and project delays

Delay events can be of several types such as:

- **Work stop** is an action of stopping the work in the activity, either before it starts (start delay) or during its progress (work suspension).
- **Slow progress (deceleration)** happens when the activity work continues but at a rate slower than planned.

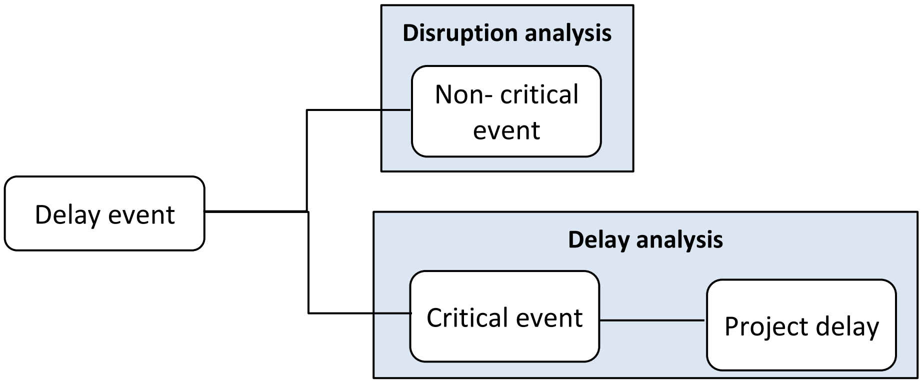


Figure 2.2 Disruption versus delay analysis

2.3.2 Project Delay

Project delay is the difference between the as-built duration and as planned duration; project delay can have either positive value (project completed behind schedule, net delay) or negative value (project completed ahead of schedule, net acceleration). The project delay is the combined effect of all of the delay events on the project completion date. In the general case of three project parties (Owner O, Contractor C, Their party N), there are seven components of project delays. In Figure 2.4a, Venn representation is used to show the seven components; OO, OC, ON are the sole responsibilities of the three parties while O⋂C, O⋂N, C⋂N, O⋂C⋂N are the components the concurrent delay. Based on the responsibility and entitlement to compensation, project delays are classified into three types: inexcusable delays, excusable compensable delays, and excusable non-compensable delays, (Vasilyeva-lyulina, 2013) (Figure 2.3). Project delay types are defined as follows:

- **Inexcusable delay** is a project delay caused by the contractor (AACE RP 10S-90, 2010), in such a situation the contractor will not be entitled to time extension nor compensation and will be liable to liquidated damages to the owner.
- **Excusable compensable delay** is a project delay caused by the owner (AACE RP 10S-90, 2010). When the project delay is caused by action or inaction of solely the owner, the contractor will be entitled to time extension and compensation for the cost impact of this delay.
- **Excusable non-compensable delay** is the project delay caused by a third party or caused by two or more project parties concurrently (concurrent delays) (AACE RP 10S-90, 2010). In such cases, the contractor is entitled only to a time extension.

Based on these definitions the seven components of project delays (OO, OC, ON, O∩C, C∩N, O∩N, and O∩C∩N) can be correlated with the three types of project delays (Excusable – Compensable, Non-excusable, Excusable - Not-Compensable) as shown in figure 2.4b. This correlation is also matched with (AACE RP 29R-03, 2011) requirements as shown in Table 2.1. Table 2.1 presents the net effect of different scenarios of concurrent events attributed to different parties as defined by AACE RP 29R-03 (2011).

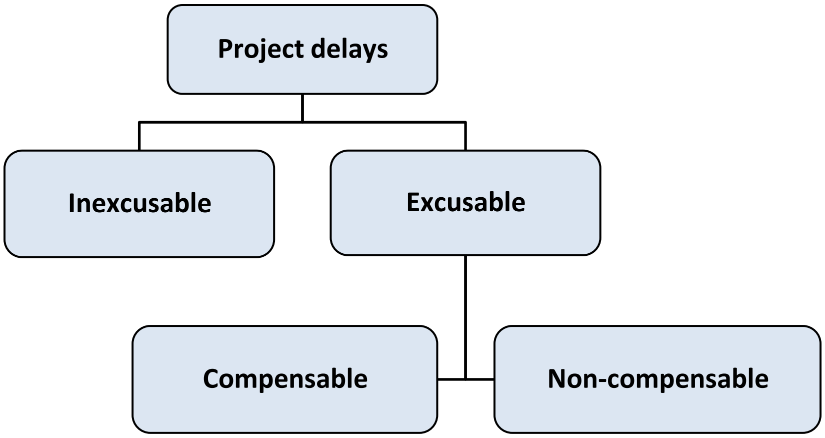
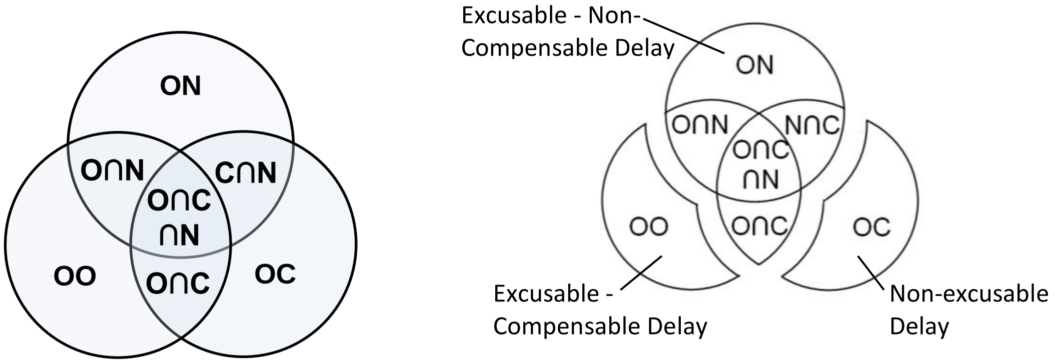


Figure 2.3 Types of project delay



a) Case of three-party delays b) Correlation of project delay components and types

* OO= Only Owner; OC = Only Contractor; ON = Only third party

Figure 2.4 Venn representation of project delays – case of three parties

Table 2.1 Compensation rules for concurrent delays (AACE RP 29R-03, 2011)

Delay event	Concurrent with	Net effect
Owner delay	Another owner delay or nothing	Compensable to contractor. Non-excusable to owner
Owner delay	Contractor delay	Excusable but not compensable to both parties
Owner delay	Force majeure delay	Excusable but not compensable to both parties
Contractor delay	Another contractor delay or nothing	Non-excusable to contractor. Compensable to owner
Contractor delay	Force majeure delay	Excusable but not compensable to both parties
Force majeure delay	Another force majeure delay or nothing	Excusable but not compensable to contractor

2.3.3 Concurrent delays

Concurrent delays usually occur at the peak of projects when multiple activities are running at the same time (Baram, 2000). It happens when multiple delay events, not exclusively under the control of one party, affect the critical path/s of the project (ASCE 67-17, 2017); the project delay in such situations is not the sole responsibility of one party. Apportioning responsibility for concurrent delays is a challenging task since it involves the effect of multiple causing events (Ibbs, Nguyen, & Simonian, 2011). Delay analysts must take into account concurrent delays to equitably allocate project delays and to avoid multiple counting of delay days. (Lee, 2007). Concurrent delay is defined as two or more delays that occur at the same time, either of which would cause a project delay (i.e. both are critical delays); if one of them had not occurred, the project would have been delayed by the other (Stumpf, 2000) (Figure 2.5).

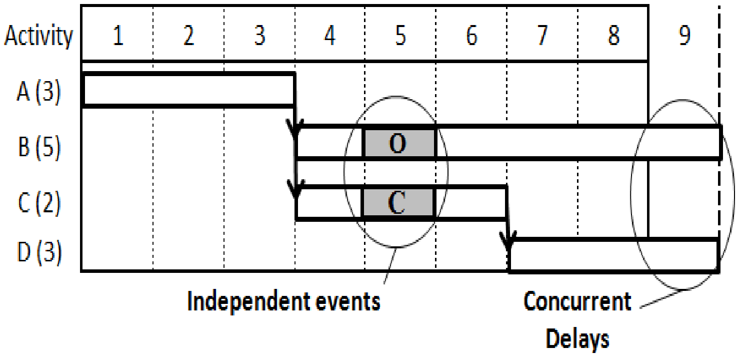


Figure 2.5 Concurrent delays

Concurrent delays were covered in the literature by several researchers. Apportioning responsibility for concurrent delays was introduced earlier by Kraiem & Diekmann (1987) using the adjusted as-built schedule; the adjusted schedule that contains only the considered concurrent delays is then compared with the as-built schedule. Arditi & Robinson (1995) criticized the adjusted as-built schedule approach for its inability to capture the critical path dynamics and introduced improvements to it. Two-step window analysis to apportion concurrent delays among the owner and the contractor was proposed by Baram (2000). Zhang & Hegazy (2005) introduced concurrent delay analysis using daily windows delay analysis considering work stops, slowdowns, and accelerations by monitoring the near-critical path float. Concurrent delay analysis considering linear production rate was proposed by Lee (2007). Ibbs et al. (2011) discussed the apportionment of damages between project parties in concurrent delay situations.

Concurrent delays also have varying definitions by the industry guidance documents; currently, three guidance documents are available in practice issued by the Society of Construction Law (SCL) of the UK, and The Association of the Advancement of Cost Engineering International (AACEI) of USA, and American Society of Civil Engineers (ASCE). These definitions reflect the differing viewpoints that originated from the inconsistent application of concurrent delay in practice (Livengood, 2017).

The Delay Disruption Protocol (SCL-DDP, 2017) distinguishes between true concurrent delay and the concurrent effect of sequential delay events. It 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. The concurrent effect of sequential delay events happens when two or more delay events, not exclusively under the control of one party, happen at different times, but the effects of them are felt at the same time.

The Forensic Schedule Analysis RP (AACE RP 29R-03, 2011) defines 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), and has a more flexible position regarding the time period from which concurrency is measured, as opposed to literally within the exact period of time. As such, AACE recommended practice accepts both the literal and functional concurrency definitions, reflecting the American practice of concurrent delays.

The Schedule Delay Analysis proposed standard (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".

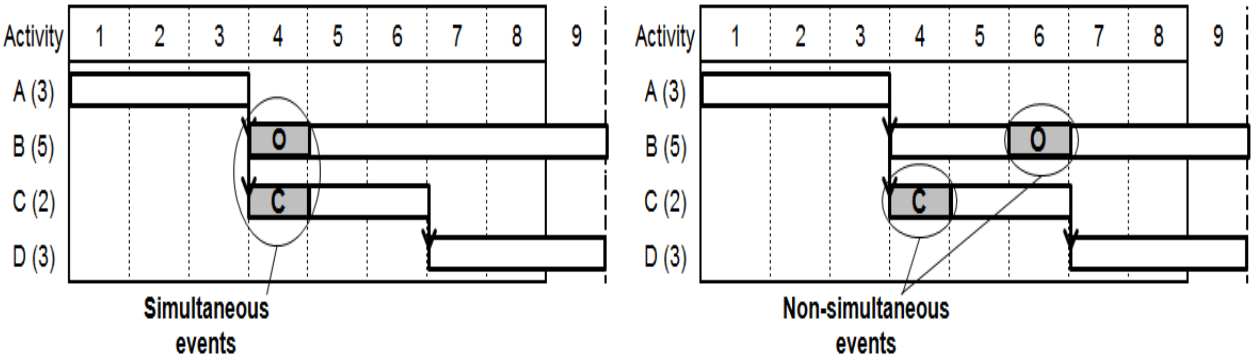
2.3.3.1 Literal Concurrency versus Functional Concurrency

In the literal concurrency, the causing events should be simultaneous, i.e. happening exactly at the same time, so both parties contribute to project delay. Under the functional theory, it is accepted that the causing events are close in time but may not be happening exactly at the same time. 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 (Ibbs et al., 2011). 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.

In this research, literal concurrency is adopted assuming that the work progress is recorded accurately, and thus the as-built schedule exactly describes the actual progress. Adopting literal theory gives the most accurate and equitable analysis results. This assumption was made to present the ideas in a clear and consistent manner. However, whenever it is appropriate, the other concurrency viewpoints are introduced.

2.3.3.2 True Concurrency and Concurrent Effects

The Delay and Disruption Protocol (SCL-DDP, 2017) adopts the literal concurrency viewpoint and calls it "true concurrency", any two or more events that happen at different times but its effects are felt at the same time are called "concurrent effects". To explain the difference between the two situations, the simple progress scenarios in Figure 2.6 are used. In Figure 2.6a, two simultaneous delay events are 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 2.6b, the contractor event is on Day 4 while the owner event is on Day 6. Both of the events are 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 as it contributed first to Day 9 project delay.



a) True concurrent delays

b) Concurrent effects

Figure 2.6 True concurrent delays versus concurrent effects

2.3.3.3 Pacing Delay and Offsetting Delay

Pacing delay and offsetting delay are project delays that have some flavor of concurrent delays (Dale & D'Onofrio, 2017). Pacing delay is the action of relaxing the progress of activity by the contractor as to pace progress of another activity being delayed by the owner, having that the contract gives the contractor the right to pace. In pacing delay, the owner causes delay to the critical path (parent event), and the contractor opts to relax its performance (pacing event) to the extent that it does not affect project completion (Figure 2.7). According to (AACE RP 10S-90, 2010) pacing event is "deceleration of the project work, by one of the parties to the contract, due to a delay to the end date of the project caused by the other party, so as to maintain steady progress with the revised overall project schedule". "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 evidence of intent to pace (AACE RP 29R-03, 2011). In pacing situations, the resulted project delay is to be attributed to the owner only. Pacing delay, however, was not discussed by (SCL-DDP, 2017) and (ASCE 67-17, 2017) guidelines.

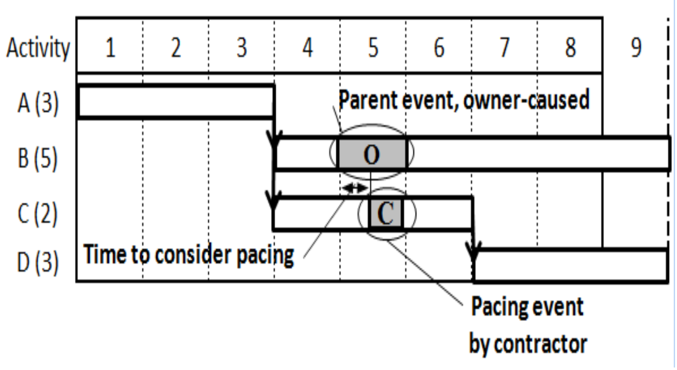


Figure 2.7 Pacing delay

The offsetting delays have been firstly introduced by (ASCE 67-17, 2017) and defined 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 a situation, any owner-caused delay may offset the contractor’s liquidated damages, even if it does not 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 to the owner and contractor (Nagata, 2018). This debate is due to the recent viewpoints on the definition of critical activities, in view of the complexities that result from using the advanced features of scheduling software such as schedule constraints and multiple resource calendars. Some practitioners and industry standards adopt the longest path/critical path to identify critical activities, while others define critical activities as those having zero or negative floats. (AACE RP 29R-03, 2011) adopt the more typical definition, while (ASCE 67-17, 2017) adopts the latter definition and uses it to justify offsetting delay. The offsetting delay is considered as a sort of concurrent delay 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.

2.3.4 Acceleration

Acceleration, as opposed to delay, is defined as carrying out the same amount of work in a shorter time or carrying out more work within the same time. In other words, acceleration means completing the work in less time than planned, leading to a consequent time-saving in project duration. For fairness, acceleration is to be attributed to the party who incurs its cost; i.e., owner-directed acceleration is to be credited to the owner, while the contractor-voluntary acceleration is to be credited to the contractor (AACE RP 29R-03, 2011). It is reasonable also to use one party's acceleration to offset that party's delays (K. Zhang & Hegazy, 2005) as it is commonly treated as a negative delay. In contrast to delays, project acceleration can only be attributed either to the owner or to the contractor, but not to any third party. Therefore, the acceleration responsibilities are Only-Owner Acceleration (OOA); Only-Contractor Acceleration (OCA); and Concurrent Owner and Contractor Acceleration (OA∩CA). Venn representation of the three acceleration components is explained in Figure 2.8.

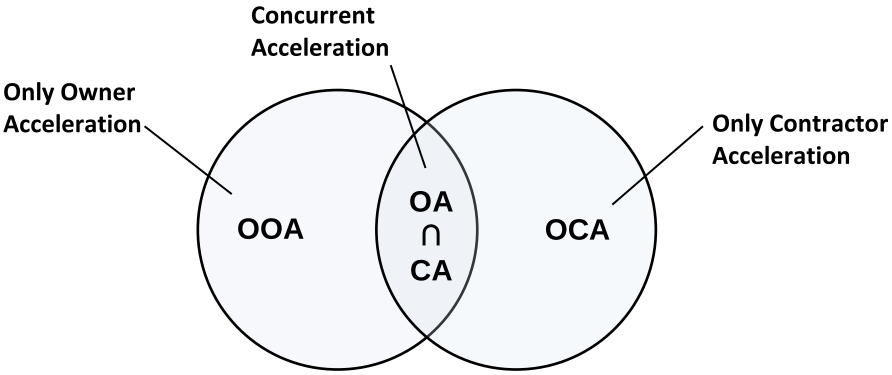


Figure 2.8 Venn representation of project acceleration

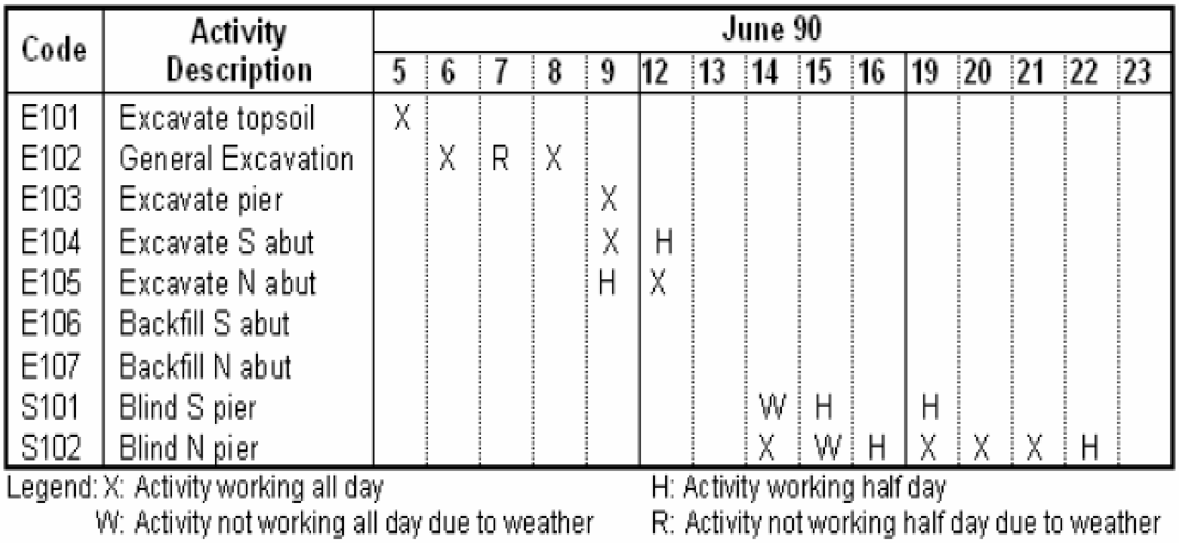
2.3.5 Documentation of progress events

The first step in the delay analysis is to identify and quantify the delay events from the available sources of data. As-planned and as-built schedules, schedule updates, and project documents and records are the main data sources for delay analysis. The analyst is firstly to identify and quantify all of the delay events and determine the responsibility of events. The delay analysis quality and reliability are dependent on the quality and accuracy of data sources as well as the reliable use of data in the analysis process (AACE RP 29R-03, 2011). The delay analysis process is highly sensitive to the timing of the delay events especially when concurrent events exist.

Project progress data are commonly recorded in different formats such as periodical reports (daily, weekly, monthly), quality control and assurance records, weather records, safety reports, progress and ad-hock minutes of meetings', inspection orders and reports, project photographs, and much more documents. These documents and records are normally kept in paper and/or digital format. The use of cloud-based electronic document management systems (EDMS) became more popular for better sharing and collaboration among the project parties and stakeholders. The documentation should include but is not limited to: weather, staffing, resources' availability at the site, material delivery, inspections, work orders, safety issues and accidents, quality control activities, and change orders (Menesi, 2007). To identify and quantify the delay event, the analyst should compile all those sources of data to arrive at a realistic image that reflects the actual progress.

In the literature, there are limited attempts to improve the progress data documentation for the purpose of forensic schedule analysis. An early attempt by Scott (1990) involved developing a computer program, Record-Keeper, that keeps the project site records in a simple fashion for easy use by delay analysts. The program allows daily record of progress on all project activities (Figure 2.9) with the following choice of options: X - Activity working all day; H - Activity working half-day; W - Activity not working all day due to weather; R - Activity not working half day due to weather.

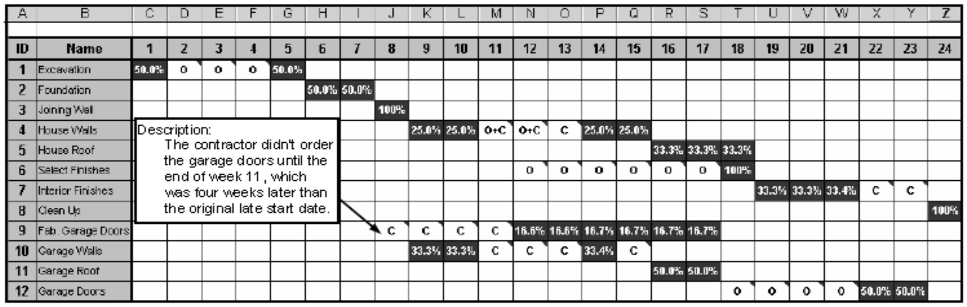
Hegazy et al. (2005) introduced the intelligent bar chart as a tool to record the actual work progress. The intelligent bar chart is made of spreadsheet cells, each representing unit of time (e.g. day, week, or month) for the activity. The activity bar is represented as a group of adjacent cells making up the duration of the activity and records the daily percentage completed of each activity, activity level delays and responsible party, and any other related data. Accordingly, slow progress can be identified when actual progress proceeds with lower productivity than planned; acceleration, when work proceeds with higher productivity than planned; and suspension, when work is completely stopped(Menesi, 2007). Delays are recorded on the bar chart on the day they occur. As shown in Figure 2.10, an “O” is shown for owner-caused delay, a “C” is shown for contractor-caused delay, an “N” is shown for third-party-caused delay or acts of God (e.g., weather). A combination of these three letters is shown (e.g., “O+N” or “O+C”) to show events with joint responsibility. Joint responsibility happens when more than one party cause delay to the same activity jointly at the same time. For example, if the work was suspended in one activity due to the inability of the contractor to mobilize the required resources and the owner was late in approving the shop drawing for the same activity; in such case, the owner and the contractor are jointly responsible for this event. To distinguish between joint responsibility and concurrent delays, the joint responsibility is delay event (activity level delay) while the concurrent delay is a project delay.



Code	Activity Description	June 90														
		5	6	7	8	9	12	13	14	15	16	19	20	21	22	23
E101	Excavate topsoil	X														
E102	General Excavation		X	R	X											
E103	Excavate pier					X										
E104	Excavate S abut					X	H									
E105	Excavate N abut					H	X									
E106	Backfill S abut															
E107	Backfill N abut															
S101	Blind S pier								W	H		H				
S102	Blind N pier								X	W	H	X	X	X	H	

Legend: X: Activity working all day H: Activity working half day
W: Activity not working all day due to weather R: Activity not working half day due to weather

Figure 2.9 Recording site data (Scott, 1990)

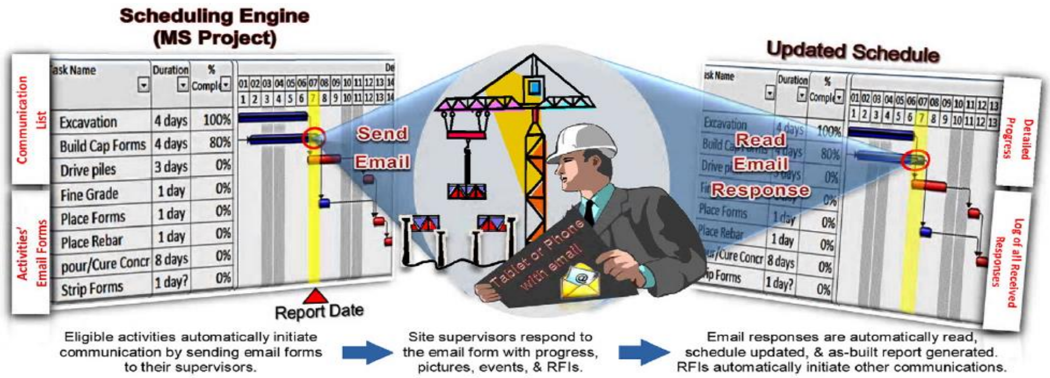


A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
ID	**Name**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
1	Excavation	50.0%	O	O	O	50.0%																			
2	Foundation						50.0%	50.0%																	
3	Joining Wall								100%																
4	House Walls									25.0%	25.0%	O+C	O+C	C	25.0%	25.0%									
5	House Roof																33.3%	33.3%	33.3%						
6	Select Finishes												O	O	O	O	O	O	100%						
7	Interior Finishes																			33.3%	33.3%	33.4%	C	C	
8	Clean Up																								100%
9	Fab. Garage Doors								C	C	C	C	16.6%	16.6%	16.7%	16.7%	16.7%	16.7%							
10	Garage Walls									33.3%	33.3%	C	C	C	33.4%	C									
11	Garage Roof																50.0%	50.0%							
12	Garage Doors																		O	O	O	O	50.0%	50.0%	

Description:
The contractor didn't order the garage doors until the end of week 11 , which was four weeks later than the original late start date.

Figure 2.10 Intelligint bar chart (Hegazy et al., 2005)

Hegazy & Abdel-Monem (2012) have introduced the email-based system for documenting the as-built details utilizing the email for progress tracking and communication between project parties (Figure 2.11). In the proposed system, the schedule automatically initiates email inquiries about activities progress information, then the system reads the response and automatically updates the schedule.



ask Name	Duration	% Comple
Excavation	4 days	100%
Build Cap Forms	4 days	80%
Drive piles	3 days	0%
Fine Grade	1 day	0%
Place Forms	1 day	0%
Place Rebar	1 day	0%
pour/Cure Concr	8 days	0%
Strip Forms	1 day?	0%

Eligible activities automatically initiate communication by sending email forms to their supervisors.

Site supervisors respond to the email form with progress, pictures, events, & RFIs.

Email responses are automatically read, schedule updated, & as-built report generated. RFIs automatically initiate other communications.

Figure 2.11 Email based system for documenting as-built details (Hegazy & Abdel-Monem, 2012)

Significant research efforts have been introduced, in the literature, in the past two decades to utilize recent technologies in automating on-site construction progress tracking (Kopsida & Vela, 2015). This automation process aims at increasing the monitoring accuracy, providing data on a real-time basis, and minimizing human intervention. Automated progress monitoring is important to detect construction deviations, assess actual construction progress, and determine as-built delays compared to the plan (Ekanayake et al., 2021). Such efforts involve a wide range of technologies that can be categorized under four groups: enhanced IT technologies, geospatial technologies, and imaging technologies (Omar & Nehdi, 2016). IT technologies include the use of multimedia tools (photos, audio, and video recordings), email-based services, and handheld technologies. Geospatial tools such as barcodes, RFID tags, Ultra-Wideband, and GIS and GPS help to monitor on-site objects (materials, tools, equipment …etc.). Imaging technologies such as photogrammetry, laser scanning, and 3D imaging are utilized for the activities progress tracking (Zhang & Arditi, 2013). Despite those efforts, such technologies were not widely adopted in the industry due to its high cost, the difficulty of implementation and need for skilled staff, and lack of information regarding the improvement in the documentation process compared to the conventional documentation practices (Alizadehsalehi & Yitmen, 2019). Such progress documentation efforts could be employed for the purpose of forensic analysis by reflecting work suspensions, accelerations, and slowdowns on project schedules.

2.4 Literature on delay causes

Lack of proper planning is one of the main causes of construction delays, however, many other causes have been reported in the literature (Guévremont & Hammad, 2020). Most of the studies adopt questioners as study methodology; however few studies collected data through real project delay cases and interviews with subject matter experts. Menesi (2007) and Yang et al. (2013) reviewed delay causes from literature; Table 2.2 presents an updated list of efforts on delay causes. The summary of studies shown covers a wide spectrum of construction projects (residential, health buildings, communication facilities, water supply and sewer systems, groundwater, build operate transfer BOT projects, etc.) in several countries and regions; it is presented in chronological order.

Table 2.2 Efforts on construction schedule delay causes

Research	Country	Causes of delay	
(Baldwin et al. 1971)	United States	- inclement weather - shortages of labour supply	- subcontracting system
(Arditi et al. 1985)	Turkey	- shortages of resources - financial difficulties - organizational deficiencies	- design delays - frequent changes - additional work
(Okpala & Aniekwu, 1988)	Nigeria	- shortages of materials - failure to pay for work	- poor contract management
(Dlakwa & Culpin, 1990)	Nigeria	- delays in payment - fluctuations in materials	- labour and plant costs
(Mansfield et al. 1994)	Nigeria	- improper financial and payment arrangements - poor contract management	- shortages of materials - inaccurate cost estimates - fluctuations in cost
(Semple et al. 1994)	Canada	- increases in the scope - inclement weather	- restricted access
(Assaf et al. 1995)	Saudi Arabia	- slow preparation and approval of shop drawings - delays in payments	- changes in design/design error - shortages of labour supply - poor workmanship
(Ogunlana et al. 1996)	Thailand	- shortages of materials - changes of design	- liaison problems among the contracting parties
(Chan & Kumaraswamy, 1996)	Hong Kong	- unforeseen ground conditions - poor site management and supervision	- slow decision by project teams - client-initiated variations
(Al-Khalil & Al-Ghafly, 1999)	Saudi Arabia	- cash flow problems/financial difficulties	- difficulties in obtaining permits -“lowest bid wins” system
(Al-Momani, 2000)	Jordan	- poor design - change orders/design - inclement weather	- unforeseen site conditions - late deliveries
(Odeh & Battaineh, 2002)	Jordan	- Owner interference - Inadequate contractor experience	- Financial and payments of completed work
(Frimpong & Oluwoye, 2003)	Ghana	- Monthly payment difficulties - Poor contractor management	- Material procurement
(Koushki et al. 2005)	Kuwait	- Change orders - Financial constraints	- Owner’s lack of experience
(Lo et al. 2006)	Hong Kong	- inadequate resources	- conflict with existing utilities

		- unforeseen ground conditions - exceptionally low bids - inexperienced contractor	- poor site management and supervision -unrealistic contract duration
(Faridi & El-Sayegh, 2006)	UAE	- slow preparation of drawings - inadequate planning of the project - low productivity of manpower	- slowness of owner’s decision - shortage of manpower - poor management and supervision
(Assaf & Al-hejji, 2006)	Saudi Arabia	- change orders by the owner - delay in progress payment - contractor difficulties in financing	- ineffective scheduling - shortage of labor
(Sambasivan & Soon, 2007)	Malaysia	- Contractor’s improper planning - Contractor’s poor management	- Inadequate contractor experience
(Sweis et al. 2008)	Jordan	- Financial difficulties by contractor - Too many change orders	- Poor planning and scheduling
(Le-Hoai et al. 2008)	Vietnam	- Poor management and supervision - Financial difficulties of owner	- Poor project management assistance
(Abd El-Razek et al. 2008)	Egypt	- Financing by contractor - Delays in contractor’s payment	- Design changes by owner
(Yang & Ou, 2008)	Taiwan	- Unforeseen site conditions - Lack of resources	- Late drawings and specifications delivery
(Yang & Wei, 2010)	Taiwan	- Changes in client’s requirement - Complicated administration process of client - Insufficient or ill-integrated basic project data	- Changes in client’s requirement - Inadequate integration on project interfaces - Change orders by deficiency design
(Yang et al. 2010)	Taiwan	- Improper contract planning - Debt problem	- Uncertainty on political issues and government- finished items
(Yang et al. 2013)	Taiwan	- change orders - changed scope of the work	- delayed site handover - weather
(Yusuwan & Adnan, 2013)	Malaysia	- inadequate scheduling or mismanagement - construction mistakes - equipment breakdowns - staffing problems	- owner failure to site hand over - changes orders - defective designs - differing site conditions
(Santoso & Soeng, 2016)	Cambodia	- adverse weather conditions - low productivity of manpower - poor supervision - differing site condition	- “lowest bid wins” system - equipment breakdown - poor site management - late contractor payments
(Chen et al. 2019)	China	- inadequate equipment - poor communication among parties - change orders	- inadequate design team experience - subcontracting problems

2.5 Delay Analysis Techniques

Delay analysis is defined as the study and investigation, using CPM or other recognized methods, of how actual events interacted into a complex model to understand the significance of a specific deviation or series of deviations from some baseline model (AACE RP 29R-03, 2011). The main target of schedule delay analysis is to apportion responsibility of project delays among project parties. Based on the assigned responsibilities, if the project is still underway, either an extension of time is granted to the contractor or a recovery plan is requested from it to mitigate further delays using schedule compression strategies. Analyzing the delays after the project completion, or after certain project milestones, aims at assessing the entitlement for time extensions and compensations by considering certain compensation rules.

Delay analysis can be either prospective or retrospective (AACE RP 29R-03, 2011). The prospective analysis is to be performed before the delay event, or just after the event gets started and before its end, to forecast the future effect of the expected or ongoing event on the project completion date. Prospective analysis is normally used by the project parties as a forecasting tool to plan ahead of time to mitigate the undesirable effects of any expected delay events. Retrospective analysis is to be performed soon after a particular delay event has occurred or after the overall project completion. Forensic delay analysis investigates the project delay responsibilities and the entitlement to an extension of time and compensation.

Several delay analysis techniques are available in practice and mentioned in the literature. The use of CPM analysis to support claims has been in use for more than 40 years, however, no particular method is used consistently to prove delay responsibility (Guévremont & Hammad, 2018). Developments to the existing techniques and new proposed techniques have been introduced and used over the past few decades. Delay analysis techniques have different data requirements and different methodologies; some of the techniques adopt a simple approach by comparing the project as-planned schedule with the as-built schedule, and some of them use the CPM models to conduct the required analysis. Different techniques may arrive at different results for the same situation, therefore no one technique is generally accepted for all situations(Yang & Kao, 2009). The selection of the proper technique depends on several factors such as dispute size, available time, available data, analysis budget(Menesi, 2007). AACE RP 29R-03 (2011) recommended 11 factors to be considered to select a technique for a particular situation. Yang & Kao (2009) reviewed delay analysis techniques found in the literature; more techniques exist in literature and are not listed in the mentioned study, however, some of them are identical with minor differences; some of the proposed techniques are not yet being used in practice.

The most known techniques in forensic analysis practice are “as-planned vs as-built”, “impacted as-planned (what-If)”, “Collapsed as-built (But-For)”, and “Window analysis”; among the four techniques, the But-For and Window analysis are often used by practitioners and accepted by courts and arbitration boards. However, both of them have been criticized in literature for some persistent drawbacks. Arditi & Pattanakitchamroon (2006) have summarized the comments on both techniques in tabular form; this summary was updated and presented in Table 2.3. The table shows the advantages and disadvantages of both techniques from the researchers’ and practitioners’ points of view.

Table 2.3 Comments on But-For and Window analysis in literature

Reference	Analysis technique	
	But-For analysis	**Windows analysis**
(Dale & D’Onofrio, 2017)	Widely used	Widely used
(SCL-DDP, 2017)	measures only incremental delay to the nearest critical path	Reliable

(El-Adaway et al., 2014)	Highly subjective	N/A
(Braimah, 2013)	Different results by different party's viewpoint	Very effective approach; the more windows the better the accuracy
(Yang & Kao, 2009)	Retrospective, widely used	Accurate; widely used
(Lovejoy, 2004)	Very good	Excellent
(Sgarlata & Brasco, 2004)	Most acceptable by courts	Useful for prospective analyses, but minimal utility supporting claims
(Sandlin et al. 2004)	Erroneous evaluation	Overcomes some disadvantages of others
(Gothand, 2003)	Major drawbacks	Reliable
(SCL-DDP, 2002)	Suitable for some situations, subjective	Most reliable when available
(Harris & Scott, 2001)	Fair, most accepted	Make some use by claims consultants
(Zack, 2000)	Unreliable, easy to manipulate	Accurate but expensive
(Fruchtman, 2000)	No baseline needed, limited	Contemporaneous basis, but future changes not considered
(Stumpf, 2000)	Easy to prepare, fundamental flaws	Reliable, but time consuming
(Finke, 1997, 1999)	Less reflective of actual events	Most reasonable and accurate
(McCullough, 1999)	Useful in some situations but easy to manipulate	Dependent on baseline schedule, accurate
(Zack, 1999)	Suitable	Suitable
(Bubshait & Cunningham, 1998a, 1998b)	Acceptable, dependent on availability of data	Acceptable, dependent on availability of data
(Levin, 1998)	Dependent on quality of as-built schedule	Dependent on how the method is applied
(Alkass et al. 1996)	Sound, but ignores changes of critical paths	Some drawbacks/propose modified method
(Zafar, 1996)	Fault analysis	N/A
(Schumacher, 1995)	Fault analysis	Effective method
(Baram, 1994)	Most practical in some circumstances	Most desirable approach
(Wickwire et al. 1991)	Alive and well	Recommended
(Bramble & Callahan, 1987)	Acceptable, dependent on availability of data	N/A

The four common techniques are briefly described in the following subsections with more focus on the widely used techniques (But-For and window analysis). Recent improvements introduced in the literature to But-For and Windows analysis are also discussed.

2.5.1 As-planned versus As-built

In this technique, the as-planned schedule (baseline) and as-built schedule (or any other schedule update that reflects progress) are compared to observe the overall project delay and to identify the causation between project parties and the project delay. AACE RP 29R-03 (2011) classifies this technique as an observational technique on contrary to the modeled techniques (CPM-based techniques); the same recommended practice methods of implementations MIP 3.1 to MIP 3.5 describe the observational techniques considering gross or periodic implantations with or without baseline updates. As-planned versus as-built technique is the simplest delay analysis technique with low cost and analysis time; it has been criticized because it determines the impact of all events as a whole rather than studying each event separately (Menesi, 2007).

2.5.2 What-If Technique (Impacted As-planned)

What-if analysis is also known in the literature as impacted as-planned delay analysis. What-if analysis is conducted by adding the events of one party to the as-planned schedule to determine how the as-planned schedule will be impacted as a result of the added event and to determine the considered party project delay responsibility. What-if analysis needs limited time and effort; however, it is less popular because it relies on the less realistic schedule that is the as-planned schedule. What-IF analysis is to be conducted from one party viewpoint; it may produce incompatible results when a different viewpoint is adopted. From the owner’s point of view, by adding the owner-caused events to the as-planned schedule; the owner is said to be responsible for the difference in duration between the as-planned schedule and the produced what-if schedule; the remaining balance of the schedule delay is attributed to the other party. Similarly, in the contractor viewpoint, adding contractor-caused events to as-planned schedule will produce the contractor's responsibility, the remaining balance is the owner's responsibility.

What-if technique is classified by (AACE RP 29R-03, 2011) recommended practice (RP) as one of the additive techniques; in the additive techniques, delay events are added to the as-planned schedule to assess its impacts and allocate project delay responsibility. The RP method of implementation (MIP 3.6 Modeled / additive / Single base) resemble what-if technique. The RP describes the method as a modeled technique since it relies on a simulation of a scenario based on a CPM model. The simulation consists of the addition of delay events into as-planned schedule to determine the hypothetical impact of those inserted activities on the network. What-if analysis is criticized for producing different results when adapting different party's points of view as reported by (Braimah, 2013; Mbabazi et al., 2005). Moreover, what-if analysis is unable to identify concurrent delays and does not consider the events' chronology since it adds all party's events, one party at a time, regardless of the sequence. These shortcomings have limited its use in practice despite its simplicity.

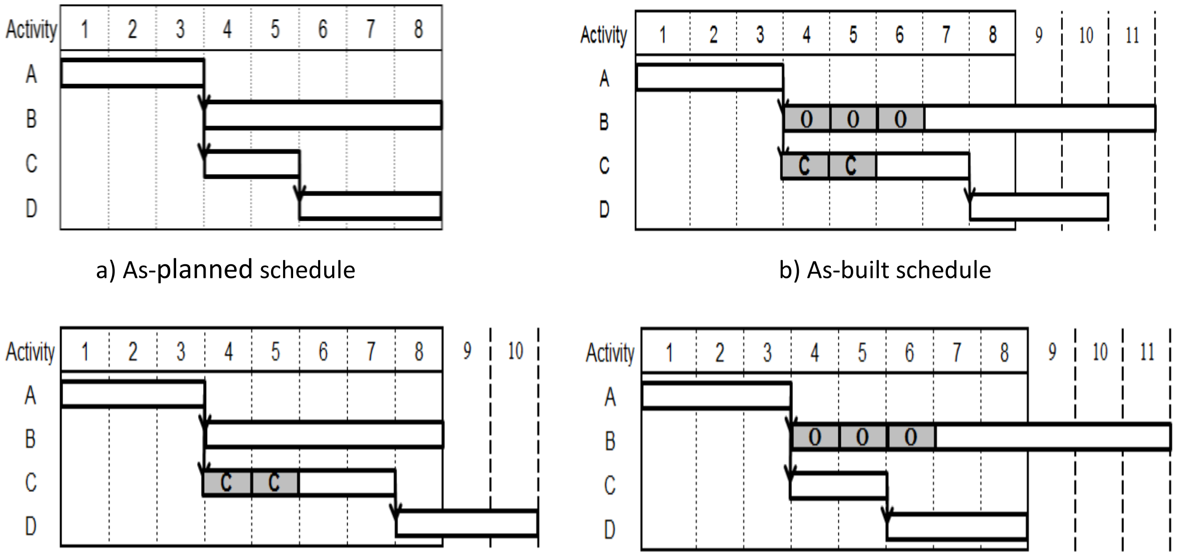
2.5.3 But-For Technique (Collapsed As-built)

But-for analysis is also known in the literature as collapsed as-built delay analysis. It is used in the preparation and justification of delay claims with less time and effort (Zack, 2000). Although it is more reliable than several other delay analysis techniques, it is too easy to manipulate (Zack, 1999). Due to its simplicity, however, it has been accepted in courts (e.g., the case of Zurn Constructors v. Castaic Lake Water Agency), reported in Lifschitz et al. (2009). Dale & D'Onofrio (2017) listed 16 court cases between 1979 and 2012 in the US, Canada, Australia, UK, and Hong Kong in which But-For analysis method was used. One of the reasons behind this acceptance is that it uses the most realistic as-built schedule in the analysis (Arditi & Pattanakitchamroon, 2006; Yang & Yin, 2009; Dale & D'Onofrio, 2017). The availability of enough as-built information to construct a relatively accurate as-built schedule is an important pre-requisites of But-For analysis (Zack, 2000). The events of each party should be identified and quantified before analysis, and a complete list of the project delay events should be created. The but-for analysis is conducted by removing the delay events of one party from the as-built schedule to determine when the project would have been completed but-for these events (Zack, 2000). But-for analysis technique is widely used in claim preparation and justification because of the limited need of time and effort; it is more suitable for small and simple projects where sophisticated analysis is not necessary. But-For analysis considers 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 party at a time from the as-built schedule to investigate its events impact on the project completion date. By subtracting the owner-caused events from the as-built schedule; the owner is said to be responsible for the difference in duration between the as-built schedule and But-For schedule and the remaining project delay is attributed to the contractor. The analysis based on subtracting contractor-caused events (contractor viewpoint) from the as-built schedule will nearly always result in an incompatible result (Vasilyeva-lyulina, 2013).

But-for technique is classified by (AACE RP 29R-03, 2011) (RP) as one of the subtractive techniques; the RP method of implementation (MIP 3.8 Modeled / Subtractive / Single Simulation) is similar to the traditional But-For analysis when applied as single window implementation. It describes the method as a modeled technique relying on a scenario simulation based on a CPM model. The simulation consists of the extraction of one party's entire events from the as-built network analysis model to determine the impact of those extracted activities on the network. The MIP 3.8 descriptively attempts to address some of the shortcomings in the common practice of the But-For as it allows stepped subtraction of events.

To explain the traditional But-For analysis, the small example of four activities adopted from literature (Bhih & Hegazy, 2019) in Figure 2.12 is used. 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 2.12c 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; thus, in the Owner viewpoint, the project delay responsibilities are (OO = 1 day, OC = 2 days, O∩C = 0 days). The same analysis from the contractor's point of view is presented in Figure 2.12d, which produces different responsibility of 3 days due to owner and zero days due to contractor; Contractor viewpoint results are (OO = 3 days, OC = 0 days, O∩C = 0 days); in both viewpoints, the concurrent delays are ignored. This disparity in results is because of the inability to identify concurrent delays by the technique.

As such, the traditional But-For analysis can allocate project delays to either the owner or the contractor, thus, it is unable to identify concurrent delays (SCL-DDP, 2002). Due to this drawback, the traditional But-For produces different results when adapting different party's viewpoint as shown in this case study. El-adaway et.al. 2014; Fawzy & El-Adaway (2012) reported that but-for analysis is highly subjective and unable to address delay concurrency. But-For analysis appears to be accurate, reliable, and easily understood; however, it is too easy to manipulate (Zack, 1999). Due to this discrepancy in results, either party can use it to support its argument (Stumpf, 2000). This discrepancy in results and inability to identify concurrent delays are due to the misconceptions involved in the common application of the technique; this was highlighted in literature by Long (2017). Moreover, but-for technique does not consider the events chronology since it removes each party's events, one party at a time, regardless of the chronology of the events' which may affect the critical path/s.



a) As-planned schedule

b) As-built schedule

c) Collapsed as-built (without owner events)

d) Collapsed as-built (without contractor events)

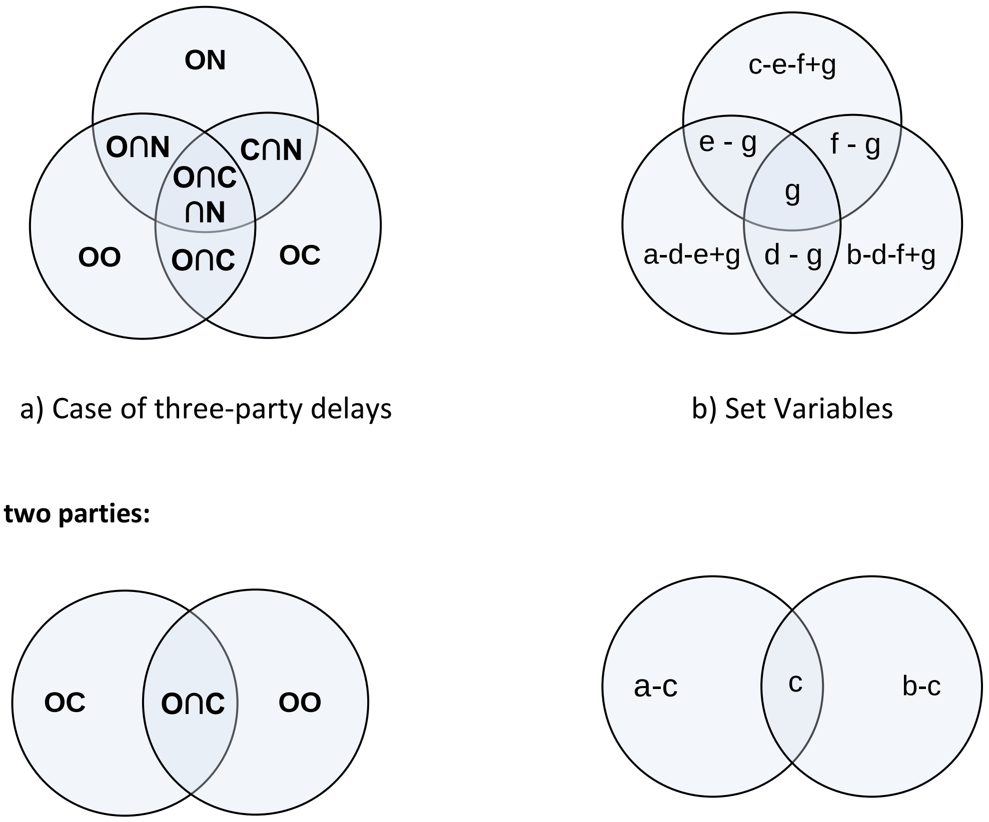
Figure 2.12 Case study schedules

2.5.3.1 Modified But-For Method (MBF)

The MBF was introduced by Mbabazi et al. (2005) to resolve the drawbacks of the traditional But-For of producing different results when different parties' viewpoints are adopted, and its inability to identify concurrent delays (Bhih & Hegazy 2019; Magdy & Georgy 2019; Zhang & Hegazy 2005). To resolve these issues, the MBF method utilizes Venn diagrams and set theorem to represent and calculate delay responsibility among three project parties (contractor C, owner O, or N for neither), as shown in Figure 2.13a. The "N" represents all factors that are outside the control of both the owner and contractor. The three "O", "C", and "N" parties can interact to produce seven combinations of responsibility as follows: OO, OC, ON, O∩C, C∩N, O∩N, and O∩C∩N as shown in Figure 2.13a. The responsibility in all segments of this Venn representation can be mathematically formulated using seven variables (a, b, c, d, e, f, and g), as shown in Figure 2.13b. The value in each segment of this representation is equal to the number of delay days of the corresponding responsibility, and all segments should add up to the total project delay days. As shown in the seven steps (rows) of Table 2.4, the MBF method subtracts seven combinations of events, one at a time, from the as-built schedule, and the resulting project duration in each case is used to calculate the values of delay responsibilities. As such, this elegant calculation provides repeatable results and considers all the intersections and contributions among the parties during the analysis of delay events. For simplicity, the MBF calculations can be reduced to the case of two project parties (owner and contractor) with only three project delay components (OO, OC, O∩C) as shown in Figure 2.13c, d, and Table 2.5

The case study of Figure 2.12 is adopted to explain MBF application and how it resolved the traditional But-For drawbacks. The 7 steps of the MBF methodology are applied to the case study and the detailed calculations are shown in Table 2.6. The MBF analysis allocated the 3 days project delays as 1 day due to the owner and 2 days owner and contractor concurrent delays (OO = 1 day, O∩C = 2 days). Similar results can be observed easily by comparing the as-planned and as-built schedules of Figure 2.12a, b, as the owner and the contractor contributed together to 2 days project delays on Days 4 and 5, while the owner caused further 1 day project delay on Day 6. As such the MBF method arrived at one result regardless of the parties' viewpoints and correctly identified the two days concurrent delays. The MBF method using the clear step-by-step analysis has resolved the two drawbacks of the traditional drawbacks. However, both techniques (But-For and MBF) are still suffering from serious drawback of insensitivity to events timing due to its single window implementation (Bhih & Hegazy, 2020a, 2021c).

Case of three parties:



a) Case of three-party delays

b) Set Variables

Case of two parties:

c) Case of two-party delays

d) Set Variables

* OO= Only Owner; OC = Only Contractor; ON = Only third party

Figure 2.13 Venn representation of project delays

Table 2.4 MBF calculations – case of three parties

Step	Delay Events Removed	Project Duration	Venn Diagram Calculations	Project Delay Responsibilities
1	O + C + N	T_1	$T_0^* – T_1$ = a+b+c-d-e-f+g	OO = $T_0 – T_2$
2	O	T_2	$T_0 – T_2$ = a-d-f+g	OC = $T_0 – T_3$
3	C	T_3	$T_0 – T_3$ = b-d-e+g	ON = $T_0 – T_4$
4	N	T_4	$T_0 – T_4$ = c-e-f+g	O∩C = $T_2 + T_3 – T_0 – T_5$
5	O+C	T_5	$T_0 – T_5$ = a+b-d-e-f+g	O∩N = $T_2 + T_4 – T_0 – T_6$
6	O+N	T_6	$T_0 – T_6$ = a+c-d-e-f+g	C∩N = $T_3 + T_4 – T_0 – T_7$
7	C+N	T_7	$T_0 – T_7$ = b+c-d-e-f+g	O∩C∩N = $T_0 + T_5 + T_6 + T_7 – T_1 – T_2 – T_3 – T_4$

T_0^* = as-built duration

Table 2.5 MBF calculations – case of two parties

Step	Delay Events Removed	Project Duration	Venn Diagram Calculations	Project Delay Responsibilities
1	O + C	T_1	$T_0^* - T_1 = a + b - c$	$OO = T_0 - T_2$
2	O	T_2	$T_0 - T_2 = a-c$	$OC = T_0 - T_3$
3	C	T_3	$T_0 - T_3 = b-c$	$O \cap C = T_2 + T_3 - T_0 - T_1$

T_0^* = as-built duration

Table 2.6 Case study MBF calculations

Step	Delay Events Removed	Project Duration	Venn Diagram Calculations	Project Delay Responsibilities
1	O + C + N	$T_1 = 8$	$T_0^* - T_1 = 3 = a+b+c-d-e-f+g$	$OO = 11 - 10 = 1$
2	O	$T_2 = 10$	$T_0 - T_2 = 1 = a-d-f-+g$	$OC = 11 - 11 = 0$
3	C	$T_3 = 11$	$T_0 - T_3 = 0 = b-d-e+g$	$ON = 11 - 11 = 0$
4	N	$T_4 = 11$	$T_0 - T_4 = 0 = c-e-f+g$	$O \cap C = 10 + 11 - 11 - 8 = 2$
5	O+C	$T_5 = 8$	$T_0 - T_5 = 3 = a+b-d-e-f+g$	$O \cap N = 10 + 11 - 11 - 10 = 0$
6	O+N	$T_6 = 10$	$T_0 - T_6 = 1 = a+c-d-e-f+g$	$C \cap N = 11 + 11 - 11 - 11 = 0$
7	C+N	$T_7 = 11$	$T_0 - T_7 = 0 = b+c-d-e-f+g$	$O \cap C \cap N = 11 + 8 + 10 + 11 - 8 - 10 - 11 - 11 = 0$

$T_0^* = 11$

2.5.4 Windows Analysis (Contemporaneous Period Analysis)

Windows analysis breaks the total project duration into digestible time increments, called windows (Weeks, Months …etc.), and examines the effects of the events attributable to each of the project parties in each window successively (Hegazy & Zhang, 2005). It adopts the as-planned schedule as its baseline and considers the periodical baseline updates. For every analysis window, the delay events of project parties are inserted into the as-planed schedule to determine its impact on the project completion date. The difference between the project completion date from the current window and the project completion date from the previous window represents the total project delay in the current window (Vasilyeva-lyulina, 2013). The responsibility for project delays is then determined by aggregating the project delays attributed to each party from all the windows. The analysis windows normally coincide with progress reporting periods or with the important project milestones; sometimes different window size is used to detect the effect of some events. The as-planned schedule may be also updated when the project parties agree to approve a new baseline for the project.

Windows technique is classified by (AACE RP 29R-03, 2011) recommended practice (RP) as one of the additive techniques; the RP method of implementation (MIP 3.7 Modeled / Additive / Multiple Base) describes windows implementation. The RP describes the method as a modeled technique since it relies on a simulation of a scenario based on a CPM model. The simulation consists of the addition of delay events into the baseline schedule to determine the hypothetical impact of those inserted activities on the network. The additive simulation is performed on multiple baselines representing the baseline updates.

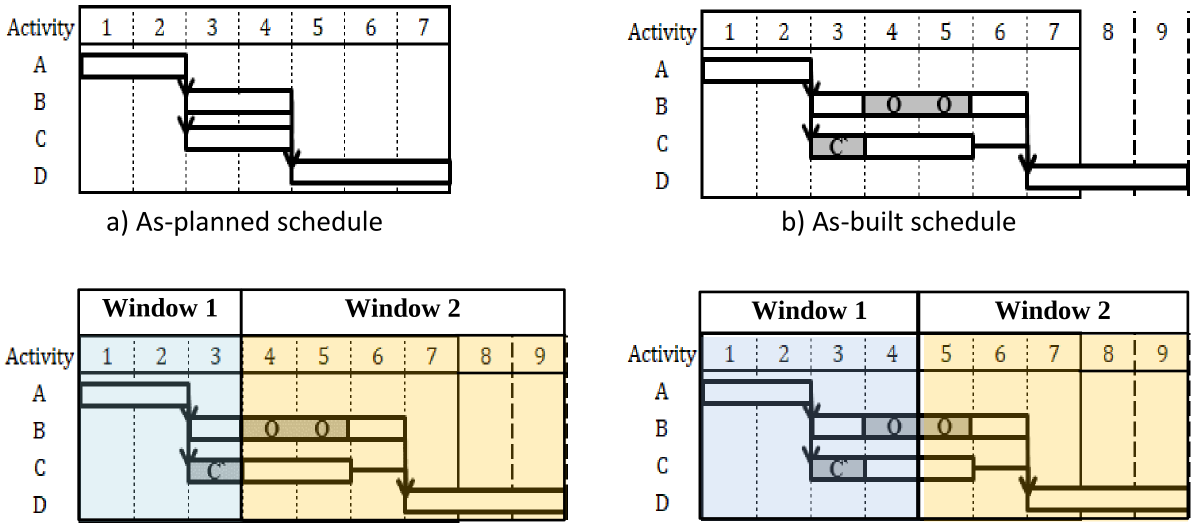
Windows analysis is one of the widely used and accepted techniques and proved, in the literature, that it gives reliable and equitable results (Menesi, 2007). On the other hand, windows analysis, compared to But-For analysis, is still time-consuming and requires a large amount of data, thus it may not be appropriate for projects that lack good progress records or when there is not enough time and budget allocated for the analysis. It is distinguished from but-for analysis by its consideration of the dynamic nature of the project's critical paths. While the use of multiple windows enables this method to capture some critical path dynamics, the accuracy of the analysis depends largely on window size.

To illustrate the Windows analysis sensitivity to window size, a small example in Figure 2.14a, b (adapted from Hegazy & Zhang, 2005) is used. The project had a 7-day planned duration; experienced delay events during construction which resulted in 2 days behind schedule (9 days as-built duration). To investigate the impact of window size, two window arrangements are shown in Figures 2.14c and d, respectively. In Figure 2.14c, the project is divided into two windows (Window 1:

Days 1-3; and Window 2: Days 4-9). Alternatively, the window arrangement in Figure 2.14d uses two different windows (Window 1: Days 1-4; and Window 2: Days 5-9). In Figure 2.14c, the first window shows 1 day contractor event that leads to 1 day project delay (from 7 to 8), thus, this delay is the contractor’s responsibility. Moving to Window 2, the 2 days owner events caused 1 day further project delay, which is the owner’s responsibility. The final analysis result is: 1 day contractor, plus 1 day owner (OO = 1 day, OC = 1day). Alternatively, the analysis result in Figure 2.14d with the different window sizes produces: 1 day due to owner and contractor together, plus one day owner only (OO = 1 day, O∩C = 1 day), which is different from the analysis result from Figure 2.14c.

2.5.4.1 Daily Windows Analysis (DWA)

To resolve the sensitivity to window size, Hegazy & Zhang, (2005) introduced a more accurate Daily Windows Analysis (DWA) method with a fixed window size of one day. Using one day fixed window size allow the analyst to consider the day-by-day events and fluctuation in critical paths, increase the analysis accuracy, and reduces the possibility of manipulations (Hegazy & Zhang, 2005). Hegazy & Menesi (2008) improved the daily windows delay analysis to consider multiple baseline updates and resource-leveling. The consideration of rework in the daily windows analysis was proposed by (Hegazy et al., 2011) One of the drawbacks of this technique is that it needs more effort and time, however, it was developed in a computer-oriented fashion to solve this problem. The technique needs also a detailed progress record on a daily basis. The daily windows analysis technique overcomes most of the drawbacks of the traditional windows analysis and other conventional techniques, but it still lacks ability to consider complex cases that involve simultaneous delay and acceleration events or situations where the delay events can cause more than one day impact on the completion date.



a) As-planned schedule

b) As-built schedule

c) Two analysis windows: 1-3, 4-9

d) Two analysis windows: 1-4, 5-9

Figure 2.14 Effect of window size on Windows Analysis results

2.5.5 Other Techniques

The mathematical formulation for the delay analysis was introduced by Jonathan et al. (2001); they also proposed activity-based-equation for computing delay events and assessing their contributions to the project delay. The calculation equations use the as-planned schedule as the basis; the equations are valid only for the finish-to-start relationship and are not applicable for other relationships.

Fuzzy logic model for estimating delay duration was proposed by Oliveros & Fayek (2005) to improve delay analysis. The model integrates daily site activity progress and delays, with a schedule updating and forecasting system to assist in estimating delay events impact on a project completion date. The model uses fuzzy logic for estimating the impact of activity delays and recalculating the project schedule. The model is partially computerized; to efficiently analyze the information that results from daily site records.

Al-Gahtani & Mohan (2007) presented a delay analysis technique using total float management. The technique primarily focuses on the ownership of total float. The ownership of noncritical activities' total float (TF) is to be distributed at the beginning of the project. As the project makes progress, delays, accelerations, change orders and suspensions occur and the TFs of the activities change. The party responsible for each delay event is assigned responsibility for the changes in network total floats and is credited or discredited TF entitlement. (C. Tsai, Wang, & Chang, 2015) explained that the length of each activity time has been taken as the basis for apportioning floats in most past methods. However, an activity with a longer duration may not directly mean more delayed risk inherent in that activity. In their research, they introduced a new approach for float allocation based on the criticality index.

A new customizable schedule analysis technique called FLORA that simultaneously captures the dynamics of float, logic, and resource allocation in its analyses was presented by Nguyen & Ibbs (2008). FLORA analyzes the direct impact of delay events and also its “secondary” effect since it tacks into consideration resource allocation. The analysis process follows ten flexible and customizable rules.

Yang & Yin (2009) developed a new delay analysis technique called the isolated collapsed but-for (ICBF) method. The ICBF method requires as-planned and as-built schedules as well as identified key delay events. The ICBF method uses the concept of isolated delay type method, but, like But-For analysis, it starts with the as-built schedule which reflects actual project progress data. Therefore, the ICBF maintains the advantages associated with the But-For and the isolated delay type methods, and overcomes their persistent drawbacks; the ICBF can be considered as a combination of them. (Golanaraghi & Alkass, 2012) modified the isolated delay type technique to account for concurrent delays.

Delay analysis method that can address the concurrent delays considering the nonlinear production rates of activities was developed by Lee & Diekmann (2011). The basic principle of the delay analysis techniques, window or But-For analysis, was used and modified to incorporate the varying rate of production into the delay analysis process.

Delay analysis is a tedious, time-consuming, and error-prone process, however, most of the techniques follow systematic steps. To build on these facts, many researchers in the literature introduced algorithms for some of the techniques and developed delay analysis computer prototypes that can handle the analysis process and save money and time.

Alkass et al. (1995) developed computer implementation of the windows delay analysis technique using commercial scheduling software. Hegazy & Zhang (2005) developed a computer prototype that can implement daily windows analysis and made it available online as an educational version. Mbabazi (2004) developed a computerized decision support tool that is readily usable for accurate claim analysis. The developed system can implement both the daily windows analysis technique and the modified but-for analysis technique. Menesi (2007) introduced a prototype that can handle the daily windows delay analysis under multiple baseline updates and considering resource-leveling. Yang & Tsai (2011) used Microsoft Visual Basic for Applications (VBA) to develop an Excel-based program for rapid delay analysis using the isolated collapsed but-for ICBF technique; the ICBF was developed by Yang & Yin (2009) and briefly described earlier in this chapter. The computer-based delay analysis technique was proposed by Tsai et al. (2013) utilizing the flow of information analysis. Al-Gahtani et al., (2016) introduced web-based software called total float management (TFM) software for delay analysis that depends on day-by-day analysis, and gives more accurate analysis over other techniques; the software has the ability to import schedule data from Primavera P6 and Microsoft Project.

2.6 Recommended Practices and Standards for Forensic Schedule Analysis

The area of forensic delay analysis has received more attention among researchers and practitioners; many studies have been published during the past few decades. In the past two decades, several professional bodies have shown their interest in forensic schedule analysis by establishing protocols, recommended practices, and proposed standards, aiming at promoting clearer contract conditions regarding float ownership, concurrent delays, and analysis methods to determine entitlement to compensation and extension of time (Braimah, 2013). Three internationally recognized professional societies issued their proposed guidance documents (Livengood 2017); briefly introduced in the following subsections:

2.6.1 Delay and Disruption Protocol, SCL, UK

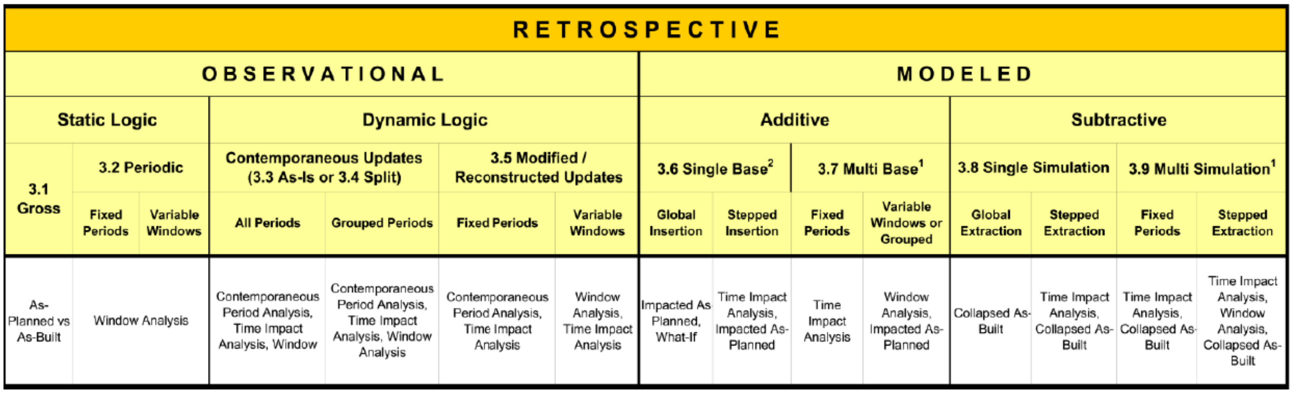
The delay and disruption protocol was firstly issued by the Society of Construction Law in October 2002 (SCL-DDP, 2002), it has been amended by Rider 1 in July 2015. In February 2017, the 2nd edition of the Protocol was published; the 2nd edition supersedes both the 1st edition and Rider 1 to the 1st edition. The protocol was prepared to help in determining extensions of time and compensation for delay and disruption. The protocol distinguishes between delay and disruption. Delay is concerned with the time of activities taking longer than planned and causing delay to the completion of the project. Disruption is concerned with disturbance, hindrance, or interruption to the contractor's working methods causing lower productivity or efficiency. Disruption is concerned with an analysis of the productivity of activity, regardless of whether this activity is on the critical path(SCL-DDP, 2002).

As explained in its introduction, the protocol aims to be consistent with the good practice, but not to be the benchmark for good practice in the construction industry. It is intended to be a balanced document to reflect equally the interest of all parties. It aims to provide guidance to all parties when dealing with delay and disruption matters and not to be a statement of the law. It recognizes that transparency of information and methodology is central to both dispute prevention and dispute resolution (SCL-DDP, 2002). The body of the protocol is structured in four parts; part 1 introduces 22 core concepts covered by the protocol, part 2 is dedicated to explain the concepts of delay, disruption, and acceleration, part 3 sets the guidance on each of the 22 core concepts, and part 4 discusses guidance on other financial issues related to delay and disruption.

2.6.2 Forensic Schedule Analysis Recommended Practice No. 29R-03, AACE, USA

In June 2007, the Association of the Advancement of Cost Engineering International ("AACEI") published its Recommended Practice (RP) No. 29R-03 (RP). The RP was revised twice later; the 1st revision was published in June 2009, the 2nd and the latest revision in April 2011. The stated objective of the RP 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. It is not intended to be a prescriptive document that can be applied without exception" (AACE RP 29R-03, 2011).

The RP comes in five chapters and two appendices. The first chapter explains the scope of the RP and the taxonomy used to categorize the analysis techniques in addition to presenting the fundamentals and principles. The approaches used to validate data sources and identify and quantify delay events are introduced in the second chapter. Considering that there are different practice methodologies of schedule analysis techniques, the RP recommends best practices for forensic schedule analysis techniques through introducing nine methods of analysis implementations (Figure 2.15). It attempts to categorize the delay analysis methods used in practice under the nine methods of implementation; the implementation methods are described in chapter three. Analysis results evaluation is the focus of chapter four. Chapter five discusses the factors affecting the selection of a particular method of implantation.



RETROSPECTIVE														
OBSERVATIONAL							MODELED							
Static Logic			Dynamic Logic				Additive				Subtractive			
3.1 Gross	3.2 Periodic		Contemporaneous Updates (3.3 As-Is or 3.4 Split)		3.5 Modified / Reconstructed Updates		3.6 Single Base[2]		3.7 Multi Base[1]		3.8 Single Simulation		3.9 Multi Simulation[1]	
	Fixed Periods	Variable Windows	All Periods	Grouped Periods	Fixed Periods	Variable Windows	Global Insertion	Stepped Insertion	Fixed Periods	Variable Windows or Grouped	Global Extraction	Stepped Extraction	Fixed Periods	Stepped Extraction
As-Planned vs As-Built	Window Analysis		Contemporaneous Period Analysis, Time Impact Analysis, Window	Contemporaneous Period Analysis, Time Impact Analysis, Window Analysis	Contemporaneous Period Analysis, Time Impact Analysis	Window Analysis, Time Impact Analysis	Impacted As Planned, What-If	Time Impact Analysis, Impacted As-Planned	Time Impact Analysis	Window Analysis, Impacted As-Planned	Collapsed As-Built	Time Impact Analysis, Collapsed As-Built	Time Impact Analysis, Collapsed As-Built	Time Impact Analysis, Window Analysis, Collapsed As-Built

Figure 2.15 AACE methods of implementation for forensic schedule analysis

2.6.3 Schedule Delay Analysis (ASCE 67-17), ASCE, USA

Recently, in 2017, the American Society of Civil Engineers ASCE published their new proposed standard entitled Schedule Delay Analysis in accordance with the American National Standards Institute ANSI. The standard is said to set industry guidelines for the use of schedule delay analysis methods. It introduces 35 guidelines of the best engineering schedule delay analysis principles which enable apportioning of responsibility for delay to project milestones and the project completion date and calculating delay damages or liquidated damages. The guidelines are categorized in the following eight categories: critical path, float, early completion, the chronology of delay events, concurrent delay, responsibility for delay, changing schedules after the fact, and acceleration (ASCE 67-17, 2017). The standard is prepared to be used in consistence with the practitioner’s professional experience and knowledge, along with project facts and contract requirements. The proposed standard does not emphasize any specific delay analysis method; rather it provides general guidelines that are applicable in conjunction with any methodology.

2.7 Summary of Knowledge Gaps

Despite the increasing research efforts in the area of delay analysis, specifically, delay analysis techniques, shortcomings and limitations still exist. In this chapter, the literature review was introduced to identify the research gaps and potential areas of improvement. The conclusion of this review can be summarized in the following:

- The current state of the delay analysis techniques has shortcomings and persistent issues that need to be addressed. But-For analysis is one of the widely used and accepted techniques despite its known drawbacks; the reason behind its acceptance is that it has a simple methodology and needs less time and effort. But-For analysis is unable to identify concurrent delays and produces conflicting results when adopting different party's viewpoints, thus, it can suit limited situations where concurrency does not exist. Furthermore, due to its single window implementation, the traditional But-For, lacks the ability to track critical path(s) fluctuations, and to respect event chronology. Improving But-For methodology will help reconciling the opposing viewpoints of the owner and the contractor, and hence will minimize delay disputes.
- Modified But-For method (MBF) has resolved some of the traditional But-For drawbacks by enabling it to identify concurrent delays. The MBF has clear step-by-step methodology to allocate schedule delays in the general practical case of three project parties (Owner, Contractor, Third party), it considers the seven components of concurrent and non-concurrent delays due to the interactions among the three parties. However, the MBF still

lacks sensitivity to event chronology and critical path(s) fluctuations due to its single window implementations. Furthermore, the MBF was developed to allocate only schedule delays overlooking the common case of schedule accelerations. Addressing the MBF drawbacks and extending its methodology to allocate both schedule delays and accelerations will provide a powerful alternative But-For analysis that is able to identify concurrent delays, and produce consistent results irrespective the adopted viewpoint.

- Windows Analysis (WA) is one of the accurate techniques; however, it suffers from sensitivity to window size. This issue was addressed in literature by Daily Windows Analysis (DWA). The DWA has the ability to track event chronology and critical path(s) on daily basis. Despite this accuracy, DWA is still unable to apportion concurrency in complex situations when delay events have more than one day effect on the project completion date, especially when simultaneous delay and acceleration events are involved. The integration of DWA with the But-For analysis to address its drawbacks will resolve the existing drawbacks of both techniques.

Chapter 3

Proposed Framework and Traditional But-For Enhancement[1]

3.1 Introduction

This chapter starts with a brief description of the main components of the proposed delay analysis framework to develop a generic delay analysis technique able to allocate concurrent and non-concurrent delays and accelerations considering event chronology and multiple baseline updates. Then, it presents the first step of the framework of enhancing traditional But-For analysis. The proposed enhancements to the traditional But-For method includes: (1) resolving the misconceptions associated with the common application of the traditional But-For analysis; (2) improving its concurrency assessment by introducing Multi-window But-For analysis; (3) extending the application of the enhanced But-For analysis to the general case of three project parties. The proposed enhancements are illustrated and validated using several case studies.

3.2 Proposed Framework Components

As discussed in chapter one, this research aims to develop a comprehensive delay analysis framework that can overcome the common delay analysis drawbacks, accurately assess concurrent delays and accelerations, and produce repeatable results considering event chronology. The proposed framework (figure 3.1) has three main components: (1) Improved traditional But-For analysis; (2) Enhanced modified But-For Analysis (EMBF); and (3) Enhanced Daily Windows Analysis (EDWA). The three components are discussed in detail in the next subsections.

[1] Parts of this chapter was published in the following manuscript:
- Bhih, M., & Hegazy, T. (2020). Improving Concurrency Assessment and Resolving Misconceptions about the But-For Delay Analysis Technique. *ASCE-Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, *12*(2), 1–10. https://doi.org/10.1061/(ASCE)LA.1943-4170.0000378

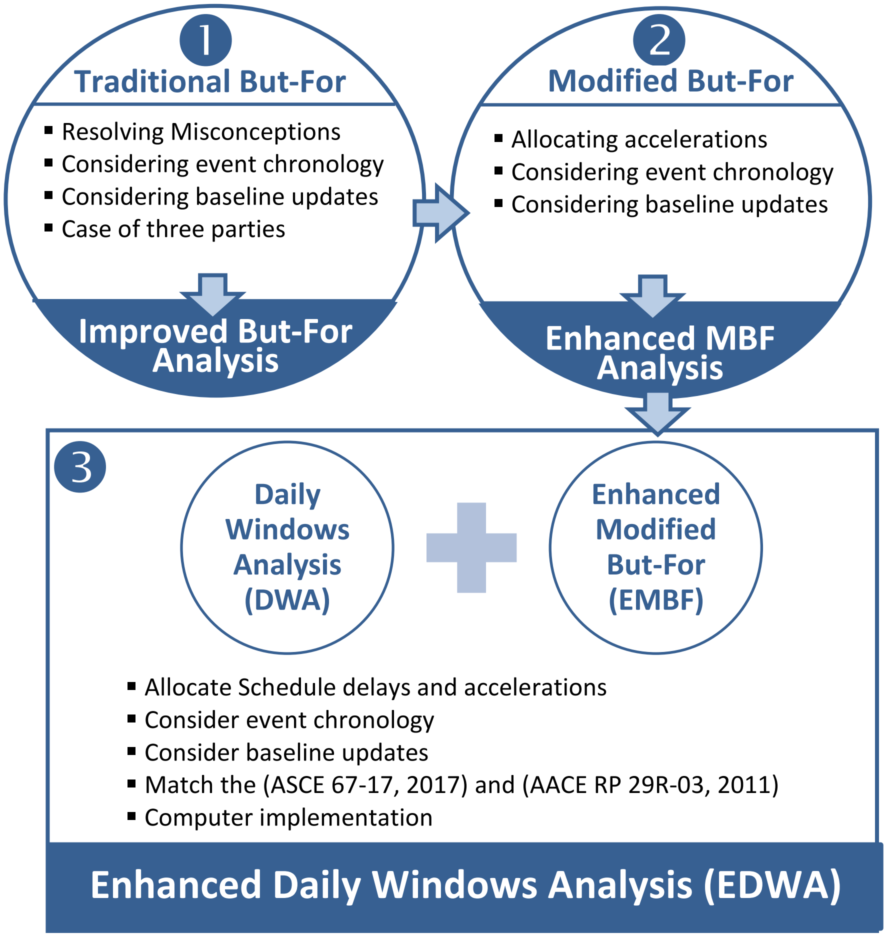

1
Traditional But-For
Resolving Misconceptions
Considering event chronology
Considering baseline updates
Case of three parties
Improved But-For Analysis
2
Modified But-For
Allocating accelerations
Considering event chronology
Considering baseline updates
Enhanced MBF Analysis
3
Daily Windows Analysis (DWA)
Enhanced Modified But-For (EMBF)
Allocate Schedule delays and accelerations
Consider event chronology
Consider baseline updates
Match the (ASCE 67-17, 2017) and (AACE RP 29R-03, 2011)
Computer implementation
Enhanced Daily Windows Analysis (EDWA)

Figure 3.1 Components of the Proposed Forensic Analysis Framework

3.2.1 Component 1: Improved But-For Analysis

Traditional But-For analysis is an important technique as it is widely used and accepted in practice (Dale & D'Onofrio, 2017), this is because its simple methodology and the use of the most realistic as-built schedule in the analysis (Arditi & Pattanakitchamroon, 2006; Yang & Yin, 2009; Dale & D'Onofrio, 2017). Among the three industry guidance documents, only the AACE RP provides guidelines for the delay analysis methods. The traditional But-For analysis has a resemblance to two of those methods (MIP 3.8 and MIP 3.9), which are descriptive guidelines rather than step-by-step methods. The AACE MIP 3.8 and MIP 3.9 methods represent guidelines for the subtractive analysis techniques in practice such as the traditional But-For analysis including both the owner and contractor points of views. The subtractive techniques use the as-built schedule as the analysis base and subtract delay events from it to assess the events' effect. AACE MIP 3.8 and 3.9 methods allow subtracting delay events altogether (global subtraction) or piecemeal in periods moving backward on the project (stepped subtraction) (Dale & D'Onofrio, 2010). The difference between the two methods is that method 3.9 considers the multiple baseline updates. In their minimum implementation, both methods require identifying and quantifying concurrent delays and respecting event chronology in the analysis.

But-For technique has been criticized in literature for several drawbacks as highlighted in chapter 2. Those drawbacks can be summarized as follows:

- It produces inconsistent results when used by different project parties', thus, it is easy to manipulate (Braimah, 2013; Fawzy & El-Adaway, 2012; Magdy & Georgy, 2019; Zack, 2000); this is due to the misconceptions involved in the common But-For application. Moreover, It allocates schedule delays only among the owner and contractor and not a third party;
- It is unable to identify concurrent delays (Bhih & Hegazy, 2020b; Mbabazi et al. 2005);
- it is insensitive to event chronology (Bhih & Hegazy, 2020b); this is due to its single window implementation; and
- it is insensitive to the fluctuations of the critical path(s) during construction(Bhih & Hegazy, 2020b).

As the first step towards a more accurate delay analysis technique the But-For analysis drawbacks need to be addressed. Therefore, in the first component of the proposed framework, the following improvements need to be introduced:

- Resolve the existing misconceptions in the common But-For implementation.

- Improve the But-For concurrency assessment by considering event chronology in the analysis.
- Enable the But-For analysis to consider baseline changes during the construction.
- Extend the But-For methodology to the general case of three project parties.

3.2.2 Component 2: Enhanced Modified But-For Analysis (EMBF)

The second component of the proposed framework investigates the improvements introduced in the literature to the traditional But-For analysis, that is the Modified But-For analysis introduced by Mbabazi et al. (2005). The modified But-For method introduces alternative systematic But-For analysis to apportion the delay responsibility among project parties utilizing Venn representation and set theorem. This methodology has an elegant calculation to apportion delays among three project parties (Owner O, Contractor C, Third party N) and produce results for the possible seven components of project delays (OO, OC, ON, O∩C, O∩N, C∩N, O∩C∩N). The MBF resolved the first two drawbacks of the original techniques mentioned in the previous section; thus producing consistent results irrespective of the adopted viewpoint, and accurately identifying concurrent delays. However, it did not resolve the drawbacks of considering event chronology and critical path(s) fluctuations. Moreover, the MBF methodology was developed only to allocate schedule delays and not accelerations. As a good candidate for the integration with the daily windows analysis to produce a more accurate delay analysis tool, the MBF method is targeted by several enhancements that adhere to recent delay analysis standards of the ASCE (ASCE 67-17, 2017) and the AACEI recommended practice (AACE RP 29R-03, 2011). In the second step towards this goal, the targeted improvements are as follows:

- Extend the MBF methodology to allocate both schedule delays and accelerations.
- Improve the MBF to consider event chronology and critical path fluctuations.

- Enable the MBF methodology to consider baseline changes during construction.

3.2.3 Component 3: Enhanced Daily Windows Analysis (EDWA)

The third and last component of the proposed framework introduces an Enhanced Daily Windows Analysis (EDWA) that integrates the most enhanced versions of windows analysis (Daily Windows Analysis (DWA)) and But-For analysis (Enhanced Modified But-For method (EMBF)). In practice, Windows Analysis and the But-For analysis methods have been the most widely used by practitioners and accepted by courts and arbitration boards (Stumpf, 2000; Yang & Kao, 2009). Despite their wide use and acceptability, both methods suffered from persistent drawbacks (Bhih & Hegazy, 2021b). The Daily Windows Analysis (DWA) was introduced in literature by Hegazy & Zhang (2005) using a fixed window size of one day to address window analysis sensitivity to window size. On the other hand, Bhih & Hegazy 2021a, 2021c (component 2 of the framework) introduced an Enhanced MBF method to allocate both delays and accelerations and better respect event chronology. Nevertheless, both techniques (DWA and EMBF) are still unable to handle complex situations of simultaneous events that cause multi-day project delay, particularly when schedule accelerations are involved. The proposed integrated technique aimed to benefit from the advantages of its parent techniques, and to overcome its drawbacks. Component 3 targets introducing integrated technique with the following features:

- Overcome its parent techniques persistent drawbacks;
- Allocates both concurrent and non-concurrent delays and accelerations even in complex situations of simultaneous delay/acceleration events of multi-day impact on project completion date;
- Considers event chronology, baseline updates, and tracks critical path(s) fluctuations on a day-by-day basis;

- Produce fair and consistent results that match the recent delay analysis standards of the ASCE (ASCE 67-17, 2017) and the AACEI recommended practice (AACE RP 29R-03, 2011); and
- Has structured step-by-step methodology to facilitate its computer implantation.

3.3 Proposed Enhancements to Traditional But-For Analysis

In the first component of the framework, the traditional But-For analysis is investigated and targeted by several enhancements to match the industry practice requirements. The current state of the traditional But-For analysis was compared with the AACE similar methods (MIP 3.8 and MIP 3.9) as highlighted in Table 3.1. Based on this comparison, improvements to But-For analysis are proposed (last column of Table 3.1) to match AACE guidelines. The other two industry guidelines, (SCL-DDP, 2017) and (ASCE 67-17, 2017), has no specific analysis methods; it introduces guidelines for schedule delay analysis to be applied with any schedule analysis technique. As shown, the key objective is to clarify the misconceptions associated with the use of the traditional But-For method, improve its concurrency assessment, and translate the introduced enhancements into a clear step-by-step analysis methodology that matches the general practice guides.

To achieve the targeted improvements to the But-For analysis, two main enhancements are introduced to the technique: correcting the existing misinterpretations of results, and improving its concurrency assessment. This chapter introduces significant improvements to the traditional But-For analysis by correcting the misinterpretation of results which enables the technique to identify concurrent delays and thus reconcile the opposing viewpoints of both project parties. Furthermore, the multi-windows But-For analysis is introduced to resolve the insensitivity of the technique to the event chronology and critical path fluctuations and to accommodate cases with baseline updates. The step-by-step methodology is extended to the general practical case of three-party delays.

Table 3.1 Comparison between the traditional and related But-For methods

Feature	Traditional But-For	AACE RP 29R-03 MIP 3.8 and MIP 3.9	Targeted Improvements
Clear steps to apply the method	×	Descriptive Methods	Detailed Method
Results not dependent on viewpoint	×	Descriptive guidelines	✓
Can use multiple analysis-windows	×	✓	✓
Concurrency assessment	×	Descriptive guidelines	Detailed
Respect event chronology	×	Partially*	Detailed
Multiple baseline updates	×	Descriptive guidelines	Detailed
Analysis of 3-party delays	×	Descriptive guidelines	Detailed

* From one window to another but not within each window

3.4 Resolving the Misinterpretation of Traditional But-For Analysis

The small case study of Figure 2.12 is used to explain the common misinterpretation of the traditional But-For results. The case study has been adopted from the literature (Mbabazi et al., 2005) with minor changes to suit the purpose. Both the as-planned and as-built schedules are shown with delay events by the owner and the contractor shown on the as-built schedule. The project as-planned duration was 8 days; while the actual completion exhibited a 3-day project delay, making the project as-built duration 11 days.

3.4.1 Owner's point of view

Removing the owner's events from the as-built schedule produces the schedule in Figure 2.12c, with 10 days duration (without the owner’s events). Thus, the as-built schedule was reduced 1 day after removing the owner's events; so in the owner’s point of view, the owner is responsible for 1 day project delay, and the remaining balance of the project delay (2 days) is due to the contractor events. The following steps summarize the owner's point of view in But-For analysis:

(1) Total project delay = 3 days

(2) As-built duration = 11 days

(3) As-built without "O" events = 10 days

(4) Resulting owner O responsibility = (2) – (3) **= 1 day**

(5) Contractor C responsibility = (1) – (4) **= 2 days**

(6) Final result: **O = 1 day, C = 2 days.**

3.4.2 Contractor's point of view

Removing the contractor's events from the as-built schedule produces the schedule in Figure 2.12d, with 11 days duration. Thus, the as-built schedule did not collapse after removing the contractor's events; so in the contractor's point of view, only the owner is responsible for all the 3 days project delay. The following steps summarize the contractor's point of view in But-For analysis, and Table 2 shows the results of the contradicting points of view:

(1) Total project delay = 3 days

(2) As-built duration = 11 days

(3) As-built without "C" events = 11 days

(4) Resulting Contractor C responsibility = (2) – (3) **= 0 days**

(5) Owner O responsibility = (1) – (4) **= 3 days**

(6) Final result: **O = 3 day, C = 0 days**

Table 3.2 traditional But-For results of Scenario 1 under common misinterpretation

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

3.4.3 Correct But-For Result Interpretation

To easily explain the common misinterpretation of results, the Venn representation proposed by (Mbabazi et al., 2005) is utilized. Figure 3.2a shows the graphic representation of total project delay (3 days) caused by "only owner OO", "only contractor OC", and "both concurrently O∩C". Figure 3.2b shows the case of removing "O" events as done in Figure 2.12c, which results in a responsibility of 1 day assigned to OO (indicated in Figure 3.2b). The misinterpretation is 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. Thus, in the common interpretation of But-For results, each party ignores its contribution to concurrent delays (O∩C).

Similarly, removing the "contractor C" events, as done in Figure 2.12d, results in an OC responsibility of zero, as indicated in Figure 3.5c. 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 in Figure 3.5d. Based on this discussion, the following step-by-step calculations represent the correct and repeatable But-For analysis when owner and contractor events exist on the schedule.

(1) Total project delay (OUC) = 3 days

(2) As-built duration = 11 days

(3) As-built without O events = 10 days

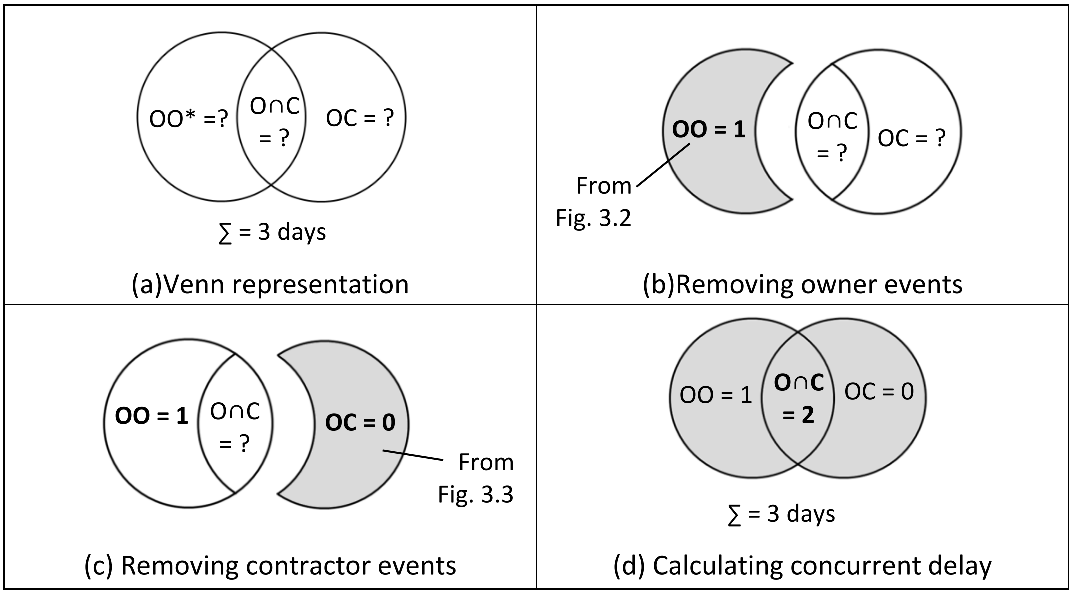
(4) OO responsibility = (2) – (3) **= 1 day**

(5) As-built without C events = 11 days

(6) OC responsibility = (2) – (5) **= 0 days**

(7) O∩C responsibility = (1) – (4) – (6) = 2 days

(8) Final result: **OO = 1 day, OC = 0 days, O∩C = 2 days**



(a)Venn representation

(b)Removing owner events

(c) Removing contractor events

(d) Calculating concurrent delay

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

Figure 3.2 Venn representation of project delays

It is important to note that by visually examining the as-built schedule of Figure 2.12b, the concurrent part of the project delays (O∩C = 2 days) is determined to be truly concurrent delays since the two delay events occurred at the same time (simultaneous events). An example of the But-For analysis good practice in the literature that does not involve such misconceptions is the one presented in (Long, 2017).

3.5 Improving Concurrency Assessment in But-For Analysis

By using the correct interpretation of results, as done in the previous section, the technique still lacks the ability to distinguish between the true concurrency and the concurrent effects as explained in section 2.2.3.2 of this thesis. To demonstrate this issue, the as-built schedule of the case study has been slightly modified in Scenario 2, as shown in Figure 3.3. The difference in this situation is that the owner and contractor events are not occurring simultaneously, thus, true-concurrency did not exist. For the reasons explained earlier in this thesis, literal concurrency is adopted in the analysis.

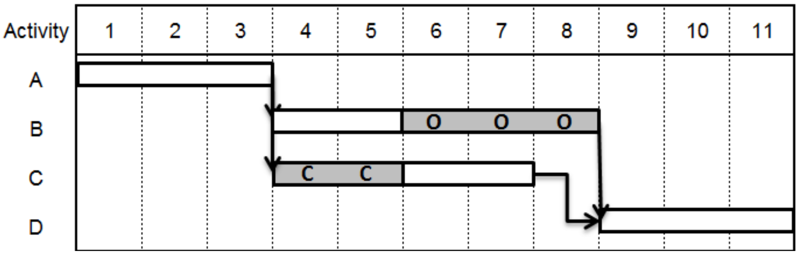


Figure 3.3 As-built schedule (Scenario 2)

Applying the correct But-For analysis to this new scenario is as follows:

(1) Total project delay (OUC) = 3 days

(2) As-built duration = 11 days

(3) As-built without O events = 10 days

(4) OO responsibility = (2) – (3) = 1 day

(5) As-built without C events = 11 days

(6) OC responsibility = (2) – (5) = 0 days

(7) O∩C responsibility = (1) – (4) – (6) = 2 days

(8) Final result: **OO = 1 day, OC = 0 days, O∩C = 2 days**

These results are identical to those of Scenario 1 (OO 1 day, O∩C 2 days), despite the difference in the timing of each party's events. If the concept of the offsetting delay was not considered in the analysis, the 2 days of O∩C concurrent delay result is clearly wrong as the as-built schedule in this scenario shows that the two parties' events are not simultaneous. As such, the But-For method, in this case, identified a "Concurrent Effect". Such a result was obtained because of the common single-window analysis, which makes the But-For technique unable to respect the chronological order of events. This is a common drawback of But-For analysis (Dale & D'Onofrio, 2017). To determine the correct results considering the chronological order of events let's first look at the as-planned critical paths (ABD and ACD) (Figure 2.12a). The contractor events on Days 4 and 5 caused 2 days delay to the project completion date (Days 9 and 10), this delay has made (ACD) the longest path and created 2 days free float for activity B before it may hit its successor (activity D); however, since the contractor delay is non-excusable by definition, no time extension will be granted for the contractor and activity B will still have zero total float.. The analysis now considers the subsequent 3 days owner events on activity B, which will first consume the 2 days free float created earlier by the contractor event before it hits activity D and causes further 1-day project delay. Based on this analysis the responsibility is (OO = 1 day, OC = 2 days, O∩C = 0 days).

Considering the offsetting delay according to the viewpoint of (ASCE 67-17, 2017) standard; since contractor delays are non-excusable, by definition, thus no time extension will be granted to the contractor, and activity B will remain critical with a total float value of zero, although it has 2 days free float. This viewpoint will produce different responsibilities of (OO = 1 day, OC = 0 days, O∩C = 2 days) as the owner-caused delay will first offset the earlier 2 days contractor-caused delay and then will cause further 1-day project delay of the owner responsibility. Although this is a considerable viewpoint, offsetting delay was not considered in this analysis because (AACE RP 29R-03, 2011) and in particular, MIP 3.8 and 3.9 were used as the benchmark to improve the traditional But-For analysis. Thus the results without considering offsetting delays are selected in this case.

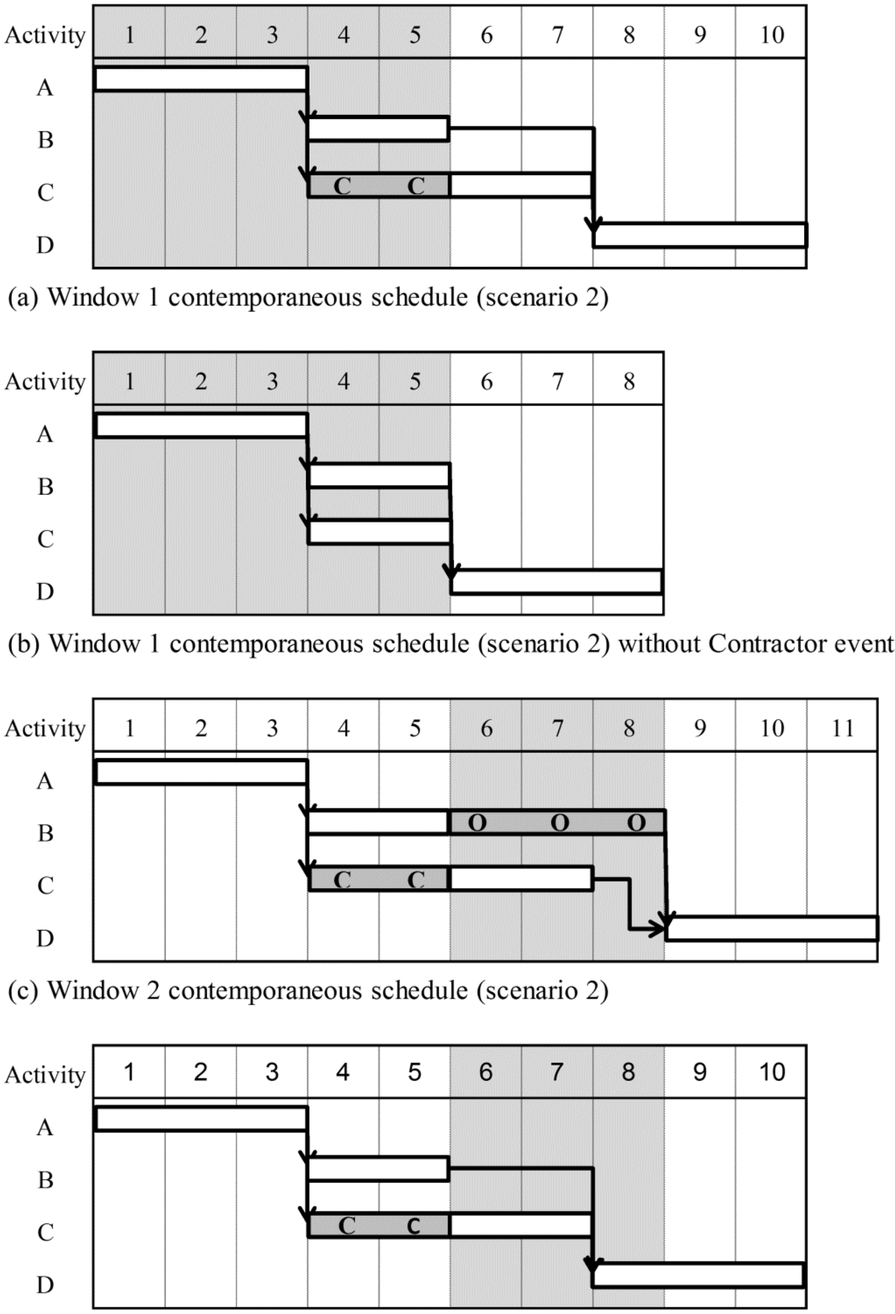
3.5.1 Improved Concurrency Assessment Using Multiple-Window But-For Analysis

To overcome the above-mentioned drawback, an additional step of testing for concurrency needs to be performed on the analysis results. In this step, the timing of critical events and the critical paths are investigated to ensure that event simultaneity exists; otherwise, the analysis results should be modified to indicate non-concurrent delays (i.e., contractor delay in this case). True concurrency exists if the delay events happened simultaneously and its effects (project delays) arose concurrently. Thus, if the analysis results show concurrent delays, the causing delay events should be investigated to ensure that they are simultaneous, and accordingly correct the analysis results if necessary. This concurrency test aims to differentiate between truly concurrent delays and concurrent effects. Because the concurrency test is not a simple task for large projects, a more accurate delay analysis method becomes necessary (e.g., the daily windows method of (Hegazy & Zhang 2005)).

However, it is possible to improve the But-For concurrency assessment by applying the correct analysis with multiple windows to increase the analysis resolution. The But-For multi-window analysis was descriptively mentioned in the AACE RP 29R-03 (2011); an example of this kind of analysis is presented in Long (2017). The window size must be carefully selected so that both party's events are on separate windows, unless the events are simultaneous. For example, in Scenario 2, using two windows of analysis (Day 1 to Day 5) and (Day 6 to Day 8), as shown in Figure 3.4, will ensure that the delay events of the two parties are on separate windows and thus it will be assessed separately.

Updating the as-planned schedule up to Day 5 produces the contemporaneous schedule in Figure 7a with a duration of 10 days. Applying the correct But-For analysis to this window is as follows:

(1) Total project delay (OUC) of this window = 2 days
(2) As-built duration (Figure 3.4a) this window = 10 days
(3) As-built without O events of this window = 10 days
(4) OO responsibility = (2) – (3) = 0 days
(5) As-built without C events (Figure 3.4b) = 8 days
(6) OC responsibility = (2) – (5) = 2 days
(7) O∩C responsibility = (1) – (4) – (6) = 0 days
(8) Final result: **OO = 0 days, OC = 2 days, O∩C = 0 days**



(a) Window 1 contemporaneous schedule (scenario 2)

(b) Window 1 contemporaneous schedule (scenario 2) without Contractor events

(c) Window 2 contemporaneous schedule (scenario 2)

(d) Window 2 contemporaneous schedule (scenario 2) without owner events

Figure 3.4 Applying But-For analysis with multiple windows (Scenario 2)

Afterwards, updating the as-planned schedule up to day 8 produces the contemporaneous schedule in Figure 3.4c with duration of 11 days. Applying the correct But-For analysis to this window is as follows:

(1) Total project delay beyond previous window = 1 day

(2) As-built duration (Figure 3.4c) this window = 11 days

(3) As-built without O events (Figure 3.4d) = 10 days

(4) OO responsibility = (2) – (3) = 1 day

(5) As-built without C events = 11 days

(6) OC responsibility = (2) – (5) = 0 days

(7) O∩C responsibility = (1) – (4) – (6) = 0 days

(8) Final result: **OO = 1 day, OC = 0 days, O∩C = 0 days**

The final analysis results are then determined by summing all windows results. Accordingly, the final analysis results are (OO = 1 day, OC = 2 days, O∩C = 0 days). Table 3.3 summarizes Scenario 2 results under the corrected But-For interpretation, for comparison.

Table 3.3 Delay analysis results of Scenario 2

Delay Analysis Technique	Responsibility (days)		
	Owner	Contractor	Concurrent
But-For – Single window - Before Concurrency Correction	1	0	2
But-For – Single window - After Concurrency Correction	1	2	0
But-For – Two windows	1	2	0

3.6 But-For Analysis Considering Multiple Baseline Updates

The as-planned schedule represents the contractor's best plan for the work execution based on its past experience and the information available in the planning stage (Menesi, 2007). However, the contractor may revise the as-planned schedule (including changes in logic) during the construction to recover unexpected project delays or to accommodate additional work, or even expedite project execution based on the owner's request. During the project execution, when project parties agree on a new as-planned schedule to replace the old one, it becomes the baseline for measuring work progress after the update while the earlier part of project work is to be measured against the original baseline. In such cases, delay analysis must consider the baseline updates if the project has two or more baselines.

Multiple-window But-For analysis can be extended to handle cases with multiple baseline updates. To account for such cases, in addition to the analysis windows selected earlier to respect event chronology and account for the adopted concurrency theory, a new analysis window is to be added every time the baseline is updated. During the analysis, if the current analysis window corresponds with a baseline update, then the responsibility of the delay/acceleration due to this update should be assigned. First, the contemporaneous schedule (updated schedule) duration before the baseline update (CSd) and the new baseline update duration (Bd) should be calculated. If CSd > Bd then the situation is acceleration; accelerations are assigned as negative delay (K. Zhang & Hegazy, 2005); If CSd < Bd then the situation is a delay. Delays and accelerations should be apportioned based on causations to the owner and/or the contractor.

To explain multiple baseline analysis procedures using But-For analysis, the case study (Scenario 2) was slightly altered in Scenario 3 by adding a second baseline at the end of Day 5. After the contractor caused 2 days non-excusable delay on Days 4 and 5, the owner did not accept the new project duration of 10 days and asked the contractor to accelerate the work and finish it within the original duration (8 days). The new baseline (Figure 3.5) was approved at the end of Day 5 by introducing a negative lag of 2 days to the finish-to-start relationship between activity C and activity D as a corrective action by the contractor. The as-built schedule remains as is for Scenario 2 including the logic change introduced in the second updated baseline so that the project was completed 3 days behind schedule.

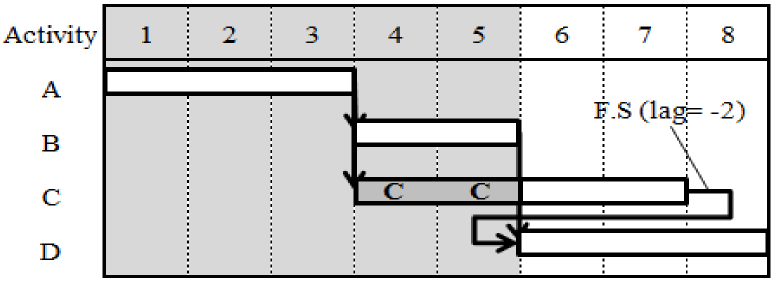


Figure 3.5 Updated baseline schedule (Scenario 3)

The correct But-For analysis considering the multiple baseline updates is applied to this scenario. The contemporaneous schedule up to Day 5 before the baseline update is the same as Scenario 2 given in Figure 3.4a with a duration of 10 days. Applying the correct But-For analysis to this window is as follows:

(1) Total project delay (OUC) of this window = 2 days

(2) As-built duration (Figure 3.4a) this window = 10 days

(3) As-built without O events of this window = 10 days

(4) OO responsibility = (2) – (3) = 0 days

(5) As-built without C events (Figure 3.4b) = 8 days

(6) OC responsibility = (2) – (5) = 2 days

(7) O∩C responsibility = (1) – (4) – (6) = 0 days

(8) Final result: **OO = 0 days, OC = 2 days, O∩C = 0 days**

At the end of this window, the contractor updated the baseline by introducing logic changes to accelerate the work and maintain the same project duration. This acceleration of 2 days is then attributed to the contractor, and thus a responsibility of OC = -2 days is accumulated.

The contemporaneous schedule up to Day 8 is similar to the schedule of Figure 3.4c with a duration of 11 days. Applying the correct But-For analysis to this window is as follows:

(1) Total project delay beyond previous window = 1 day

(2) As-built duration (Figure 3.4c) this window = 11 days

(3) As-built without O events (Figure 3.5) (baseline update) = 8 days

(4) OO responsibility = (2) – (3) = 3 day

(5) As-built without C events = 11 days

(6) OC responsibility = (2) – (5) = 0 days

(7) O∩C responsibility = (1) – (4) – (6) = 0 days

(8) Final result**: OO = 3 day, OC = 0 days, O∩C = 0 days**

The final analysis results are then determined by summing all windows results including the results accumulated in the baseline update calculations. Accordingly, the final analysis results are (OO = 3 day, OC = 0 days, O∩C = 0 days).

3.7 Second Case Study

To highlight the correct but-for analysis and the improvements in concurrency assessment, a larger practical case study obtained from the literature (Stumpf, 2000) was analyzed. The case study involves twelve activities for the construction of a house and a garage. The as-planned and as-built schedules (Figure 3.6) show the activities' durations and relationships, in addition to the delay events of the owner and the contractor. The project as-planned duration was originally 16 weeks, with the as-built duration being 24 weeks (a total project delay of 8 weeks). In this case study, the result of the traditional But-For under different perspectives are as follows:

– But-For with owner’s perspective: OO = 3 weeks, OC = 5 weeks (from Figure 3.7).

– But-For with contractor’s perspective: OO = 6 weeks, OC = 2 weeks (from Figure 3.8).

Applying the correct But-For analysis with a single window to this case study is then done in the following steps:

(1) Total project delay (OUC) = 8 weeks

(2) As-built duration = 24 weeks

(3) As-built without O events (Figure 3.7) = 21 weeks

(4) OO responsibility = (2) – (3) = 3 weeks

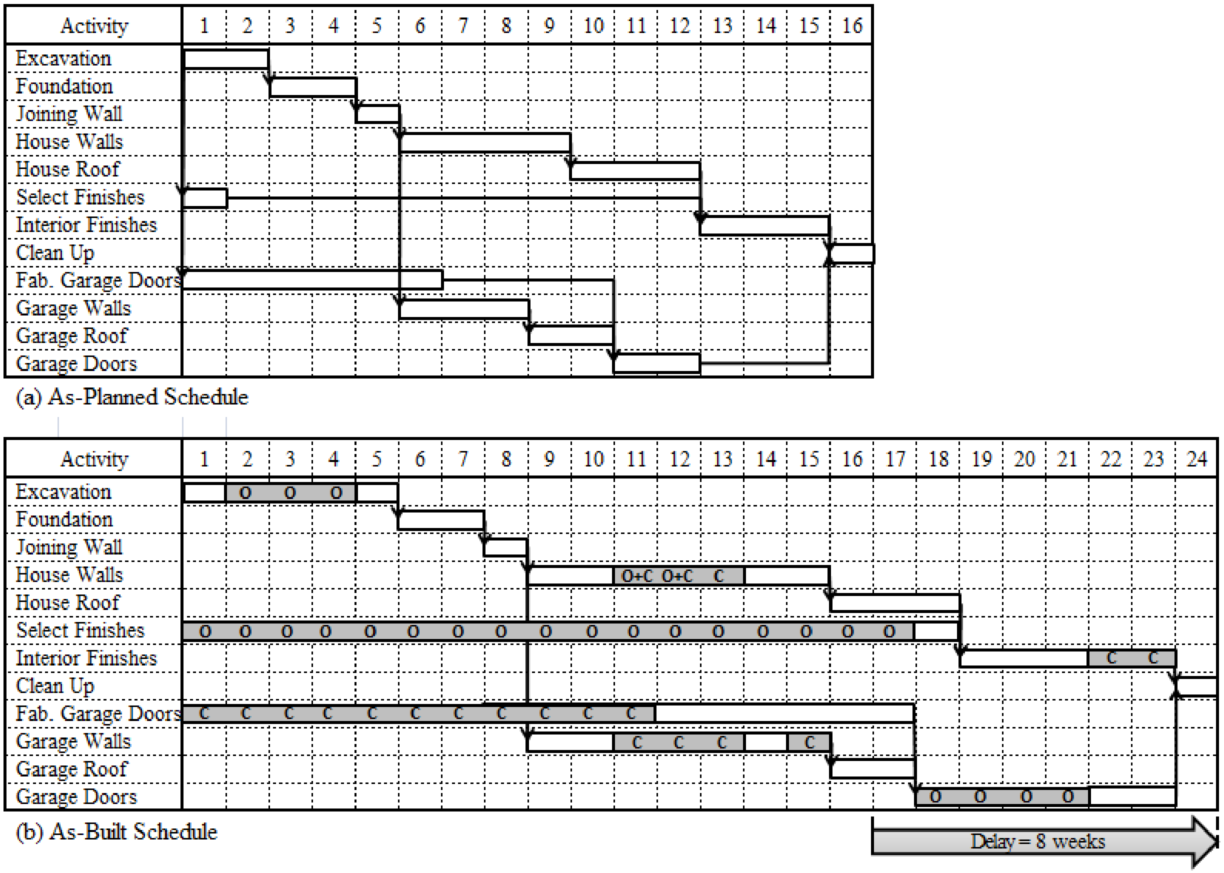
(5) As-built without C events (Figure 3.8) = 22 weeks

(6) OC responsibility = (2) – (5) = 2 weeks

(7) O∩C responsibility = (1) – (4) – (6) = 3 weeks

(8) Final result: **OO = 3 weeks, OC = 2 weeks, O∩C = 3 weeks**

Accordingly, the overall result of the proposed correct But-For method is: OO = 3 weeks, OC = 2 weeks, O∩C = 3 weeks. To improve But-For results, the multi-window But-For analysis is applied by first looking at the as-built schedule of Figure 3.6b, and identifying 7 distinct windows to satisfy the window-size rule described earlier so that different party’s events are on separate windows, unless the events are simultaneous. The analysis windows are arranged to separate critical events of different parties in different windows unless the events are happening simultaneously to ensure that event chronology is respected. For example Window 1 was set as (week 1) to have the owner and contractor events on day 1 assessed together. In Window 2 (Weeks 2-4) are selected since the owner and the contractor events on (Select Finishes) and (Fab. Garage Doors) activities are non-critical and have no impact on the project completion date; using this window the effect of owner events on (Excavation) activity will be assessed separately. Similarly, the other windows are selected to include delay critical events of only one party. The final arrangement of the 7 analysis windows is shown in table 4.4.



(a) As-Planned Schedule

(b) As-Built Schedule

Figure 3.6 Second case study schedules

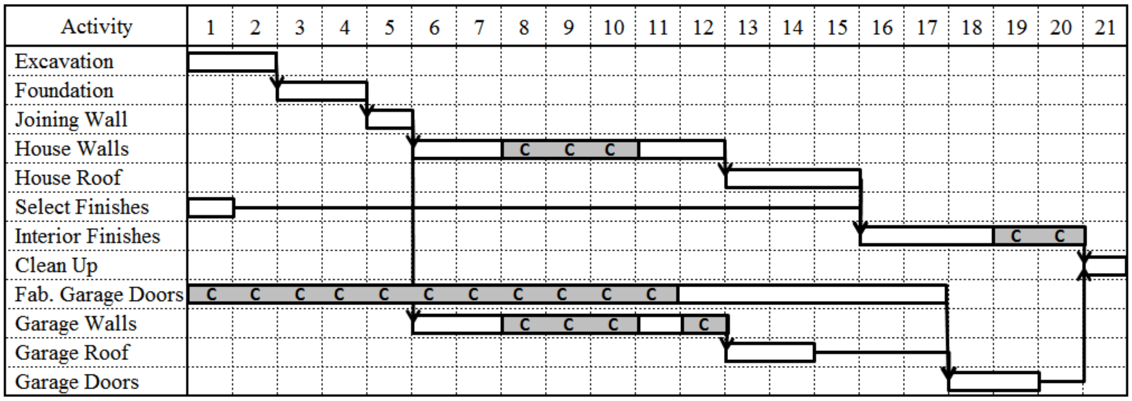


Figure 3.7 As-built schedule without owner's events (Second case study)

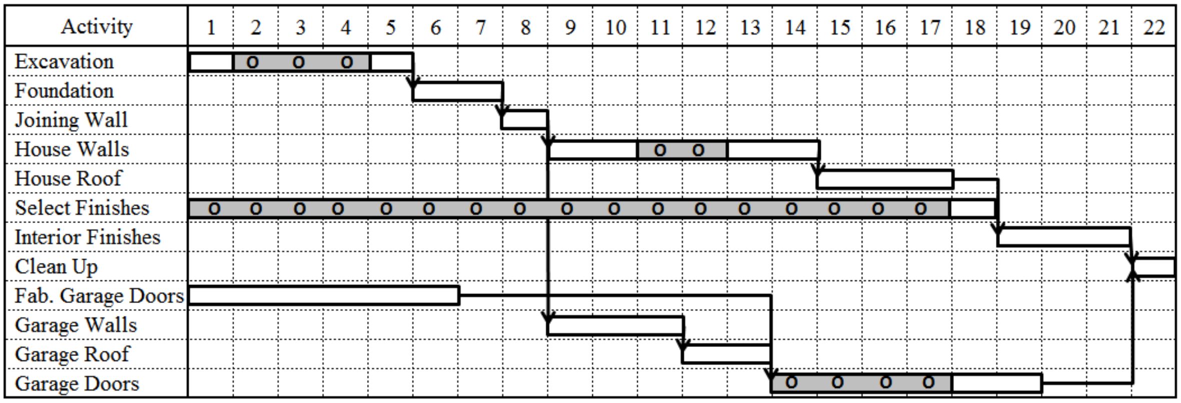


Figure 3.8 As-built schedule without contractor's events (Second case study)

Window 1: Week 1 only – one week of non-critical owner and contractor simultaneous events. Responsibility (weeks): OO = 0, OC = 0, O∩C = 0.

Window 2: Weeks 2-4 – three weeks of critical owner event simultaneous with non-critical events (owner and contractor). Responsibility (weeks): OO = 3, OC = 0, O∩C = 0.

Window 3: Weeks 5-10 – five weeks of non-critical owner and contractor simultaneous events. Responsibility (weeks): OO = 0, OC = 0, O∩C = 0.

Window 4: Weeks 11-13 – two weeks of critical owner and contractor events and one week of critical contractor event, simultaneous with non-critical owner and contractor events. Reasonability (weeks): OO = 0, OC = 1, O∩C = 2.

Window 5: Weeks 14-15 – one week of non-critical owner event and another week of non-critical simultaneous owner and contractor events. Reasonability (weeks): OO = 0, OC = 0, O∩C = 0.

Window 6: Weeks 16-21 – two weeks of non-critical owner events followed by four weeks of owner events on another activity, of which only 2 weeks events are critical (weeks 20 and 21). Responsibility (weeks): OO = 2, OC = 0, O∩C = 0.

Window 7: Weeks 22-24 –two weeks non-critical contractor events. Responsibility (weeks): OO = 0, OC = 0, O∩C = 0.

Summing the results of these windows represents the final analysis results, as shown in Table 3.4, which summarizes the results of the different methods.

Table3.4 shows that the proposed correct But-For analysis combines the two viewpoints of the owner and the contractor to produce repeatable results. Applying multiple-window But-For analysis increased the analysis resolution and arrived at more accurate results and represents the best that But-For can do. The results obtained, in this case, using the proposed multiple-window But-For analysis are identical to those obtained by (Hegazy & Zhang, 2005) using the more advanced daily windows analysis. The analysis windows were selected, as required by the proposed procedures, to separate different parties' events; this made the multiple windows But-For analysis able to capture the critical path fluctuations and arrive at the correct results.

Table 3.4 Delay analysis results of the second case study

Method		**Responsibility (Weeks)**		
		Owner	**Contractor**	**Concurrent**
Traditional:	Owner perspective	**3**	**5**	**0**
	Contractor perspective	**6**	**2**	**0**
Correct:	Single-Window	**3**	**2**	**3**
Correct:	Multiple-Window:			
	Week 1	0	0	0
	Weeks 2-4	3	0	0
	Weeks 5-10	0	0	0
	Weeks 11-13	0	1	2
	Weeks 14-15	0	0	0
	Weeks 16-21	2	0	0
	Weeks 22-24	0	0	0
		5	**1**	**2**

3.8 Case of Three-Party Delays

To extend the ideas presented in this chapter to the general practical case where three project parties are involved (Owner O, Contractor C, and third party N), the possible seven combinations of project delay responsibilities (OO, OC, ON, O∩C, C∩N, O∩N, and O∩C∩N) were correlated with the three common types of project delays (Excusable – Compensable, Non-excusable, Excusable - Not-Compensable) as shown in Table 3.5. This correlation matrix was established based on the (AACE RP 29R-03, 2011) and (ASCE 67-17, 2017) requirements. Venn representation of project delays of Figure 2.4b shows that third party delays and all combinations of concurrent delays (ON, O∩C, C∩N, O∩N, and O∩C∩N) are excusable not-compensable delays, while the Owner Only OO delay is an excusable compensable delay, and the Contractor Only OC delay is a non-excusable delay.

Table 3.5 Delay responsibility and delay type correlation matrix

Delay responsibility		Delay Type
Concurrent Delays	Only Owner OO	Excusable - Compensable
	Only Contractor OC	Non-excusable
	Only Third-party ON	Excusable - Not-Compensable
	Owner and Contractor O∩C	Excusable - Not-Compensable
	Owner and Third-party O∩N	Excusable - Not-Compensable
	Third-party and Contractor N∩C	Excusable - Not-Compensable
	Owner, Contractor and Third-party O∩C∩N	Excusable - Not-Compensable

Based on the above, the step-by-step calculations of the correct But-For analysis that incorporate three-party events are as follows:

(1) Total project delay = w

(2) As-Built duration = x

(3) As-Built duration without Owner O events = y

(4) As-Built duration without Contractor C events = z

(5) Responsibility:

Excusable compensable delays (EC) = $x – y$ = OO

Non-excusable delays (NE) = $x – z$ = OC

Excusable not-compensable delays (ENC) = w – OO – OC = (ON + O∩C + C∩N + O∩N + O∩C∩N)

If a more detailed analysis is needed for different compensation rules that require calculating the project delays for the seven project delay combinations (OO, OC, ON, O∩C, C∩N, O∩N, and

O∩C∩N) separately, the modified But-For method (MBF) proposed by (Mbabazi et al., 2005) can be used. The MBF utilizes sets theorem to calculate the seven possible combinations of project delays. An improved MBF procedures to allocate schedule delays and accelerations are presented in chapter 4.

With this ability to handle the 3-party delay, the targeted improvements shown in Table 3.1 have been achieved; this includes correcting the misconceptions involved in the original technique, improving its concurrency assessment, and interpreting these improvements into a detailed step-by-step methodology, as shown in the flowchart of Figure 3.9. But-For analysis with multiple windows respects the event chronology only between the windows but can violate the chronology within each window. Window size, therefore, has an impact on the analysis results; the smaller the window size, the more accurate are the results (i.e., higher analysis resolution). However, this is not always true; if the functional theory is adopted, selecting too small window size may separate events close in time in different windows and disqualify them for concurrency. Analysis window size should be selected carefully to enable the technique to distinguish between true concurrency and concurrent effects, but also to account for the adopted concurrency theory. As the literal theory is adopted in this research, the window size should be selected to separate both parties' events in different windows. If the functional theory is adopted in the analysis, analysis windows should be arranged to qualify critical events close in time for concurrency. Furthermore, to consider multiple baselines in the analysis, a new analysis window is to be added every time the baseline is updated. As mentioned earlier in large projects, with many events and more complex interactions, a concurrency test is not a simple task, and identifying the proper window size that could arrive at the correct results is not clear. In such situations, a more accurate delay analysis method becomes necessary. The Enhanced Daily Windows Analysis presented in chapter 5 can handle such complex situations.

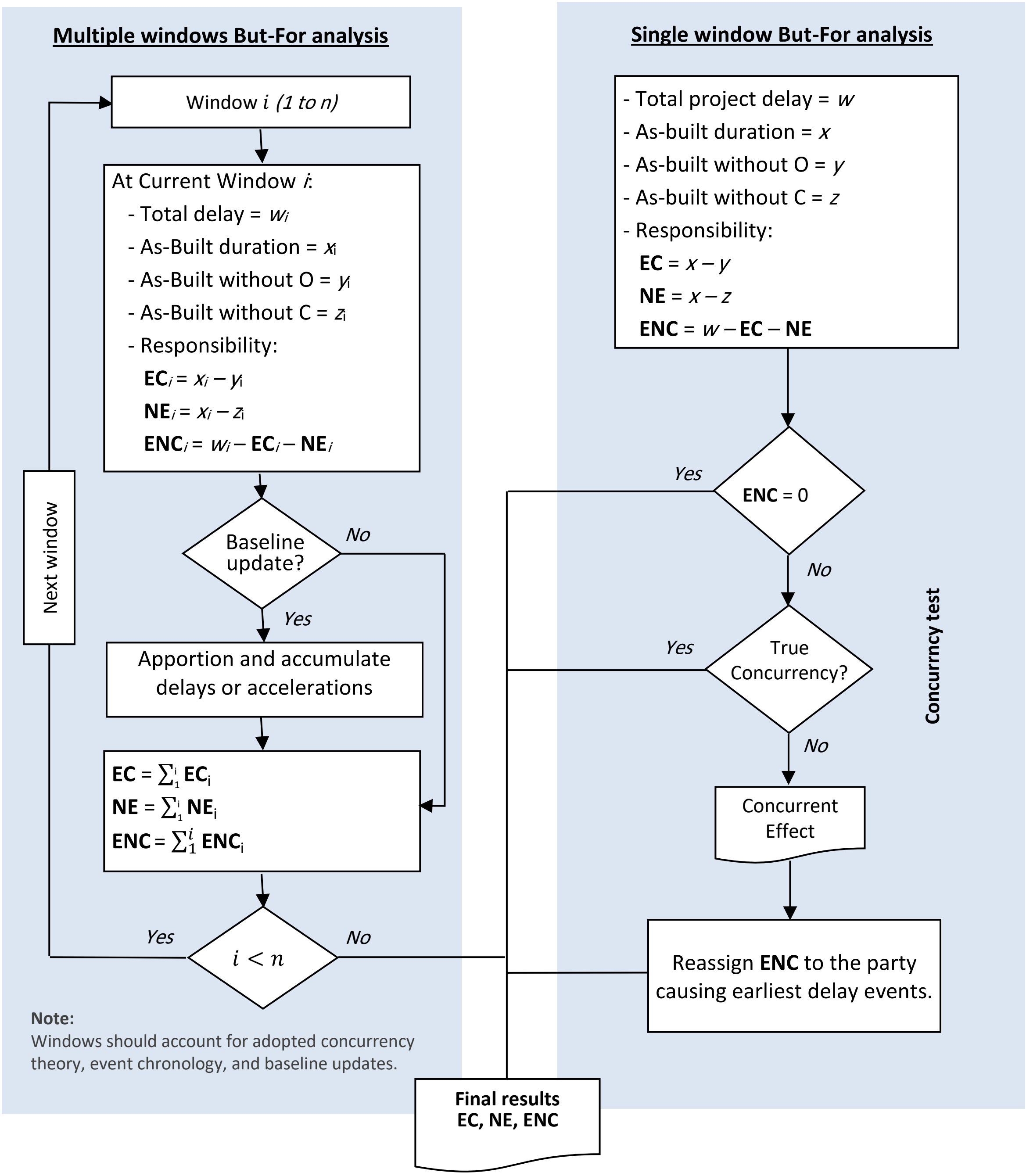

Multiple windows But-For analysis
Window i (1 to n)
At Current Window i:
- Total delay = w_i
- As-Built duration = x_i
- As-Built without O = y_i
- As-Built without C = z_i
- Responsibility:
$\mathbf{EC}_i = x_i - y_i$
$\mathbf{NE}_i = x_i - z_i$
$\mathbf{ENC}_i = w_i - \mathbf{EC}_i - \mathbf{NE}_i$
Baseline update?
No
Yes
Apportion and accumulate delays or accelerations
$\mathbf{EC} = \sum_1^i \mathbf{EC}_i$
$\mathbf{NE} = \sum_1^i \mathbf{NE}_i$
$\mathbf{ENC} = \sum_1^i \mathbf{ENC}_i$
Next window
Yes
$i < n$
No
Note:
Windows should account for adopted concurrency theory, event chronology, and baseline updates.
Single window But-For analysis
- Total project delay = w
- As-built duration = x
- As-built without O = y
- As-built without C = z
- Responsibility:
$\mathbf{EC} = x - y$
$\mathbf{NE} = x - z$
$\mathbf{ENC} = w - \mathbf{EC} - \mathbf{NE}$
Yes
$\mathbf{ENC} = 0$
No
Yes
True Concurrency?
No
Concurrncy test
Concurrent Effect
Reassign ENC to the party causing earliest delay events.
Final results EC, NE, ENC

Figure 3.9 Improved But-For analysis methodology

3.9 Summary

The delay analysis results of But-For method have traditionally been misinterpreted. Correct application of But-For analysis has been presented in this chapter using Venn representation to identify concurrent delays. However, even with this correct analysis, using But-For with a single analysis window produces results that do not differentiate between truly concurrent delays and concurrent effects, thus requiring adjustments to the results. To improve the But-For analysis, a process of investigating the as-built schedule is suggested to examine the true concurrency and apply correct But-For analysis with properly selected multiple windows. Alternatively, for larger projects that cannot be analyzed manually, concurrency analysis is done more accurately using a more rigorous analysis method than But-For. The developments made in this chapter provide explicit and structured details to correctly implement the But-For analysis in a manner that matches the general delay analysis guidelines of international organizations such as AACE.

Chapter 4

Modified But-For Analysis: Considering Accelerations and Event Chronology[2]

4.1 Introduction

This chapter builds upon the work in the previous chapter to continue enhancing the But-For analysis. As the second component of the proposed framework, this chapter introduces enhancements to the modified But-For method to address its known drawbacks of allocating only delays overlooking the common schedule accelerations, and its insensitivity to critical path(s) fluctuations and event chronology. Thus, this chapter extends the MBF methodology to allocate both schedule delays and accelerations, and introduces multi-windows MBF analysis to consider event chronology and multiple baseline updates. The application of the proposed enhancements is illustrated using several case studies.

4.2 MBF Method

But-For is one of the most widely used and accepted techniques (Dale & D'Onofrio, 2017; Lifschitz et al. 2009). However, as mentioned in chapter 2, the traditional But-For method exhibits four main drawbacks: (1) it produces different results when adapting different parties' viewpoints (Braimah, 2013; Fawzy & El-Adaway, 2012; Magdy & Georgy, 2019); (2) it is unable to identify the concurrent delays that occurred during execution (Bhih & Hegazy, 2020b; Mbabazi et al., 2005); (3) it is insensitive to the specific timing of parties' events on the schedule; and (4) it is insensitive to the fluctuation of

[2] This chapter is part of the following two published manuscripts.
- Bhih, M., & Hegazy, T. (2020). Enhanced But-For Method (EBFM) to Apportion Net Delays and Accelerations. *ASCE - Journal Construction Engineering and Management*, https://doi.org/10.1061/(ASCE)CO.1943-7862.0002094.
- Bhih, M., & Hegazy, T. (2021). Multiple-Window Modified But-For Analysis of Project Delays and Accelerations. International Journal of Construction Management, https://doi.org/10.1080/15623599.2021.1916708.

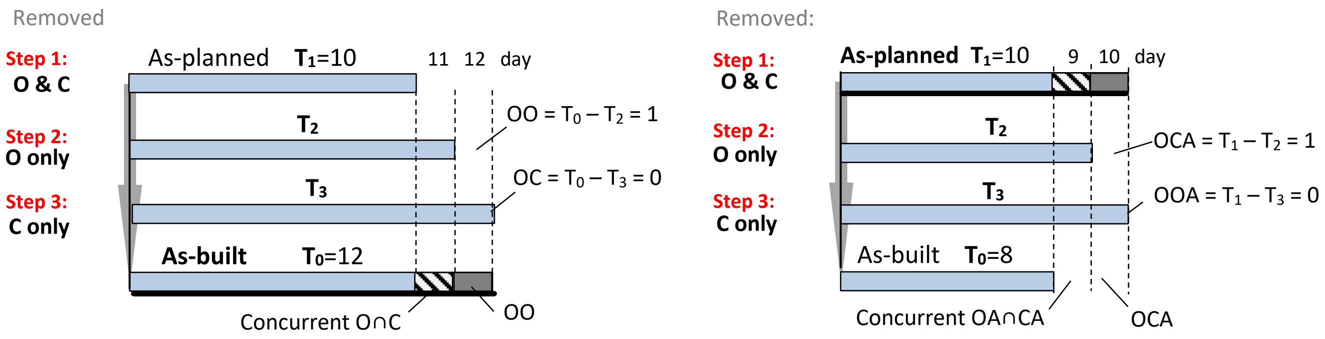
the critical path(s) during construction. In the literature, Mbabazi et al. (2005) introduced the Modified But-For (MBF) method to address the first two drawbacks of the traditional But-For analysis, thus producing consistent results irrespective of the adopted viewpoint, and accurately identifying concurrent delays. The MBF methodology was presented in section 2.4.3.1 of this thesis; it utilizes Venn representation and set theorem to allocate delay responsibilities. Venn representation and related equations are shown in Figure 2.13a, b, and Table 2.4. The seven steps in Table 2.4 can be used to calculate the project delay responsibilities. Despite the improvements introduced by the MBF, the latter drawbacks (3 and 4) still persist due to the fact that MBF uses a single time window, of the whole as-built schedule, for the analysis (Bhih & Hegazy, 2019). Furthermore, the MBF method was geared to allocate only schedule delays overlooking the common schedule accelerations.

Improvements to the MBF method are introduced to resolve its persistent limitations, improve its accuracy, and avoid calculation errors by introducing enhancements that adhere to recent delay analysis standards of the ASCE (ASCE 67-17, 2017) and the AACEI recommended practice (AACE RP 29R-03, 2011). Firstly, extending the MBF methodology to allocate both delays and accelerations is introduced. Afterwards, the proposed MBF improvements to consider event chronology and their application under multiple baseline updates are explained through the use of a multiple-window MBF analysis that separates the events of different parties in smaller time windows. This increases the analysis resolution, accounts for critical path fluctuations, and considers the detailed chronology of progress events. The analysis, as such, becomes comprehensive enough to analyze individual and/or concurrent delays and accelerations, and schedules involving multiple baseline updates. Example applications are then used to discuss and validate the introduced enhancements.

4.3 Apportioning Accelerations

Schedule acceleration means completing the work in less time than planned, leading to a consequent time-saving in project duration. For fairness, acceleration is to be attributed to the party who incurs its cost; i.e., owner-directed acceleration is to be credited to the owner, while the contractor-voluntary acceleration is to be credited to the contractor (AACE RP 29R-03, 2011). It is reasonable also to use one party's acceleration to offset that party's delays (K. Zhang & Hegazy, 2005). In contrast to delays, project acceleration can only be attributed either to the owner or to the contractor, but not to any third party. Therefore, the acceleration responsibilities are: Only-Owner Acceleration (OOA); Only-Contractor Acceleration (OCA); and Concurrent Owner and Contractor Acceleration (OA∩CA).

Despite the MBF's ability to reconcile all parties' viewpoints and produce repeatable results, the formulation in Table 2.4 was developed only for projects that exercise net delays (i.e., projects ended behind schedule), and not designed for projects that exercise net acceleration (i.e., projects ended ahead of schedule). The assessment of project delays and project accelerations using But-For analysis are quite different; the MBF method can produce wrong results when applied to the case of acceleration, as illustrated in Figure 4.1, comparing a case of net project delay (Figure 4.1a) versus a case of net project acceleration (Figure 4.1b). In the case of net project delay with an as-built duration of T_0 = 12 days, the as-planned duration (T_1 = 10 days) is produced by removing all O+C events (top row of Figure 4.2a); the project ended 2 days behind schedule. In the next step, MBF-delay analysis removes all owner events from the as-built schedule (second row of Figure 4.1a), arriving at a project duration of T_2 = 11 days. As such, the only-owner responsibility OO = T_0 - T_2 = 12 – 11 = 1. Next, removing all contractor events, the OC = 0 can be evaluated, then the concurrent delay O∩C = 1 is computed, as shown in Figure 4.2a).



a) Project with net delay of 2 days

b) Project with net acceleration of 2 days

Figure 4.1 Difference between the assessment of delays versus accelerations

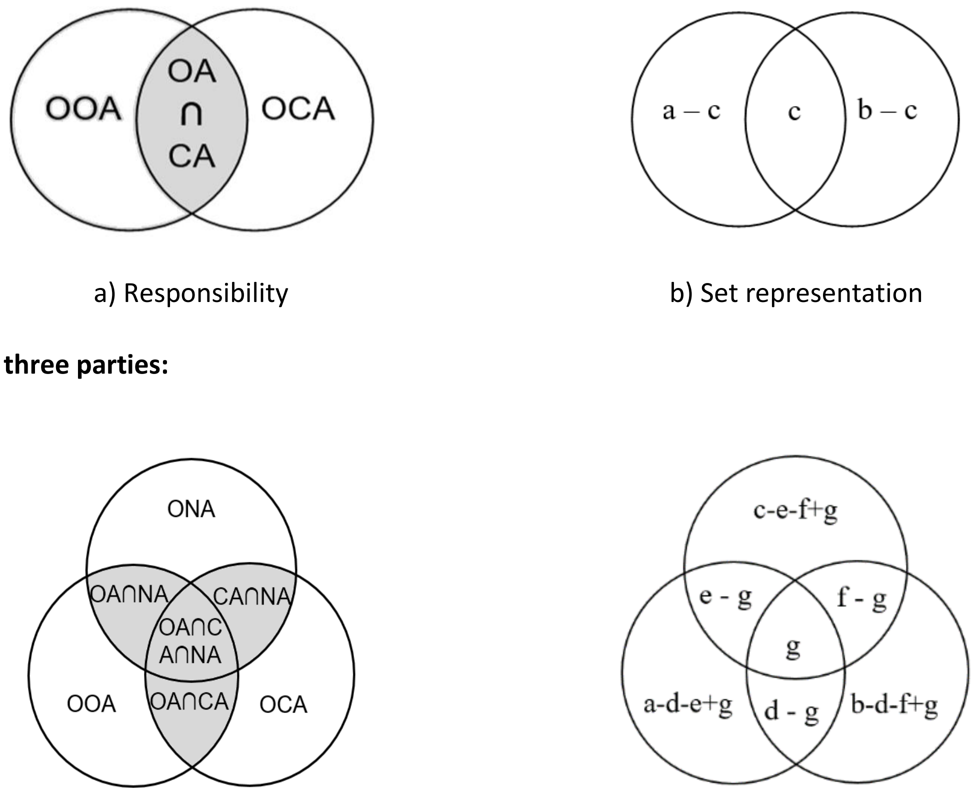
In the case of net project acceleration (Figure 4.1b), on the other hand, the as-planned duration (T_1 = 10 days) is the same as before but the as-built duration is shorter (T_0 = 8 days), which means that the project ended 2 days earlier than the deadline. When removing all owner events (second row of Figure 4.1b), a project duration of T_2 = 9 days is obtained, thus the project duration increased, and it is obvious that the 1 day increase is due to owner missing events, including its contribution to concurrent acceleration, while the other project acceleration day is attributed solely to the contractor. Thus, OCA = $T_1 - T_2$ = 1 day. Similarly, the Only-Owner Acceleration can be determined by removing all contractor events, thus, OOA = 0. Finally, the concurrent acceleration OA∩CA = 1 is computed, as shown in Figure 4.1b.

The departure in the calculation between the cases of delay versus acceleration becomes clear by further analysis of Figure 4.1. In essence, the delays deducted from the as-built schedule due to removing the events of one party represent the sole responsibility of that party (i.e., removing O events calculates OO delay responsibility, $T_0 - T_2$). On the other hand, the acceleration to the as-planned duration as a result of removing the events of one party from the as-built schedule represents the sole acceleration of the other party (i.e., removing O events calculates OCA acceleration responsibility, $T_1 – T_2$). Also, delays are measured based on the as-built schedule, while accelerations are measured based on the as-planned schedule.

Based on this discussion, the proposed Enhanced Modified But-For Method uses the MBF delay computation in Table 2.4 only for the projects with net delay (projects ended behind schedule). The extended analysis in the case of net acceleration, on the other hand, has been formulated using a new set of equations, as shown in Figure 4.2a, b, and Table 4.1. The shaded circle in Figure 4.2a shows the region of concurrent acceleration. As explained earlier in the case of delay in chapter 2, three variables (a, b, and c) are used to represent project acceleration responsibilities (Figure 4.2b). In the first step, all O+C events are removed from the as-built schedule and the total project acceleration is calculated as the difference between the as-planned and as-built schedules ($OOA+OCA+OA\cap CA = T_1\text{-}T_0 = a + b - c$). In the second step, all owner events are removed from the as-built schedule to obtain duration T_2, then, the only-contractor acceleration is determined using the as-planned duration as explained in the previous section as ($OCA = T_1 – T_2 = b – c$). Similarly, the only-owner acceleration is determined as ($OOA = T_1 − T_3 = a − c$). The three equations are then solved together to obtain the set-variables values and accordingly the project acceleration responsibility as shown in Table 4.1.

As a generalization of the two-party acceleration analysis, a three-party analysis has been formulated as shown in Figure 4.2d, e, and Table 4.2. As such, the proposed enhanced MBF uses the formulations in Tables 2.4 and 4.2 to address the projects with net delays and net accelerations, respectively. Even though the acceleration can be attributed only to owner or contractor, the formulation of three-party acceleration was prepared to handle cases of net accelerations that involve third-party delay; it is noted that in the enhanced MBF acceleration analysis, delays are considered as negative acceleration.

Case of two parties:



a) Responsibility

b) Set representation

Case of three parties:

c) Responsibility

d) Set representation

* OOA= Only Owner Acceleration; OCA = Only Contractor Acceleration; ONA = Only third party Acceleration

Figure 4.2 Calculation of acceleration responsibility

Table 4.1 MBF acceleration calculations – case of two parties

Removed Events	Resulted Project Duration	Venn Calculations	Project Acceleration Responsibilities
O + C	T_1	$T_1 – T_0^* = a + b - c$	$OOA = T_1 – T_2$
O	T_2	$T_1 – T_2 = b - c$	$OCA = T_1 – T_3$
C	T_3	$T_1 – T_3 = a - c$	$OA \cap CA = T_2 + T_3 – T_0 – T_1$

T_0^* = as-built duration

Table 4.2 MBF acceleration calculations – case of three parties

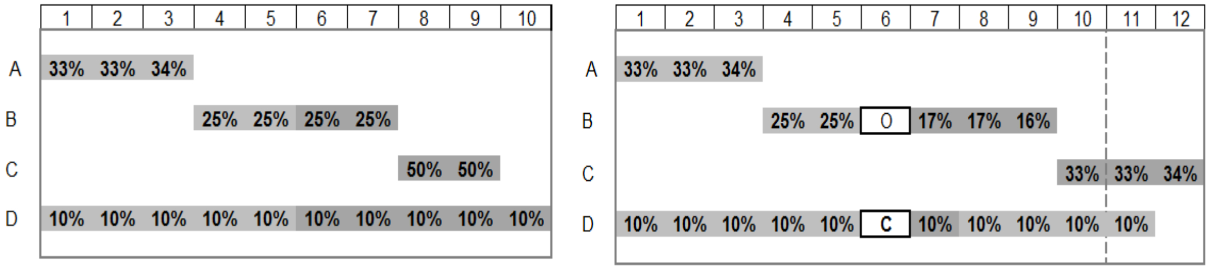
Removed Events	Resulted Project Duration	Venn Calculations	Project Acceleration Responsibilities
O + C + N	T_1	$T_1–T_0^* = a+b+c-d-e-f+g$	$OOA = T_1 – T_7$
O	T_2	$T_1 – T_2 = b+c-d-e-f+g$	$OCA = T_1 – T_6$
C	T_3	$T_1 – T_3 = a+c-d-e-f+g$	$ONA = T_1 – T_5$
N	T_4	$T_1 – T_4 = a+b-d-e-f+g$	$OA \cap CA = T_6 + T_7 – T_1 – T_4$
O+C	T_5	$T_1 – T_5 = c-e-f+g$	$OA \cap NA = T_5 + T_7 – T_1 – T_3$
O+N	T_6	$T_1 – T_6 = b-d-e+g$	$CA \cap NA = T_5 + T_6 – T_1 – T_2$
C+N	T_7	$T_1 – T_7 = a-d-f+g$	$OA \cap CA \cap NA = T_1 + T_2 + T_3 + T_4 – T_5 – T_6 – T_7 – T_0$

T_0^* = as-built duration

4.3.1 Example Application

To illustrate the MBF analysis for the case of delay and acceleration, a small case study of four activities with two different progress scenarios, exhibiting schedule delay and acceleration respectively, (adopted from Zhang & Hegazy, 2005) is analyzed. The project has the as-planned schedule of Figure 4.3a of 10 days duration; the as-built schedule (Scenario 1) of Figure 4.3b exhibits simultaneous owner and contractor events on Day 6 that led the project to be completed 2 days behind schedule. The owner event to activity B shifted the remaining portions of activities B and C to a period of lower productivity, thus taking longer time than planned and delaying the project. For simplicity, the MBF-delay methodology of two project parties is applied to this scenario according to the steps in Table 2.4; having the as-built duration of $T_0 = 12$, the first MBF step is to remove all the Owner (O) events from the as-built schedule. Accordingly, the schedule becomes $T_2 = 11$ days. Therefore, the Only-Owner (OO) responsibility is directly evaluated as the difference between T_0 and T_2, i.e., $OO = T_0 - T_2 = 12 - 11 = 1$ day. In the second step, removing all the Contractor events from the as-built, the schedule duration becomes $T_3 = 12$ days, thus, the Only-Contractor (OC) responsibility = $T_0 - T_3 = 12 - 12 = 0$ days. Also, in the third step, removing both the owner and contractor events results in the as-planned duration $T_1 = 10$ days. Accordingly, concurrent delays $O \cap C$ = Project delays - OO -

OC = (T0 – T_1) - OO - OC = 1 day. Summary results are shown in Table 4.3 (OO = 1 day, OC = 0, O∩C = 1 day). This analysis, as such, is repeatable and does not depend on any party's viewpoint. Furthermore, it does not ignore concurrent delays.



a) As-planned schedule

(b) As-built Schedule (Scenario 1)

Figure 4.3 Case study schedules

Table 4.3 Case study MBF two-party delay calculations (Scenario 1)

To get	Remove events of	Project duration (days)	Project delay responsibilities (days)
OO	O	$T_2 = 11$	OO = $T_0 – T_2$ = 12 – 11 = 1
OC	C	$T_3 = 12$	OC = $T_0 – T_3$ = 12 – 12 = 0
O∩C	O+C	$T_1 = 10$	O∩C = Project delays - OO - OC = $T_0 – T_1$ - OO - OC = 1

$T_0^* = 12$

To explain the MBF-accelerations application compared to MBF-delay calculations, progress Scenario 2 of Figure 4.4 is analyzed. The project has the as-planned schedule of Figure 4.3a of 10 days duration; the as-built schedule (Scenario 2) of Figure 4.4 exhibits net acceleration of two days. Day 6 shows two simultaneous acceleration events, one attributed to the owner and one attributed to the contractor. Table 4.4 compares the acceleration-based analysis using the MBF-acceleration steps in Table 4.1 versus the delay-based analysis of the MBF method steps in Table 2.4. The MBF-acceleration results show the responsibility being one day contractor acceleration and one day concurrent acceleration (i.e., OCA = 1 day, and OA∩CA = 1 day). This result can be logically confirmed by observing the difference between the as-built (Figure 4.4) and as-planned (Figure 4.3a) schedules. In this scenario, the delay-based MBF method produced incorrect results (OO = -1 day OC = -2 days, O∩C = 1 day), even considering that the negative signs to mean acceleration.

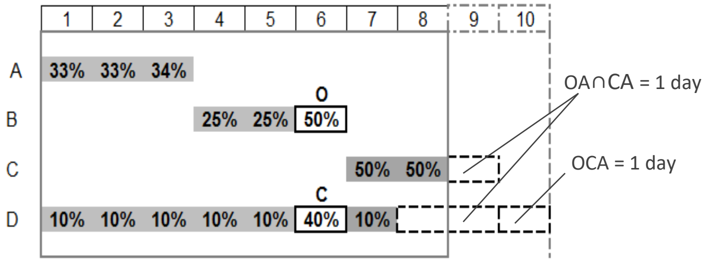


Figure 4.4 Scenario 2 as-built schedule

Table 4.4 Analysis of Scenario 2

Remove Events of	Project Duration (days)	Responsibility using MBF-acceleration	Responsibility using MBF-delay
O	$T_2 = 9$	$OOA = T_1 – T_3 = 10-10 = 0$	$OO = T_0 – T_2 = 8-9 = -1$
C	$T_3 = 10$	$OCA = T_1 – T_2 = 10 – 9 = 1$	$OC = T_0 – T_3 = 8 – 10 = -2$
O + C	$T_1 = 10$	$OA∩CA = T_2+T_3–T_0–T_1 = 9+10 –8–10 = 1$	$O∩C = T_2+T_3–T_0–T_1= 9+10–8–10 = 1$

$T_0^* = 8$

4.4 Modified MBF Analysis Considering Event Chronology

As the MBF method uses the entire project as a single analysis window, by nature, it cannot respect events' chronology, i.e., the specific timing of the events. Furthermore, the MBF only considers the final critical path(s) of the as-built schedule, overlooking path fluctuations over the course of the project, which affect the analysis results. To highlight this issue, the as-built schedule of Scenario 1 is slightly altered as shown in Figure 4.5. In this scenario, the contractor and owner delay events were shifted so that each one is happening on a different day, thus are no longer being simultaneous.

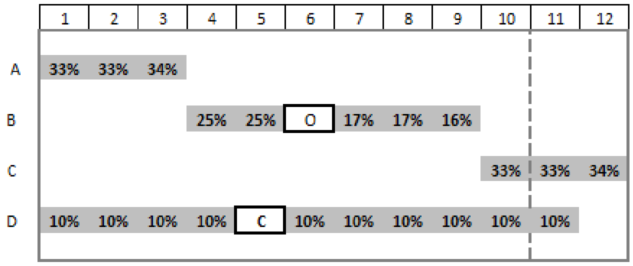


Figure 4.5 As-built Schedule (Scenario 3)

Using a single-window MBF delay analysis, Scenario 3 was analyzed following the step-by-step process explained earlier. The results are shown in Table 4.5 being (OO = 1 day, O∩C = 1 day). These results are identical to those obtained for the original Scenario 1 despite the difference in the timing of events. This shows how the MBF analysis is not sensitive to the timing of delay events.

Table 4.5 MBF calculations (Scenario 3)

To get	Remove events	Project duration (days)	Project delay responsibilities (days)
OO	O	T_2 = 11	OO = $T_0 - T_2$ = 12 – 11 = 1
OC	C	T_3 = 12	OC = $T_0 - T_3$ = 12 – 12 = 0
O∩C	O+C	T_1 = 10	O∩C = project delays – OO - OC = 2 - 1 - 0 = 1

T_0^* = 12

Irrespective of the identical results of scenarios 1 and 3, they represent two different cases of concurrency, as defined by the “SCL Delay and Disruption protocol” (SCL-DDP, 2017). Scenario 1 is a case of “true concurrency” since the events are on the same day, while Scenario 3 is a case of "concurrent effect” since the events do not occurred on the same day; the effects (project delays) of both cases are identical. Thus, the identical results indicate that single-window MBF analysis suffers from its inability to differentiate between true concurrent delays and concurrent effects due to its insensitivity to event chronology.

Taking Scenario 3 as an example, the chronological order of events is considered starting from the as-planned schedule of Figure 4.3a. The schedule had one critical path (task D) and a near-critical path (A-B-C) with 1-day float. As shown in Figure 4.6, the contractor event on Day 5 caused 1-day project delay, resulting in expected project completion on Day 11, and this also increased the float of the near-critical path (A-B-C) to 2 days. Afterwards, on Day 6, the owner event on path (A-B-C) not only consumed all the float but also caused an additional project delay, ending on Day 12. Thus, the correct responsibility of Scenario 3 considering the correct timing of events is (OO = 1 day, OC = 1 day, O∩C = 0 days). Such logical analysis cannot be done manually for large and complex projects, and modifications to the MBF are needed.

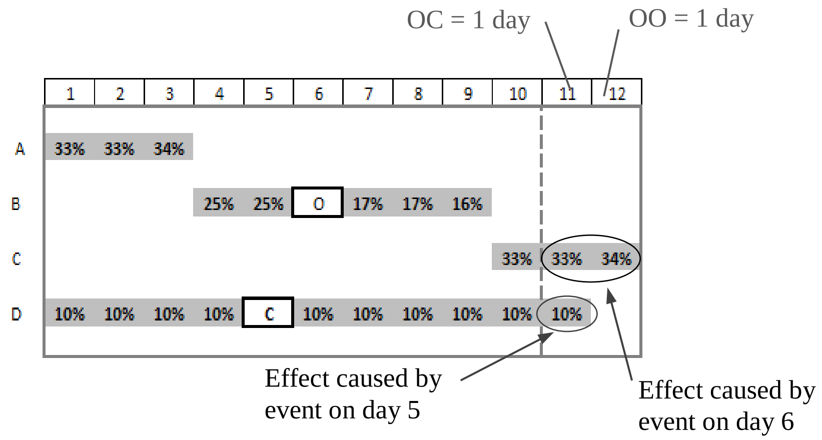
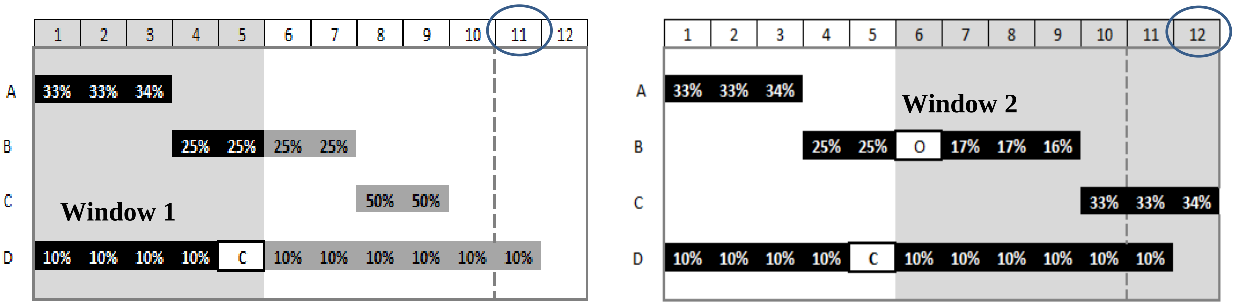


Figure 4.6 Analysis based on event chronology (Scenario 3)

4.4.1 MBF with Multiple-Window Analysis

The proposed multiple-window MBF analysis improves the tracking of critical path(s) fluctuations. The analysis windows should be selected carefully as the window size could affect the results. Having different parties' events in separate analysis windows will ensure that their effects will be assessed separately unless the events happened simultaneously or very close in time. For instance, in Scenario 3, if the literal concurrency (AACE RP 29R-03, 2011) is adopted (i.e., concurrent events must happen on the same day), then using two analysis windows (Window 1: Days 1 to 5; and Window 2: Days 6 to

12), as shown in Figure 4.7, will ensure that the analysis follows the standards and arrives at the correct answer.



(a) Window 1

(b) Window 2

Figure 4.7 Multiple window analysis of Scenario 3

Applying the two-window MBF analysis to Scenario 3 is shown in Table 4.6. Each window is analyzed separately and the result is shown in the corresponding row; finally, the analysis results are accumulated to obtain the project delay responsibility. The results obtained using this multiple-window analysis are identical to those obtained by Hegazy & Zhang (2005) using the more accurate daily-windows analysis. As such, applying multi-window MBF analysis can produce accurate results by properly separating the different parties' events in different windows. This increases the analysis resolution and makes the MBF analysis capable of better tracking of the critical path(s) fluctuations and the chronology of delay events.

Table 4.6 MBF multi-window calculations (Scenario 3)

Window	Days	As-built without party(ies) events (days)				Project delay responsibilities (days)		
		As-built T_0	O and C T_1	O T_2	C T_3	OO	OC	O∩C
1	1-5	11	11	10	10	0	1	0
2	6-12	12	11	12	11	1	0	0
						1	1	0

4.4.2 Varying concurrency viewpoints

Existing delay analysis guidelines have different definitions of concurrent delays; the disparity in definitions reflects the inconsistent application of concurrent delays in practice (Livengood, 2017). Adopting a specific concurrency viewpoint may affect the analysis results. The results of Scenario 3 in Table 4.6 was obtained under the literal definition. The AACEI recommended practice (AACE RP 29R-03, 2011) highlighted two theories of concurrency; literal vs functional concurrency. As such, the results of Scenario 3 may differ under the two theories. Under the literal theory, as shown in Table 4.7, the correct results are (OO = 1 day, OC = 1 day, O∩C = 0 days) and no concurrency exists as the causing events are happening on different days (Day 5 and 6) as shown on Figure 4.6. Under the functional theory, as the two events are close in time, thus, their effects can be considered concurrent delays; the responsibility, in this case, becomes (OO = 1 day, OC = 0 days, O∩C = 1 day). Therefore, to conduct the analysis under the literal theory, event chronology should be fully respected by analyzing them in separate windows. Under the functional theory, on the other hand, events close enough in time should be analyzed together in the same analysis window. Thus, the adopted concurrency theory may affect the arrangement of the analysis windows.

4.5 MBF Analysis under Multiple Baseline Updates

The as-planned schedule is the project baseline for measuring the actual work progress. It represents the contractor’s best estimate of the project plan based on its past experience and the information available during the planning stage (Bhih & Hegazy, 2020b). During the project execution, the contractor may need to revise its plan to recover any unexpected delay or to finish earlier than planned; revised plans are needed also to accommodate any changes in the work scope. After approving a new baseline update, the original baseline will be used only for measuring the work before the update, while the new baseline is to be used to measure the work progress after the update. The multi-window MBF analysis can accommodate multiple baseline updates by adding a new analysis window at the time of the baseline updates. The new baseline duration (B) is compared with the updated schedule duration immediately before the update (S) to know if the case is delay or acceleration. In both cases, B > S or B < S, the delay or acceleration days are to be allocated to the owner and/or contractor based on the cause of this update. At the end of the analysis, the responsibilities are accumulated through all analysis windows including the baseline updates.

4.5.1 Baseline update in case of project delay

To explain the application of the analysis procedure in case of baseline update while the project is net delay, Scenario 3 was modified in Scenario 4 by adding a baseline update (Figure 4.8). At the end of Day 8, the owner approved a new baseline update proposed by the contractor to crash the work on activity C and recover the 1 day delay caused earlier by the contractor. The work progressed according to the updated plan until the end of the project.

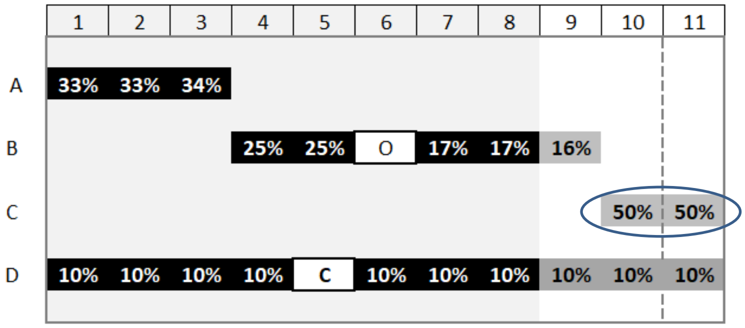


Figure 4.8 Baseline update at end of Day 8 (Scenario 4)

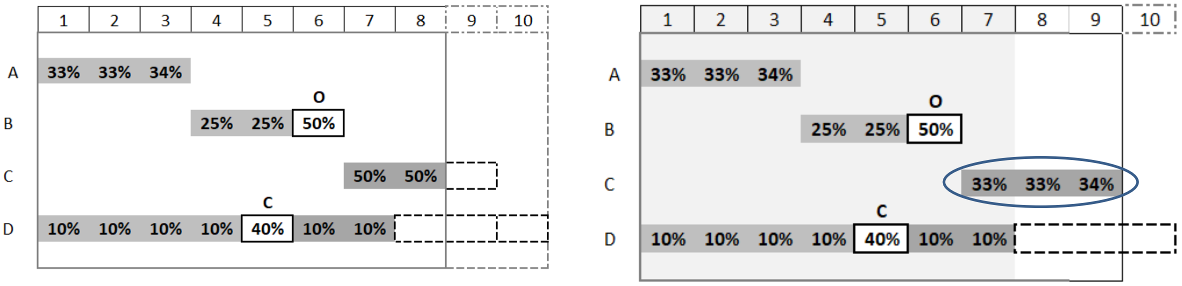
Appling the MBF Multi-Window analysis considering the baseline update starts with arranging the analysis windows as explained earlier. To respect event chronology, two analysis windows are needed to separate the two delay events (Window 1: Day 1 to 5, Window 2: Day 6 to 8); and an additional window is also needed at the end of Day 8 to allocate baseline update responsibility. The MBF analysis results are shown in Table 4.7. The first two windows use Figure 4.3a (schedule before baseline update). The first window arrived at a responsibility of 1 day attributed to the contractor (OC= 1 day), while the second window arrived at a responsibility of 1 day due to owner (OO= 1day), thus the first two rows in Table 4.7 are identical to those in Table 4.6. On Day 8, before the baseline update, the schedule duration (Figure 4.5) is 12 days. Applying the baseline update on Day 8 changes the schedule to that of Figure 4.8, with project duration updated to become 11 days, thus experiencing an acceleration of 1 day by the contractor (OCA = 1 day). The project responsibilities are obtained by accumulating the results from all windows (OO = 1 day, OC = 1 day, OCA = 1 day). Schedule accelerations can be used to offset the same party delays, since accelerations are treated as negative delay, thus, the 1 day contractor acceleration will offset the 1 day contractor delay and the final result is (OO = 1 day).

Table 4.7 MBF multiple baseline updates calculations (Scenario 4)

Window	Days	As-built without party(ies) events (days)				Project delay responsibilities (days)		
		As-built T_0	O and C T_1	O T_2	C T_3	OO	OC	O∩C
1	1-5	11	11	10	10	0	1	0
2	6-8	12	11	12	11	1	0	0
Baseline update	End of Day 8	-	-	-	-	0	-1	0
						1	0	0

4.5.2 Baseline update in case of project acceleration

To analyze the case of baseline update when the current progress is net acceleration instead of delay, another progress scenario is used (Scenario 5), which has the same as-planned schedule of Case study 1 (Figure 4.3a) with as-planned duration of 10 days. The as-built schedule (Figure 4.9a) shows acceleration events on Days 5 and 6 and as-built duration of 8 days. At the end of Day 7, the owner approved new baseline of 9 days proposed by the contractor (Figure 4.9b). To respect event chronology, two analysis windows to separate the different party events are selected (Window 1: Days 1 – 5; Window 2: Days 6 – 7); the MBF-acceleration methodology is then applied to both widows. One additional window is also selected at the end of Day 7 to account for the baseline update. The analysis results are shown in Table 4.8; first and second windows show acceleration responsibility of OCA = 1 day and OOA = 1, respectively. At the end of Day 7, before the baseline update, the schedule duration is 8 days. After the baseline update, the duration becomes 9 days, thus, this case is a 1-day delay due to the contractor (OOA = -1 day), which is used to offset the 1 day contractor acceleration; thus the final result is (OOA = 1 day).



(a) as-built schedule (b) baseline update at end of Day 7

Figure 4.9 Case study 1 (Scenario 5) schedules

Table 4.8 MBF multiple baseline updates acceleration calculations (Scenario 5)

Window	Days	As-built without party(ies) events (days)				Project delay responsibilities (days)		
		As-built T_0	O and C T_1	O T_2	C T_3	OOA	OCA	OA∩CA
1	1-5	9	10	9	10	0	1	0
2	6-7	8	9	9	8	1	0	0
Baseline update	End of Day 7	-	-	-	-	0	-1	0
						1	0	0

As demonstrated using the 5 progress scenarios of Case study 1 discussed above, the proposed multiple-window MBF has increased accuracy in considering fluctuations in the schedule including baseline updates and changes from delays to accelerations or vice versa. The general analysis procedure is shown in Figure 4.10.

4.6 Extended Case Study

To demonstrate the practical application of the multiple-window MBF analysis, an extended case study is adapted from literature (Bhih & Hegazy, 2021b). The case study schedules are shown in Figure 4.11 which shows owner and contractor acceleration and delay events. The project ended in 23 weeks with 7 weeks behind schedule, and the as-planned duration was 16 weeks. The analysis windows are then arranged according to the rule set earlier so that critical events of one type (delay or acceleration) of different parties are in separate windows unless the events are happening on the same day to ensure that event chronology is respected. For example Window 1 was set as (Day 1 to Day 5) as the owner and the contractor events on "Select Finishes" and "Fab. Garage Doors" activities are non-critical (have no impact on the project completion date); using this window the effect of owner events on "Excavation" activity will be assessed separately. Similarly, the other windows are set to include delay or acceleration critical events of only one party. The final arrangement of 8 analysis windows is shown in Figure 4.12.

The summary results of every analysis window are presented in Table 4.9, showing the following responsibilities: OO = 7 weeks, OC = 1 week, O∩C = 2 weeks, OOA = 1 week, OCA = 1 week, OA∩CA = 1 week. It is possible to use the acceleration days to offset the parties' delays and obtain the result of (OO = 6 weeks, OC = 0 weeks, O∩C = 1 week). This result matches the results obtained using the more accurate enhanced daily windows analysis (Bhih & Hegazy, 2021b).

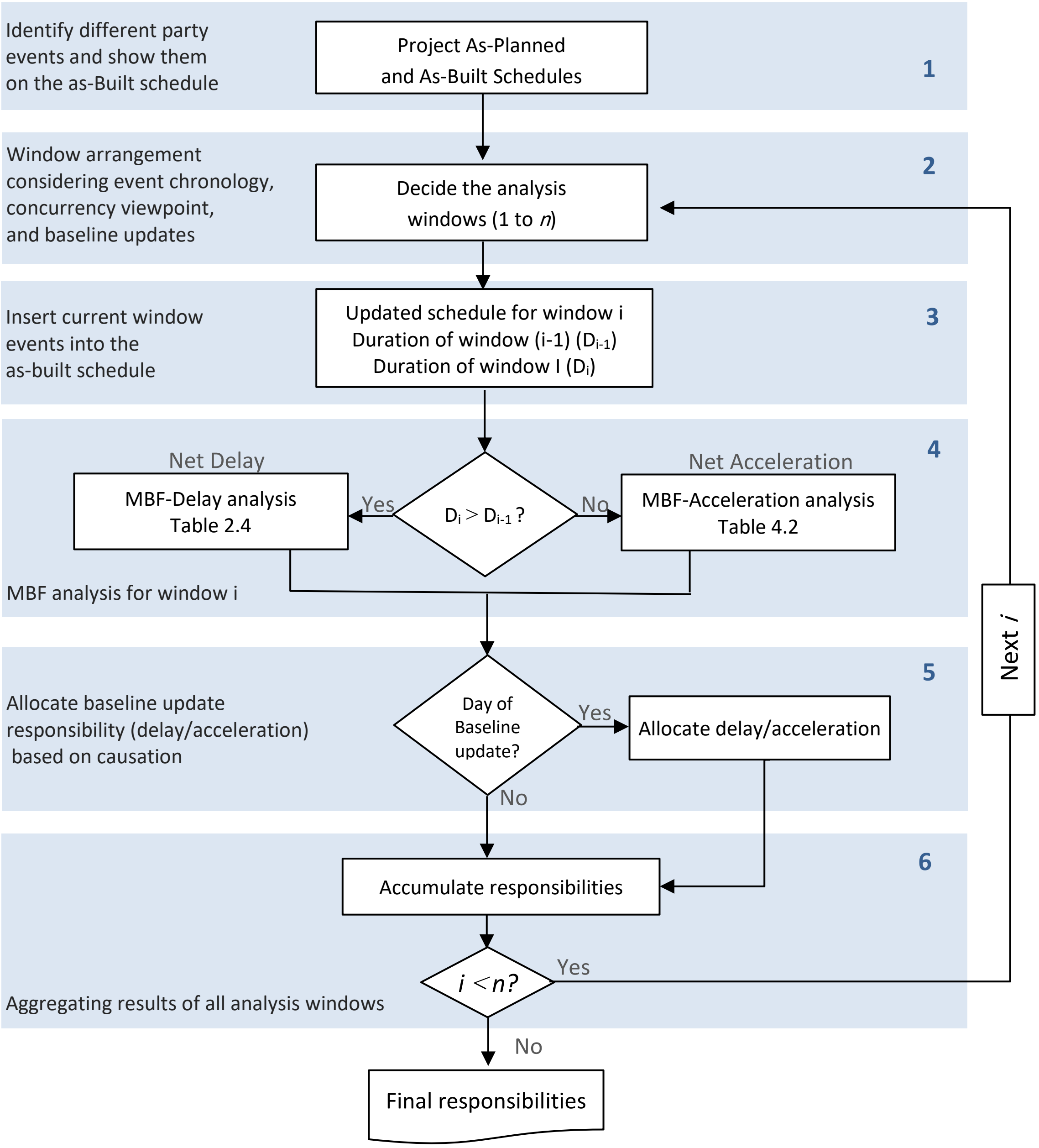
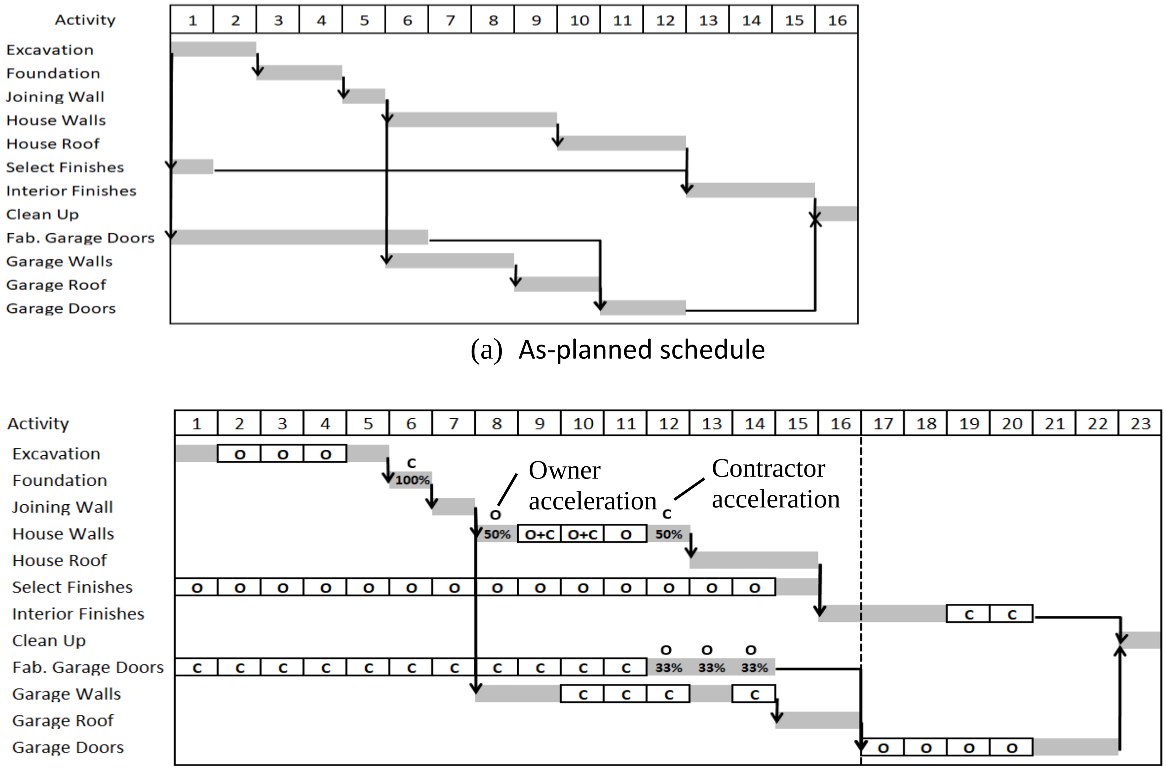

Identify different party events and show them on the as-Built schedule
Project As-Planned and As-Built Schedules
1
Window arrangement considering event chronology, concurrency viewpoint, and baseline updates
Decide the analysis windows (1 to n)
2
Insert current window events into the as-built schedule
Updated schedule for window i
Duration of window (i-1) (Di-1)
Duration of window I (Di)
3
Net Delay
MBF-Delay analysis Table 2.4
Yes
Di > Di-1 ?
No
Net Acceleration
MBF-Acceleration analysis Table 4.2
4
MBF analysis for window i
Allocate baseline update responsibility (delay/acceleration) based on causation
Day of Baseline update?
Yes
Allocate delay/acceleration
No
5
Accumulate responsibilities
6
i < n?
Yes
Aggregating results of all analysis windows
No
Final responsibilities
Next i

Figure 4.10 Multiple-window MBF analysis

The proposed improvements enabled the technique to track critical path fluctuations and respect the chronological order of events. It is important to highlight that the multiple-window MBF analysis is highly sensitive to window size. In large complex projects, selecting the analysis windows to satisfy the event chronology conditions is not an easy task, it is the key challenge for this method and could affect the results and lead to disputes among parties. In such a case, a more detailed analysis such as the daily windows analysis becomes necessary.



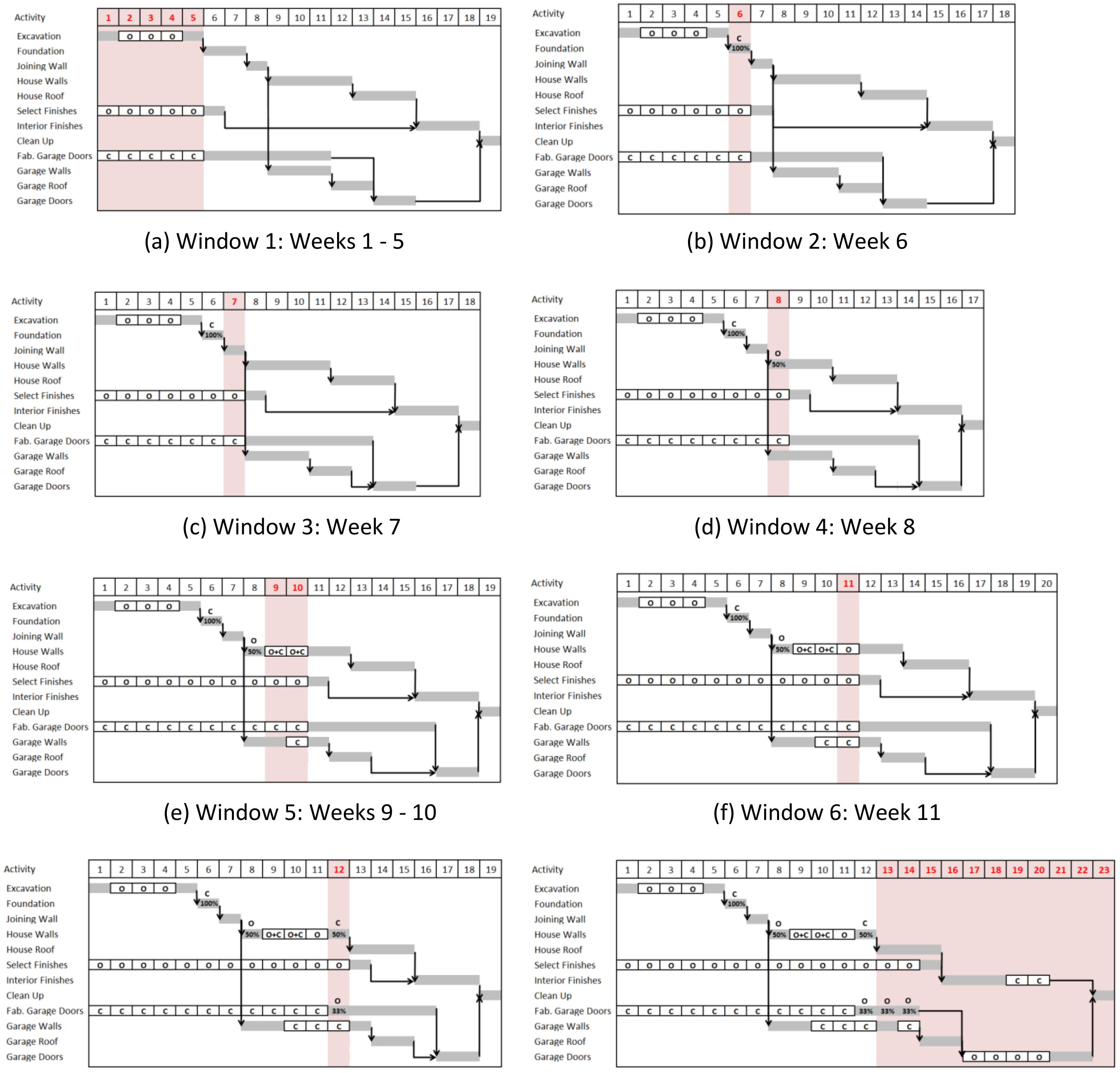
(a) As-planned schedule

(b) As-built schedule

Figure 4.11 Extended case study schedules

Table 4.9 Extended case study MBF multiple-window results summary

Analysis Window		**As-built without party events (weeks)**							**Project delay responsibilities (weeks)**							**Project acceleration responsibilities (weeks)**		
Window	As-built	O, C, N	O	C	N	O, C	O, N	C, N	OO	OC	ON	O∩C	O∩N	C∩N	O∩C∩N	OOA	OCA	OA∩CA
1-5	19	16	16	19	19	16	16	19	3	0	0	0	0	0	0			
6	18	19	18	19	-	-	-	-									1	
7	18	18	18	18	18	18	18	18	0	0	0	0	0	0	0			
8	17	18	18	17	-	-	-	-								1		
9-10	19	17	19	19	19	17	19	19	0	0	0	2	0	0	0			
11	20	19	20	19	20	19	20	19	0	1	0	0	0	0	0			
12	19	20	20	20	-	-	-	-										1
13-23	23	19	19	23	23	19	19	23	4	0	0	0	0	0	0			
									7	**1**	**0**	**2**	**0**	**0**	**0**	**1**	**1**	**1**



(a) Window 1: Weeks 1 - 5

(b) Window 2: Week 6

(c) Window 3: Week 7

(d) Window 4: Week 8

(e) Window 5: Weeks 9 - 10

(f) Window 6: Week 11

(g) Window 7: Week 12

(h) Window 8: Weeks 13 - 23

Figure 4.12 Extended case study - multiple-window MBF analysis

4.7 Summary

The proposed MBF methodology represents a more accurate, fair, and repeatable analysis, whenever But-For analysis is selected for a particular case. It has a clear procedure that meets the requirements of delay analysis standards in terms of considering detailed event chronology. For researchers, the proposed Enhanced MBF methodology represents a step towards developing a more accurate and fair forensic schedule analysis tool. It was developed in a computer-oriented fashion with step-by-step calculations to facilitate its computer implementation and minimize analysis time and effort. Furthermore, with its superiority in allocating concurrent delays in the general case of three parties, the MBF methodology could be combined with other analysis techniques to further enhance its analysis resolution and eliminate its limitations.

The traditional but-For technique has been criticized for its highly subjective nature and its inability to assess concurrent delays. Existing enhancements to but-for, namely MBF, uses Venn representation to overcome its drawbacks by reconciling the opposing viewpoints of project parties, apportioning concurrent delays, and analyzing both delays and accelerations. However, due to its single-window implementation, the MBF method suffered from the inability to respect the chronology of delay events and to capture the critical path(s) dynamics. The proposed multiple-window MBF analysis is an enhanced version of the MBF analysis. The multi-window MBF analysis is a systematic step-by-step procedure that overcomes the traditional But-For analysis shortcomings. It has the ability to track critical path fluctuations and respect the chronological order of events, provided that the analysis windows are properly selected. Multi-window schedule analysis is highly sensitive to window size and may arrive at different results using different window sizes. The criteria set in the proposed methodology to select analysis windows is able to assure obtaining correct results under the adopted concurrency theory; however, for large complex projects, selecting the analysis windows to satisfy these conditions is not easy.

This chapter, therefore, introduced a multiple-window MBF methodology to allocate both schedule delays and accelerations and is capable of considering cases with multiple baseline updates. The application of the proposed methodology was illustrated using simple case studies. An extended case study is presented to demonstrate the application of these improvements.

Chapter 5

Enhanced Daily Windows Analysis Technique with Modified But-For Integration[3]

5.1 Introduction

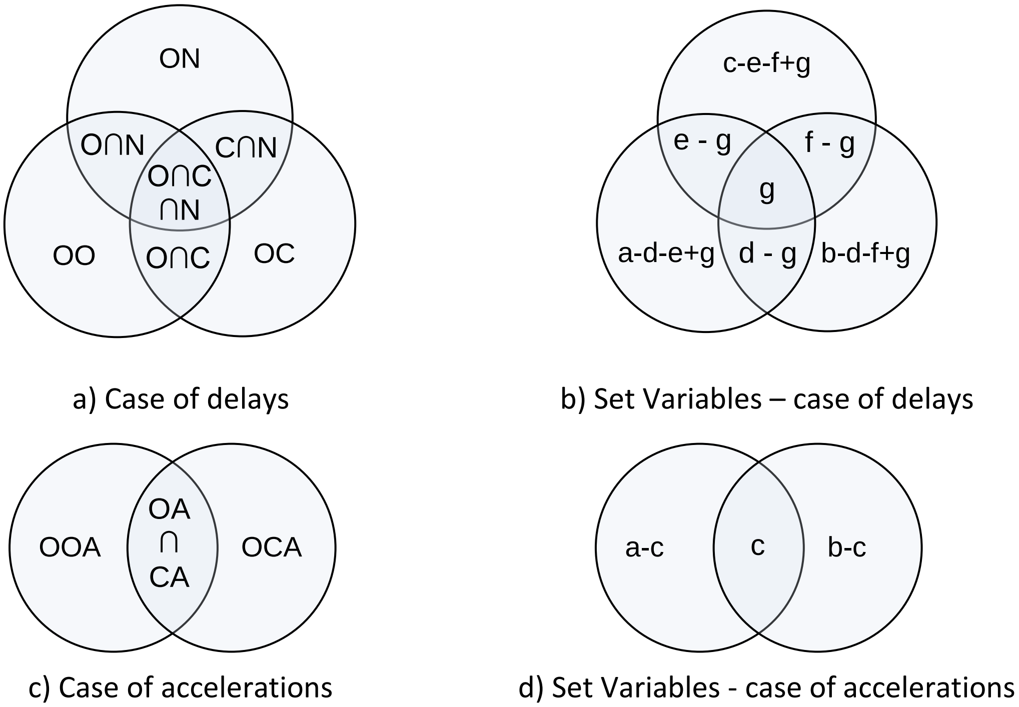
This chapter represents the third and the last component of the proposed framework that aims to integrate the most accurate version of windows analysis (daily windows analysis), with the enhanced Modified But-For method. The proposed technique benefits from the advantages of its parent techniques; it gains the high analysis resolution of the daily windows analysis and the ability of the modified but-for analysis to apportion concurrent delays/accelerations. In this chapter, the parent techniques are briefly introduced, then, the application of the proposed EDWA technique is explained on a simple case study. Afterwards, the ability of the EDWA to allocate responsibility in complex situations of extensive concurrent project delays and accelerations is demonstrated using complex progress scenarios. A step-by-step methodology is introduced to facilitate the EDWA application, followed by computer implementation supported by a validation case study.

5.2 Parent Techniques

Because the proposed technique integrates both DWA and Enhanced MBF methods, brief background and persistent drawbacks are introduced in this section. Windows analysis utilizes the as-planned schedule and adds delay events to it on time intervals called windows; the responsibility for project delays is then aggregated from all windows. However, its analysis results are highly sensitive to window size, as shown in chapter 2. Daily Windows Analysis (DWA) has resolved this sensitivity using

[3] This chapter is part of a published manuscript.
- Bhih, M., & Hegazy, T. (2021b). Enhanced Daily Windows Delay-Analysis (EDWA) Technique. ASCE-Journal of Legal Affairs and Dispute Resolution in Engineering and Construction, https://doi.org/http:\10.1061/(ASCE)LA.1943-4170.0000475

a fixed window size of one day (Hegazy & Zhang, 2005). However, DWA is still unable to handle complex cases that involve simultaneous delay and acceleration events. To address this persistent shortcoming, DWA is combined with the latest developments in But-For analysis (enhanced MBF) presented in chapter 4. The modified but-For methodologies for schedule delays and accelerations are presented in Figure 5.1, Table 5.1, and Table 5.2.



a) Case of delays

b) Set Variables – case of delays

c) Case of accelerations

d) Set Variables - case of accelerations

* OO= Only Owner; OC = Only Contractor; ON = Only third party
OOA= Only Owner Acceleration; OCA = Only Contractor Acceleration

Figure 5.1 Venn representation of project delays and accelerations

While the enhancement to MBF addresses key issues, one drawback still persists, which is its inability to consider the chronological order of events and the fluctuation of the critical path(s) due to the single window implementation, which DWA can do. As such, both DWA and EBFM can complement each other and can meet the requirements of recent standards such as the ASCE

Schedule Delay Analysis standard (ASCE 67-17, 2017); and AACE recommended practice (AACE RP 29R-03, 2011).

Table 5.1 MBF calculations

Events Removed	Project Duration	Venn Calculations	Project Delay Responsibilities
O + C + N	T_1	$T_0^* - T_1$ = a+b+c-d-e-f+g	OO = $T_0 - T_2$
O	T_2	$T_0 - T_2$ = a-d-f-+g	OC = $T_0 - T_3$
C	T_3	$T_0 - T_3$ = b-d-e+g	ON = $T_0 - T_4$
N	T_4	$T_0 - T_4$ = c-e-f+g	O∩C = $T_2 + T_3 - T_0 - T_5$
O+C	T_5	$T_0 - T_5$ = a+b-d-e-f+g	O∩N = $T_2 + T_4 - T_0 - T_6$
O+N	T_6	$T_0 - T_6$ = a+c-d-e-f+g	C∩N = $T_3 + T_4 - T_0 - T_7$
C+N	T_7	$T_0 - T_7$ = b+c-d-e-f+g	O∩C∩N = $T_0 + T_5 + T_6 + T_7 - T_1 - T_2 - T_3 - T_4$

T_0^* = as-built duration

Table 5.2 MBF-acceleration calculations

Events Removed	Project Duration	Venn Calculations	Project Acceleration Responsibilities
O + C	T_1	$T_0^* - T_1 = a + b - c$	OOA = $T_1 - T_3$
O	T_2	$T_0 - T_2 = b - c$	OCA = $T_1 - T_2$
C	T_3	$T_0 - T_4 = a - c$	OA∩CA = $T_2 + T_3 - T_0 - T_1$

T_0^* = as-built duration

5.3 Enhanced Daily Windows Analysis (EDWA)

Based on the unique advantages of the DWA and EMBF analyses, the proposed technique aims to integrate the accurate multi-window technique of DWA and the ability to assess concurrent delays of multiple parties of the EMBF method. The proposed Enhanced Daily Windows Analysis (EDWA) is schematically shown in Figure 5.2. EDWA follows the analysis frequency of DWA to provide the most accurate monitoring of critical path(s) fluctuations as influenced by the event chronology. In the analysis of each day, as shown in Figure 5.2, the delay events until the current day are entered (left side of Figure 5.2), followed by this day's events, and accordingly, the remaining schedule as of this day is calculated (discussed in next section), as shown on the right side of the schedule. The consequent project duration as of this day's events is then calculated and the EMBF formulation in Tables 5.1 and 5.2 are used to allocate the delay and/or acceleration from the previous day's window to the individual and/or concurrent causing party(ies). The proposed EDWA method, as such, represents an integration of the DWA and the EMBF methods. EDWA, as such, is able to apportion both delays and acceleration even in complex situations of extensive simultaneous events, when one-day events produce more than one-day effect on the project completion date, using the EMBF calculations in each analysis window.

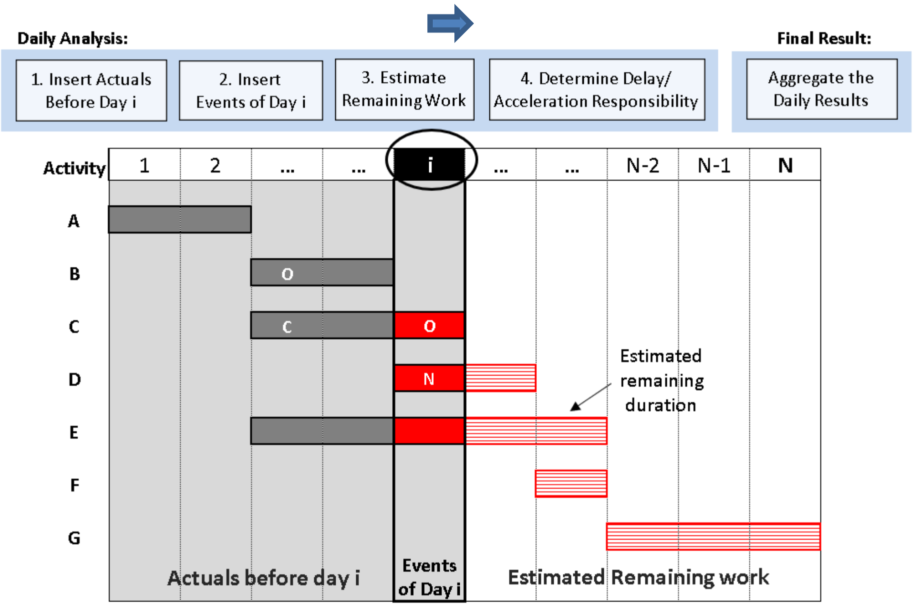

Daily Analysis:
1. Insert Actuals Before Day i
2. Insert Events of Day i
3. Estimate Remaining Work
4. Determine Delay/ Acceleration Responsibility
Final Result:
Aggregate the Daily Results
Activity
1
2
...
...
i
...
...
N-2
N-1
N
A
B
C
D
E
F
G
O
C
O
N
Estimated remaining duration
Actuals before day i
Events of Day i
Estimated Remaining work

Figure 5.2 Schematic representation of the proposed EDWA method

Table 5.3 shows the features of the EMBF, the DWA, as compared to the proposed EDWA features. It is important to highlight that forensic schedule analysis results are greatly affected by the accuracy and the level of detail of project schedules (as-planned, as-built, baseline updates). Inaccurate as-planned or as-built schedules will produce unfair allocation of project delays, thus leading to disputes among parties. In forensic schedule analysis, it is the analyst's role to examine the project documentation (minutes of meetings, correspondences, daily site reports, periodical reports, etc.) to quantify all delay events and construct an accurate and detailed as-built schedule. In literature, several efforts targeted improving as-built schedule accuracy; among those efforts are the intelligent bar chart by Hegazy et al. (2005), and the email-based system for documenting the as-built details by Hegazy & Abdel-Monem (2012)

Table 5.3 Feature comparison between existing and proposed techniques

Competent forensic analysis feature	Parent techniques		Proposed combined technique (EDWA*)
	EMBF*	DWA*	
Obtain same results irrespective of viewpoint	✓	✓	✓
Identify concurrent delay	✓	✓	✓
Apportion extensive** concurrent delay and acceleration	✓	×	✓
Respect event chronology	×***	✓	✓
Track critical path(s) fluctuation	×	✓	✓

* EMBF = Enhanced Modified But-For; DWA = Daily Windows Analysis; EDWA = Enhanced Daily Windows Analysis

**Extensive delays or accelerations happen when one day events cause more than one day effect on project completion.

*** EMBF respect chronology from one window to another, with difficulty in selecting window size for complex large projects.

5.3.1 EDWA: Estimating the Remaining Part of the Schedule

Because the EDWA uses daily analysis, the analysis of each day requires an assessment of the remaining part of the schedule after this day. However, this needs to be done consistently with the existing delay analysis guidance documents: the Delay and Disruption Protocol (SCL-DDP, 2017) of the Society of Construction Law in the UK; the AACE Recommended Practice for Forensic Delay Analysis (AACE RP 29R-03, 2011); and the ASCE Schedule Delay Analysis standard (ASCE 67-17, 2017).

AACE distinguishes two types of schedule analysis: Prospective vs Retrospective. The prospective analysis is to be performed before or contemporaneously with the delay event, i.e., while the project is still in progress and the delay effects are still unknown; thus relies on subjective best estimates of the remaining part of the schedule. On the other hand, retrospective analysis is to be performed after the delay event or after project completion, i.e., takes full advantage of the hindsight provided by the as-built information.

Prospective delay analysis is recommended by delay and disruption protocol (SCL-DDP, 2017) core principle 4 which recommends that time extension and compensation applications due to owner events should be submitted as close in time as possible to the event., i.e., before project completion. The "wait and see" practice regarding the impact of events is discouraged; therefore, the remaining duration of the activities should be estimated as no further information is available at the time of analysis. However, if time extension and compensation are assessed after the project completion or significantly after the delay event, the different recommended practices have different views. The (SCL-DDP, 2017) protocol, on one hand, recommends that the prospective analysis is no longer accepted and retrospective analysis should be used to benefit from the available as-built information. The AACE recommended practice (AACE RP 29R-03, 2011) also supports the use of retrospective analysis in the same situation but allows two options for the analysis, based on the analyst judgement: the hindsight and the blind-sight approaches. In the hindsight method, the analyst should use the actual as-built data in the remaining part of the activities. On the other hand, the blind-sight analyst assumes no access to the as-built information and estimates the remaining parts of the activities considering the project circumstances at the time the event happened. This blind-sight approach is also required by Guideline 7.3 of the ASCE proposed standard (ASCE 67-17, 2017). The guideline argues that since the analysis is often conducted after-the-fact, this may reveal what may be considered bad decisions in the light of the as-built schedule. However, those decisions may be still justifiable in absence of as-built schedule.

For the proposed EDWA method to be applicable both during construction and after project completion, and consistent with all guidelines, the retrospective hindsight method is adopted for the analysis after project completion if the as-built information is accessible. For the analysis during construction, the prospective method is adopted in this study, with the remaining parts of the activities estimated based on the state of information at the time of the events. For this approach, the formulation of the remaining duration suggested by Hegazy & Petzold (2004) is adopted to calculate the remaining part of the schedule (right side of the schedule in Figure 5.2), as follows:

Activity Remaining Duration = $(1 - P)$ * Planned Duration / f

Where, P is the activity percentage complete to-date, and f is the seasonal productivity factor associated with the construction season or project-specific conditions. It is noted that this calculation method matches what has been suggested by the AACE recommended practice (sec. 2.3) for calculating the remaining duration based on the as-planned progress rate, however, the adopted equation considers the productivity factor which is not considered by the AACE practice. The seasonal factor gives more flexibility to the contractor to account for the seasonal effects on its productivity based on the contractor's past experience on similar projects. It is represented as a fraction (e.g., 0.8) of the full productivity in the best working conditions.

5.4 EDWA Implementation

As illustrated in Figure 5.3, the proposed EDWA technique proceeds day by day. In each day (i), the as-built schedule till Day (i - 1) is entered, followed by the events of the current Day (i). Afterwards, the remaining schedule is calculated, and accordingly, the project duration is evaluated. If the project duration exhibits a delay from the analysis of the previous day, then, the responsibility for this delay is assessed using the computation in Table 5.1. If, on the other hand, the project duration exhibits an acceleration from the analysis of the previous day, then, the responsibility for this acceleration is assessed using the computation in Table 5.2. The same procedure is then repeated for every day of the project duration. Once all days are assessed, the results are aggregated by adding the responsibilities of the individual days.

Case Study 1

To demonstrate the application of the EDWA analysis, a small case study of four activities has been analyzed. As-planned and as-built schedules are shown in Figure 5.4. The project as-planned duration is 8 days; while the project as-built duration is 11 days. In this case study, the owner and contractor events are not occurring simultaneously, thus, true-concurrency does not exist. The MBF analysis details is shown in Table 5.4, and arrived at a delay responsibility of (OO= 1 day, O∩C = 2 days), showing a concurrency of 2 days, which is clearly wrong result since all events are not simultaneous. The correct results can be obtained by considering event chronology in the analysis. On Days 4 and 5, the contractor suspended the work in activity C causing 2 days project delay due to the contractor (OC = 2 days) and creating a float of 2 days to the other critical path (path A B). Afterwards, the owner event to activity B consumed the 2 days float and caused further 1 day project delay due to the owner(OO = 1 days), thus the correct result, in this case, is (OO= 1 day, OC = 2 days). In the MBF single-window analysis, the whole project period is considered as one analysis window, thus any two delay events by different parties on parallel critical paths will be captured as simultaneous events regardless of its timings; thus, the resulting project delays will be allocated as concurrent delays. The wrong result was obtained, in this case, because the contractor events (Days 4 and 5) and the owner events (Days 7 and 8) were treated as simultaneous despite the difference in its timing; this was due to the inability of the MBF to consider event chronology. The EDWA analysis details, on the other hand, is shown in Table 5.5, where only five daily windows were considered in the analysis (Days 4, 5, 7, 8, and 9) as there are no events on the other days. The analysis obtained the correct result of (OO=1 day, OC = 2 days) and fully respected event chronology. The EDWA, through its daily analysis windows, is able to analyze the effects of delay events cumulatively and chronologically and to track critical-path dynamics. This proves its superiority over the MBF which analyzes the schedule as one window, and as such, only considers the final critical path, irrespective of event timings and their impact on critical-path fluctuations.

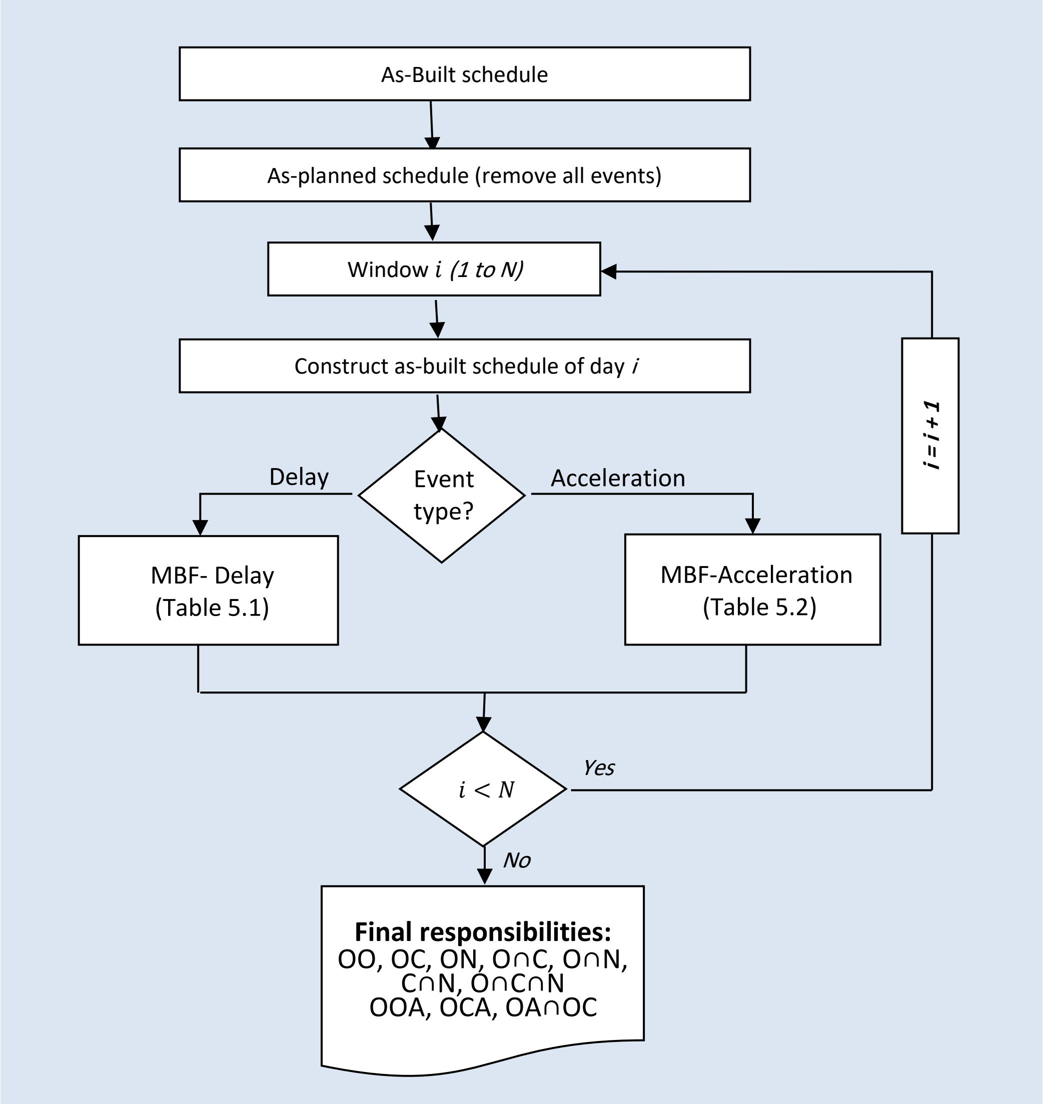
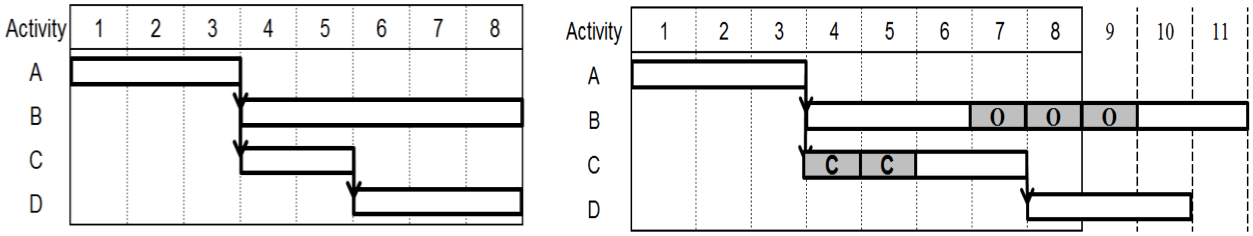


Figure 5.3 EDWA implementation



a) As-planned schedule

b) As-built schedule

Figure 5.4 Case study 1 schedules of

Table 5.4 Modified But-For (MBF) results – Case study 1

As-built	**As-built without party(ies) events (days)**							**Project delay responsibilities (days)**						
	O, C, N	O	C	N	O, C	O, N	C, N	OO	OC	ON	O∩C	O∩N	C∩N	O∩C∩N
11	8	10	11	11	8	10	11	1	0	0	2	0	0	0

Table 5.5 Enhanced Daily Windows Analysis (EDWA) results – Case study 1

Analysis Day	**Project Duration**	**As-built without party(ies) events (days)**							**Project delay responsibilities (days)**						
		O, C, N	O	C	N	O, C	O, N	C, N	OO	OC	ON	O∩C	O∩N	C∩N	O∩C∩N
4	9	8	9	8	9	8	9	8	0	1	0	0	0	0	0
5	10	9	10	9	10	9	10	9	0	1	0	0	0	0	0
7	10	10	10	10	10	10	10	10	0	0	0	0	0	0	0
8	10	10	10	10	10	10	10	10	0	0	0	0	0	0	0
9	11	10	10	11	11	10	10	11	1	0	0	0	0	0	0
									1	**2**	**0**	**0**	**0**	**0**	**0**

5.5 Apportioning Extensive Delays and Accelerations

In this section, further experiments were conducted to test the proposed EDWA methodology in terms of: the ability to handle cases of extensive delays and accelerations; demonstrate the analysis of concurrent delays and accelerations; and ability to handle cases that cannot be solved using daily windows.

Complex cases of extensive delays and accelerations can happen when the parties' events on some activities shift the successors to another period on time with less available resources, lower productivity rates, or different work calendars. In this case, the effect on project duration can be multiple delay days. On the other hand, acceleration events, by nature, can cause more than one day saving of project duration. Such situation is demonstrated in the following case study.

Case Study 2

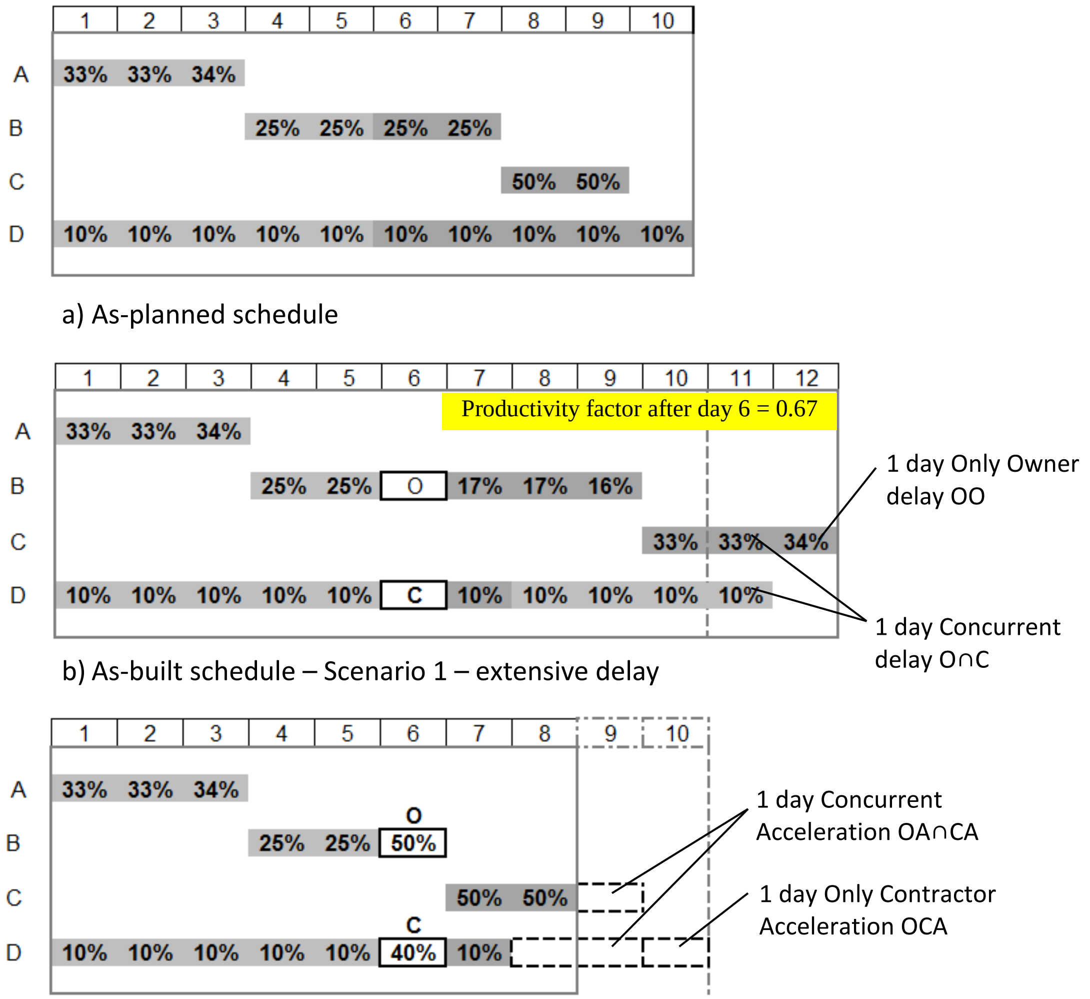
This case study (adopted from Zhang & Hegazy, 2005) has an as-planned duration of 10 days, (Figure 5.5a) and the two progress scenarios shown in Figure 5.5b,c, with extensive delays and extensive acceleration, respectively. The daily percentages on the figure show daily progress to help identify where slow progress or acceleration events occurred. In both scenarios, construction progressed as-planned for the first five days. On Day 6, Scenario 1 (Figure 5.5b) shows that activity B was delayed due to an owner event, while activity D was delayed due to a contractor event, simultaneously. Consequently, the work on activities B and C was shifted to a time period of productivity factor of (f = 0.76). For activity B, the percentage complete up to Day 6 is P = 50%, planned duration is 4 days, thus the remaining duration of activity B is calculated using equation (1):

Activity B Remaining Duration = (*1 – 0.5*) * 4 / *0.67 = 3 days*

For activity C, the percentage complete up to Day 6 is P = 0%, planned duration is 2 days, thus the remaining duration of activity C is:

Activity C Remaining Duration = (*1 – 0*) * 2 / *0.67 = 3 days*

With the remaining work of activities B and C is three days after Day 6, the two events on day 6 caused 2-day project delay to day 12. The EDWA analysis was then applied to this scenario; with only the window of Day 6 considered, as no other events exist in the other days. The EDWA results are shown in Table 5.6. The results show a responsibility of 1 day due to owner and 1 day concurrent delay, i.e., OO = 1 day, and O∩C = 1 day. These results can be logically inferred by observing the difference between the as-built and the as-planned schedules. From Figure 5.5b, it can be easily observed that the first delay day is attributed to both paths and hence to the owner and the contractor concurrently. The second delay day is attributed only to the top path, where the owner event occurred; thus, it is owner responsibility. The results are also identical to those obtained by (K. Zhang & Hegazy, 2005) considering the total-float changes on the different paths.


1 2 3 4 5 6 7 8 9 10
A 33% 33% 34%
B 25% 25% 25% 25%
C 50% 50%
D 10% 10% 10% 10% 10% 10% 10% 10% 10% 10%
1 2 3 4 5 6 7 8 9 10 11 12
Productivity factor after day 6 = 0.67
A 33% 33% 34%
B 25% 25% O 17% 17% 16%
C 33% 33% 34%
D 10% 10% 10% 10% 10% C 10% 10% 10% 10% 10%
1 day Only Owner delay OO
1 day Concurrent delay O∩C
1 2 3 4 5 6 7 8 9 10
A 33% 33% 34%
B 25% 25% O 50%
C 50% 50%
D 10% 10% 10% 10% 10% C 40% 10%
1 day Concurrent Acceleration OA∩CA
1 day Only Contractor Acceleration OCA

a) As-planned schedule

b) As-built schedule – Scenario 1 – extensive delay

c) As-built schedule – Scenario 2 – extensive acceleration

Figure 5.5 Case study 2 schedules

Table 5.6 EDWA results of Case study 2 –Scenario 1

Scenario 1		As-built without party(ies) events (days)							Project delay responsibilities (days)						
Day	As-built	O, C, N	O	C	N	O, C	O, N	C, N	OO	OC	ON	O∩C	O∩N	C∩N	O∩C∩N
6	12	10	11	12	12	10	11	12	1	0	0	1	0	0	0
									1	**0**	**0**	**1**	**0**	**0**	**0**

Progress Scenario 2 (Figure 5.5c), on the other hand, has two simultaneous acceleration events on Day 6; owner-directed acceleration of activity B, and contractor-voluntary acceleration of activity D. Applying the EDWA analysis on this scenario produced the results in Table 5.7; only Day 6 was considered in the analysis as there are no events on the other days. From Figure 5.5c, the as-built schedule exhibits 2 days acceleration; also it can be easily observed that 1 day acceleration is attributed to activity D and hence to the contractor while the other day is concurrent acceleration due to both parties. The result in Table 5.7 shows the same results of (OCA= 1 day, OA∩CA= 1 day). It is important to mention that due to the simplicity of this case study, it was possible to verify the results by observing the schedule, however, for large and complex projects, this may not be an easy task. The two scenarios of Case study 2, however, proved that EDWA methodology is able to handle complex situations of delays only or accelerations only. The next case study will demonstrate the case of both simultaneously.

Table 5.7 EDWA results of Case study 2 –Scenario 2

Scenario 2		As-built without party(ies) events (days)			Project acceleration responsibilities (days)		
Day	As-built	O, C	O	C	OOA	OCA	OA∩CA
6	8	10	9	10	0	1	1
					0	**1**	**1**

Case study 3 (simultaneous extensive delays and accelerations)

As explained earlier, the EDWA has two distinct analysis calculations for delay and acceleration cases: EDWA-delay and EDWA-acceleration, respectively. Case study 3 represents a more complex situation where delay and acceleration events by different parties are happening on the same day to show how the EDWA methodology can be applied in such a generic situation. The as-built schedule of Case study 2 was altered to a case of simultaneous delay and acceleration of more than one day effect as shown in Figure 5.6. The project has the same 10-day as-planned schedule of Figure 5.5a. Day 6, exhibits two simultaneous events: owner delay on activity B and contractor voluntary acceleration on activity D, with net effect being 2-day project delay. As this project has only events on Day 6, the other days will not be considered in the analysis as it will produce no responsibilities. For Day 6, two-step analysis process has been introduced to apply EDWA methodology as illustrated in the following subsection. To fairly apportion delay and acceleration responsibilities of Day 6, the effects of the two events are first untangled by decomposing and analyzing the acceleration and delay events separately, then aggregating the final result.

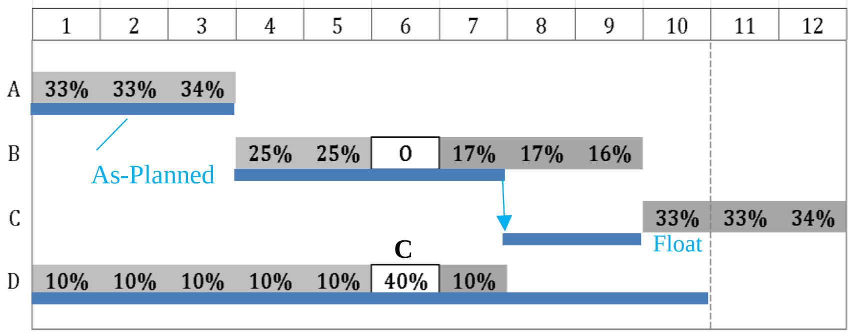
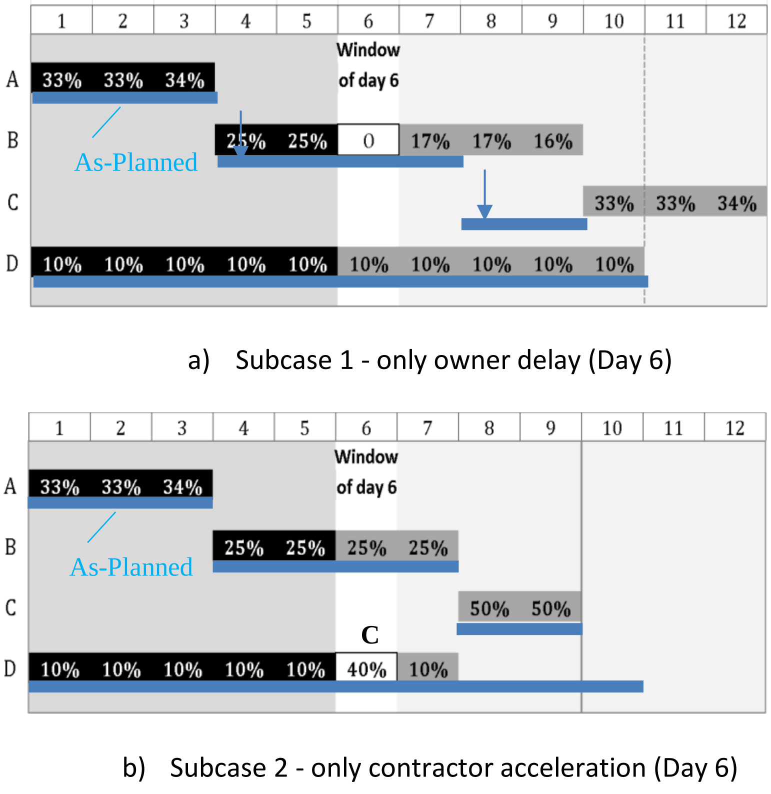


Figure 5.6 As-built versus as-planned schedules of Case study 3

Step 1: Separation of Events on Day 6

The as-built schedule on Day 6 was decomposed into two schedules, each one includes one of the events separately, as shown in Figure 5.7. In Figure 5.7a, the subcase of owner delay exhibits two days project delay. Applying the EDWA-delay analysis to this subcase is shown in Table 5.8. The analysis allocated the 2 days project delay to Only Owner (OO = 2 days); the same result can be observed easily from the as-built schedule of Figure 5.7a. Conversely, the subcase of contractor acceleration (Figure 5.7b) exhibits one day project acceleration. Applying the EDWA-acceleration analysis to this acceleration subcase is shown in Table 5.9 and arrived at a 1-day Only Contractor Acceleration (OCA = 1 day). This result can be observed from the as-built schedule of Figure 5.7b.



a) Subcase 1 - only owner delay (Day 6)

b) Subcase 2 - only contractor acceleration (Day 6)

Figure 5.7 Case study 3 – decomposed as-built schedules of Day 6

Table 5.8 EDWA results – Case study 3 (Subcase of delay)

Subcase of delay		**As-built without party(ies) events (days)**							**Project delay responsibilities (days)**						
Day	As-built	O, C, N	O	C	N	O, C	O, N	C, N	OO	OC	ON	O∩C	O∩N	C∩N	O∩C∩N
6	12	10	10	12	12	10	10	12	2	0	0	0	0	0	0

Table 5.9 EDWA results – Case study 3 (subcase of acceleration)

subcase of acceleration		**As-built without party(ies) events (days)**			**Project acceleration responsibilities (days)**		
Day	As-built	O, C	O	C	OOA	OCA	OA∩CA
6	9	10	9	10	0	1	0

Step 2: Result Compilation of Day 6

After analyzing the subcases individually, the final analysis result is not a simple algebraic summation of their results, and depends on: which party consumed the float; and if the float is a shared commodity or not. From the contractor's viewpoint, on Day 6, the contractor accelerated activity D on the bottom path (Figure 5.7b), which reduced the project duration from 10 days to 9 days. Afterwards, the owner delay caused the project to be extended from 9 days to 12 days (Figure 5.7a). Thus, project delay responsibility is (OO = 3 days, OCA = 1 day). Adopting this viewpoint means that the contractor owns the float. From the owner's viewpoint, on the other hand, the owner caused a delay to activity B (top path), which first consumed the existing 1 day float on the top path, and extended the project to 12 days, thus the delay responsibility is OO = 2 days, with contractor acceleration on the bottom path not affecting the project duration. Adopting this point of view means that the owner owns the float. The general rule of float ownership is that float can be consumed on first come first serve bases (AACE RP 29R-03, 2011), i.e., the float belongs to the party that uses it first (ASCE 67-17, 2017), unless float ownership is explicitly stipulated in the contract (SCL-DDP, 2017).

Considering the float as a shared commodity, and since both parties (owner and contractor) had started to consume the float in the same day, the 1-day float on the as-planned top path is to be shared between the two parties, 0.5-day each (Figure 5.8). Therefore, the contractor used its portion to save 0.5 days off the 10-day longest path (bottom) to become 9.5 days, while the owner consumed its portion and extended the project from 9.5 to 10 days. Afterwards, the owner event caused further 2-day extension of the project to day 12. Thus, the final result is (only owner OO = 2 + 0.5 = 2.5 days, and only contractor acceleration OCA = 0.5 day).

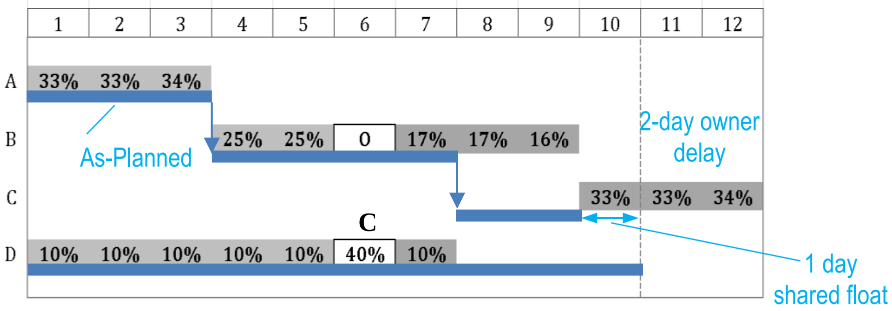


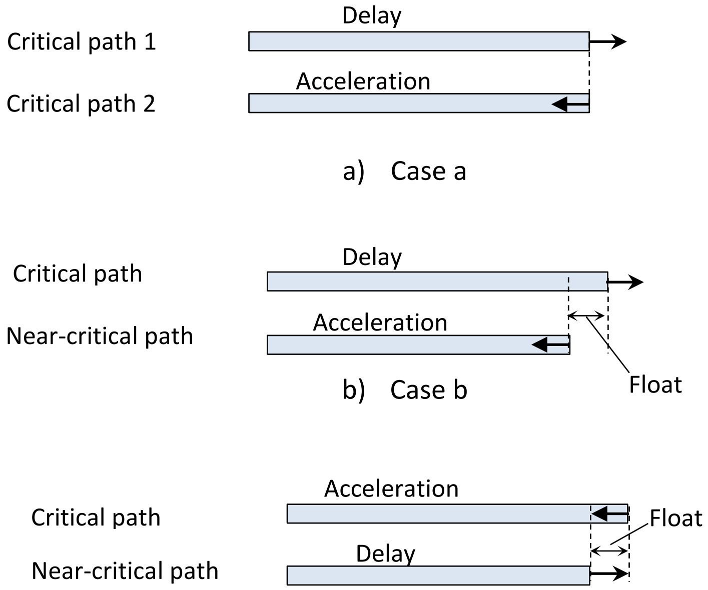
Figure 5.8 Case study 3 - concurrent delay and acceleration allocation

The above analysis shows that compiling the results of the individual subcases needs to consider the possible arrangements between the delay and acceleration subcases. To generalize, the following list of scenarios of delay and acceleration interactions are possible, as shown in the three cases of Figure 5.9, as follows:

a) Case a (Figure 5.9a): Both delay and acceleration events are on parallel critical paths. In this case, there is no float to be shared among parties; the acceleration will have no effect on the project completion date because the other critical path will prevent reducing project duration. The delay event is the dominant;

b) Case b (Figure 5.9b): The acceleration event is on the near-critical path while the delay event is on the critical path. In this case, the acceleration event will cause only an increase in the near-critical bath float without any effect to the project completion date. The delay event is the dominant; and

c) Case c (Figure 5.9c): the delay event is on the near-critical path while the acceleration event is on the critical path. Both parties will start consuming the float at the same time; the float is to be shared between them as demonstrated in Case study 3.



a) Case a

b) Case b

c) Case c

Figure 5.9 Interactions among simultaneous delay and acceleration events

As can be noted from case a and b (Figure 5.9a, b), the subcase with only-delay event will allocate the responsibility of the delay days to the party responsible for the delay event, while the subcase with only-acceleration event will produce no responsibility (zero acceleration days) as the project duration in this subcase will be controlled by the critical path with delay event. Thus, the final results in these two scenarios can be obtained simply by adding the results obtained from the subcases of only-delay and only-acceleration. On the other hand, in case c (Figure 5.9c) the subcase with only-delay event will allocate the responsibility of delay days to the party responsible for the delay event as if there is no acceleration on the critical path (i.e. ignoring the delay caused by consuming the time saved as a result of the acceleration of the other party). Furthermore, the subcase with only-acceleration event will allocate acceleration to the party responsible for the acceleration event as if this party owns the near-critical path float. To consider the float ownership in the analysis, as explained in detail in Case study 3, the acceleration days produced from the subcase with only-acceleration event should be shared equally between the two parties (i.e. 50% of the acceleration days are assigned to the party responsible for acceleration event and the other 50% are added as delay responsibility to the party responsible for delay event).

Another important note is that the rule used to compile the results of the third scenario is also applicable to the first two scenarios. In cases a and b, as the only-acceleration subcase produces acceleration responsibility of zero days, splitting zero into two halves to share it among the two parties will not change the results. Thus, to simplify the compilation of results process, the rule applied in case three can be used as a general rule regardless of the case under consideration.

Detailed Analysis Process

In essence, the proposed EDWA methodology is summarized as follows:

1. The EDWA technique requires accurate and detailed as-planned and as-built schedules. Before starting the analysis, all delay events (acceleration, deceleration, and work suspension) should be identified, quantified, and shown on the as-built schedule. Any baseline updates, if any, are also required;
2. For each analysis Day i:
    a. If Day i exhibit either delay or acceleration events use the appropriate analysis calculation (delay or acceleration) to allocate responsibilities; go to the next day.

b. If Day i exhibits simultaneous delay and acceleration events, then:

i. Prepare separate as-built schedules for the subcases of only delay and only acceleration events;

ii. Analyze the subcases separately to obtain total delay days and total acceleration days;

iii. Assign delay and acceleration responsibilities as follows:

- Assign delay responsibility to the party responsible for delay event as: delay days (from only-delay subcase) + 1/2 acceleration days (from only-acceleration subcase); and
- Assign the acceleration responsibility to the party responsible for acceleration event as: 1/2 acceleration days (from only-acceleration subcase); go to the next day.

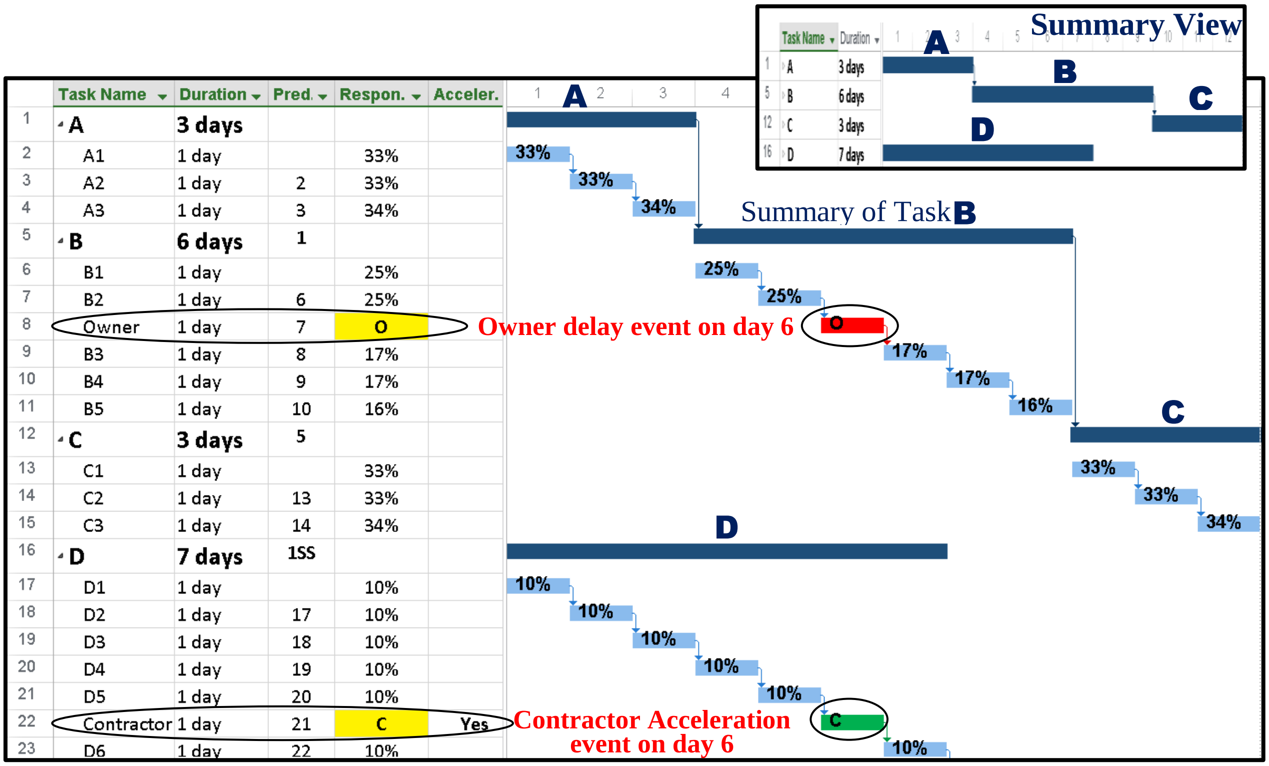
It is noted that only the owner and the contractor can share the float ownership and thus the acceleration days in this methodology are shared only if the delay was due to the owner or the contractor. If, on the other hand, the analysis results attribute delay days due to a third party, then the acceleration days will not be shared. This general methodology enables the EDWA to overcome an important shortcoming of the daily windows analysis, which is the inability to handle complex cases of extensive concurrent delay and acceleration.

5.6 EDWA Computer implementation

To automate the proposed EDWA analysis, a computer prototype was developed using the VBA macro language of Microsoft Project scheduling software (appendix A). Because the software does not allow mid-activity events such as acceleration or work interruption to be specified on specific dates, some manipulation was necessary, as shown in Figure 5.10, which represents the as-built information for Case study 3 of Figure 5.8. Each activity (e.g., A with 3-day duration) is represented by three daily sub-activities (A1, A2, and A3) equal to the activity duration, with a summary task and additional relations between the sub-activities. Also, two new activity attributes (Responsibility Column and Acceleration

Column) were added as placeholders for the event information. The responsibility column shows the identifier (O, C, or N) that represents a delay/acceleration on a specific day on any activity. The acceleration column also specifies if the event is acceleration in nature. With this data representation, the EDWA analysis follows the detailed analysis process discussed earlier.

The EDWA macro was applied to Case study 3 and the results are shown in Figure 5.11. In the beginning, the macro shows the as-planned and as-built durations along with the total project delay (Figure 5.11a). As the analysis proceeds to Day 6 where simultaneous delay and acceleration events occur, the macro recognizes this complex case and applies the EDWA methodology: events decomposition (Figure 5.11b), followed by result compilation (Figure 5.11c). The final responsibilities are identical to those obtained earlier using EDWA hand calculations.

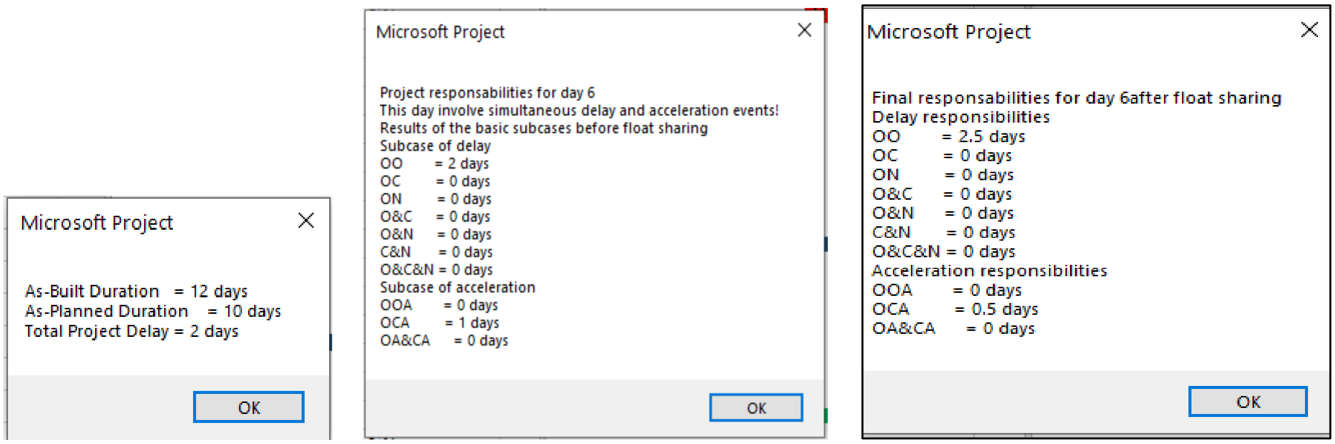


	Task Name	Duration	Pred.	Respon.	Acceler.
1	A	3 days			
2	A1	1 day		33%	
3	A2	1 day	2	33%	
4	A3	1 day	3	34%	
5	B	6 days	1		
6	B1	1 day		25%	
7	B2	1 day	6	25%	
8	Owner	1 day	7	O	
9	B3	1 day	8	17%	
10	B4	1 day	9	17%	
11	B5	1 day	10	16%	
12	C	3 days	5		
13	C1	1 day		33%	
14	C2	1 day	13	33%	
15	C3	1 day	14	34%	
16	D	7 days	1SS		
17	D1	1 day		10%	
18	D2	1 day	17	10%	
19	D3	1 day	18	10%	
20	D4	1 day	19	10%	
21	D5	1 day	20	10%	
22	Contractor	1 day	21	C	Yes
23	D6	1 day	22	10%	

	Task Name	Duration
1	A	3 days
5	B	6 days
12	C	3 days
16	D	7 days

Summary View
A
B
C
D
Summary of Task B
Owner delay event on day 6
O
Contractor Acceleration event on day 6
C
33%
33%
34%
25%
25%
17%
17%
16%
10%

Figure 5.10 Case study 3 implemented on Microsoft Project software



```
Microsoft Project

As-Built Duration   = 12 days
As-Planned Duration   = 10 days
Total Project Delay = 2 days
```

```
Microsoft Project

Project responsabilities for day 6
This day involve simultaneous delay and acceleration events!
Results of the basic subcases before float sharing
Subcase of delay
OO      = 2 days
OC      = 0 days
ON      = 0 days
O&C     = 0 days
O&N     = 0 days
C&N     = 0 days
O&C&N = 0 days
Subcase of acceleration
OOA      = 0 days
OCA      = 1 days
OA&CA     = 0 days
```

```
Microsoft Project

Final responsabilities for day 6after float sharing
Delay responsibilities
OO      = 2.5 days
OC      = 0 days
ON      = 0 days
O&C     = 0 days
O&N     = 0 days
C&N     = 0 days
O&C&N = 0 days
Acceleration responsibilities
OOA      = 0 days
OCA      = 0.5 days
OA&CA     = 0 days
```

a) Total project delay

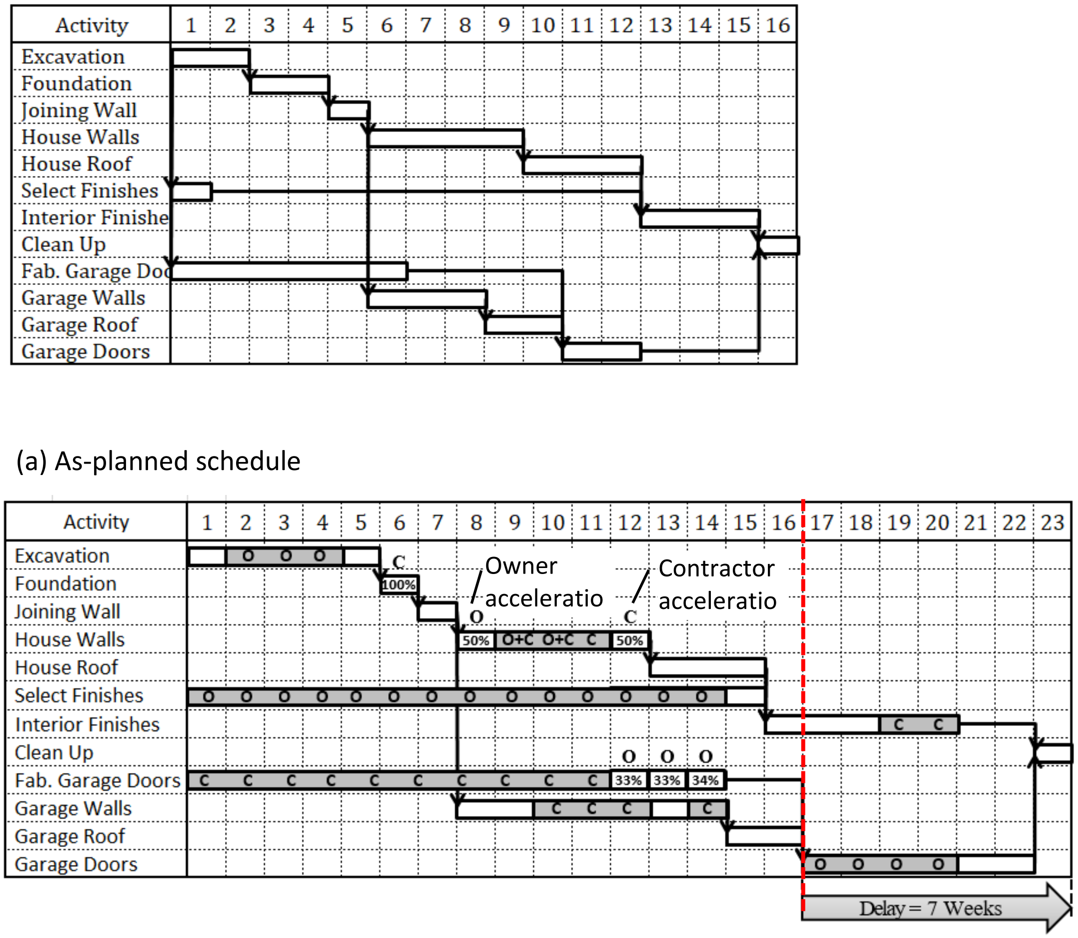
b) Day 6 events decomposition

c) Day 6 final responsibilities

Figure 5.11 Case study 3 - VBA-macro results

5.7 Validation Case

To further validate the developed application and demonstrate the practicality of the proposed EDWA approach, a case study was adapted from the literature (Stumpf, 2000) by adding owner-directed and contractor voluntarily accelerations. The as-planned and as-built schedules are shown in Figure 5.12. The project was planned to finish in 16 weeks while the as-built schedule shows that the construction was completed in 23 weeks, with 7 weeks behind schedule. The EDWA was applied manually to this project and the results in Table 5.10 show a delay responsibility of: 7 weeks due to Only Owner, 1 week due to Only Contractor, and 2 weeks of concurrent delays. On the acceleration side, it shows 1 week Only Owner acceleration; 1 week Only Contractor acceleration; and 1 week of concurrent acceleration. Therefore, the case study results are: OO = 7 weeks, OC = 1 week, O∩C = 2 weeks, OOA = 1 week, OCA = 1 week, OA∩CA = 1 week.



(a) As-planned schedule

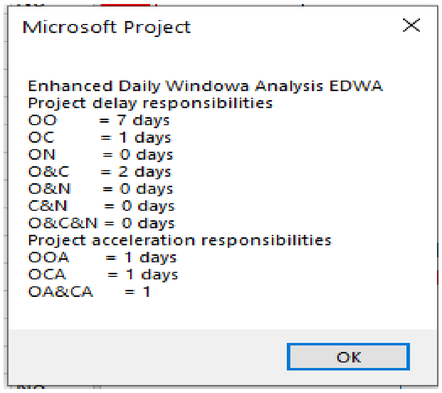
(b) As-built schedule

Figure 5.12 Validation case study schedules

The results obtained using the EDWA macro (Figure 5.13) are identical to those obtained using the manual EDWA procedure. In this case study, the total project delay shown in Figure 5.12 is 7 weeks, matching the summation of results (7+1+2-1-1-1 = 7 weeks), considering the acceleration as a negative delay. The manual calculation for this moderate-size case study tool a large time and effort and is error-prone. As such, the VBA-macro proved to be very useful, fast, and very suitable for larger and more complex cases.

Table 5.10 Validation case study EDWA analysis results

Validation case		As-built without party events (weeks)							Project delay responsibilities (weeks)							Project acceleration responsibilities (weeks)		
Day	As-built	O, C, N	O	C	N	O, C	O, N	C, N	OO	OC	ON	O∩C	O∩N	C∩N	O∩C∩N	OOA	OCA	OA∩CA
1	16	16	16	16	16	16	16	16	0	0	0	0	0	0	0			
2	17	16	16	17	17	16	16	17	1	0	0	0	0	0	0			
3	18	17	17	18	18	17	17	18	1	0	0	0	0	0	0			
4	19	18	18	19	19	18	18	19	1	0	0	0	0	0	0			
5	19	19	19	19	19	19	19	19	0	0	0	0	0	0	0			
6	18	19	18	19	-	-	-	-									1	
7	18	18	18	18	18	18	18	18	0	0	0	0	0	0	0			
8	17	18	18	17	-	-	-	-								1		
9	18	17	18	18	18	17	18	18	0	0	0	1	0	0	0			
10	19	18	19	19	19	18	19	19	0	0	0	1	0	0	0			
11	20	19	20	19	20	19	20	19	0	1	0	0	0	0	0			
12	19	20	20	20	-	-	-	-										1
13	19	19	19	19	19	19	19	19	0	0	0	0	0	0	0			
14	19	19	19	19	19	19	19	19	0	0	0	0	0	0	0			
15	19	19	19	19	19	19	19	19	0	0	0	0	0	0	0			
16	19	19	19	19	19	19	19	19	0	0	0	0	0	0	0			
17	20	19	19	20	20	19	19	20	1	0	0	0	0	0	0			
18	21	20	20	21	21	20	20	21	1	0	0	0	0	0	0			
19	22	21	21	22	22	21	21	22	1	0	0	0	0	0	0			
20	23	22	22	23	23	22	22	23	1	0	0	0	0	0	0			
21	23	23	23	23	23	23	23	23	0	0	0	0	0	0	0			
22	23	23	23	23	23	23	23	23	0	0	0	0	0	0	0			
23	23	23	23	23	23	23	23	23	0	0	0	0	0	0	0			
									7	**1**	**0**	**2**	**0**	**0**	**0**	**1**	**1**	**1**



Microsoft Project

```
Enhanced Daily Windowa Analysis EDWA
Project delay responsibilities
OO       = 7 days
OC       = 1 days
ON       = 0 days
O&C      = 2 days
O&N      = 0 days
C&N      = 0 days
O&C&N = 0 days
Project acceleration responsibilities
OOA      = 1 days
OCA      = 1 days
OA&CA    = 1
```

Figure 5.13 Validation case study - VBA-macro results

5.8 Summary

But-For and Window analysis techniques have been used extensively in practice to prepare and review compensation and extension of time claims due to schedule delays/accelerations. Each of these two techniques, however, has its pros and cons. The traditional But-For technique has been criticized for its inconsistent results when applied from different parties' viewpoints, and for its inability to identify concurrent delays. Although the Modified But-For (MBF) method resolved these issues and could identify concurrent delays using Venn representation, it still suffers from a serious drawback due to its single-window application that is insensitive to the chronology of events and critical path fluctuations. Such drawback does not exist in the Daily Window Analysis (DWA) technique which uses a window size of one day to be able to capture all critical path fluctuations. Yet, DWA has no mechanism to identify concurrent delays and cannot address situations when extensive delay and accelerations events are involved. Thus, this research proposed the EDWA as an innovative hybrid technique that combines the computational benefits of both the DWA and the MBF techniques to accurately allocate both delays and accelerations among project parties. The EDWA combines the high analysis resolution of the daily window analysis and the accurate allocation of concurrent delays/accelerations of the MBF method. The application of EDWA methodology was demonstrated on case studies; the obtained results showed that the EDWA excels over its parent techniques. The proposed methodology proved that it can handle more complicated scenarios and obtain accurate and equitable results that match the requirements of industry guidance documents. A systematic and structured procedure has been presented to facilitate accurate application and a VBA macro on MS Project was written to minimize the manual work needed. The computer implementation was validated using a practical case study from the literature.

Chapter 6

Conclusions and Future Research

6.1 Summary

Schedule delays and accelerations are common in construction projects. At the beginning of the project, the as-planned schedule is to be prepared by the contractor to reflect its best estimate of how the work execution will progress. However, it is unlikely that the work will be executed according to the plan, rather, with the work moving forward in the project, delays might happen due to the actions or inactions of project parties. To recover such delays, the contractor may decide, or the owner may direct, an acceleration by expediting the remaining work to avoid compensating the other party or mitigate the unfavorable delay in delivering the project. Moreover, project parties may agree to revise the as-planned schedule when more information is available throughout the course of the project leading to changes in the project baseline. In such situations, project parties will be in conflicting positions asking/defending time extension and/or compensation claims; forensic schedule analysis techniques are the tool to support claims and fairly settle construction disputes.

Several delay analysis techniques have been used in practice and proposed in the literature. The common four techniques used in practice are: as-planned versus as-built, impacted as-planned, collapsed as-built (But-For), and windows analysis. Among those, the But-For and Windows analysis are widely used and accepted by courts and arbitrators. Improved versions of those techniques have been introduced in the literature to address some of its shortcomings. Despite the increasing research efforts in this area, shortcomings, limitations, and persistent issues still exist. Thus, this research has targeted improving the But-For and windows analysis techniques, as the most used techniques, to overcome its drawbacks. Both techniques, then, were integrated to complement each other and produce a more accurate analysis tool that overcome the drawbacks of the original techniques.

But-For analysis is one of the widely used and accepted techniques despite its known drawbacks; the reason behind this acceptance is its simple methodology. The delay analysis results of But-For method have traditionally been misinterpreted. Due to this misinterpretation, traditional But-For analysis is unable to identify concurrent delays and produces conflicting results when adopting different party's viewpoints. Thus, it can suit limited situations when concurrency does not exist. Furthermore, due to its single window implementation, the traditional But-For, lacks the ability to track critical path(s) fluctuations and to respect event chronology. In chapter 3, the correct application of But-For analysis has been presented to identify concurrent delays and reconcile the conflicting parties' viewpoints. However, even with this correct analysis, using But-For with a single analysis window produces results that do not differentiate between truly concurrent delays and concurrent effects. To further improve the But-For analysis, a process of applying correct But-For analysis with properly selected multiple windows was proposed. The developments made provide explicit and structured details to correctly implement the But-For analysis in a manner that matches the general delay analysis guidelines of international organizations such as AACE. Alternatively, for larger projects, delay analysis needs a more rigorous analysis method than But-For.

Due to the traditional But-For drawbacks, an enhancement to But-For, namely the MBF method, has been introduced in the literature; it uses Venn representation to produce consistent results and to identify concurrent delays. Similar to traditional But-For, due to its single-window implementation, the MBF method suffered from the inability to respect the chronology of delay events and to capture the critical path(s) dynamics. Another limitation of the MBF method is that it was developed only for the case of schedule delays, overlooking the common schedule accelerations.

In chapter 4, an enhanced MBF methodology was proposed to allocate both schedule delays and accelerations. Multiple-window MBF analysis is also introduced to overcome the But-For analysis's inability to track critical path fluctuations and respect the chronological order of events. Multi-window schedule analysis is highly sensitive to window size and may arrive at different results using different window sizes. The criteria set in the proposed methodology to select analysis windows can assure arriving at correct results under the adopted concurrency theory; however, for large complex projects, selecting the analysis windows to satisfy these conditions is not easy. Therefore, the introduced enhanced MBF methodology is able to allocate both schedule delays and accelerations considering cases with multiple baseline updates.

The MBF and Daily Windows Analysis (DWA) have its pros and cons. The enhanced version of But-For analysis, namely the enhanced MBF method, eliminated some of its drawbacks. With its superiority in allocating concurrent delays and accelerations in the general case of three parties, the enhanced MBF methodology was a strong candidate to be combined with other analysis techniques to enhance its concurrency assessment. On the other hand, the DWA technique which uses a window size of one day is able to capture all critical path fluctuations on daily basis. Yet, DWA has no mechanism to identify concurrent delays and cannot address situations when extensive delay and accelerations events are involved. Thus, chapter 5 of this research proposed the Enhanced Daily Windows Analysis (EDWA) as an innovative hybrid technique that combines the computational benefits of both the DWA and the enhanced MBF techniques to accurately allocate both delays and accelerations among project parties. The EDWA combines the high analysis resolution of the daily window analysis and the accurate allocation of concurrent delays/accelerations of the MBF method. The proposed methodology proved that it can handle more complicated scenarios and obtain accurate and equitable results that match the requirements of industry guidance documents. A systematic and structured procedure has been presented to facilitate accurate application and a VBA macro on MS Project was written to minimize the manual work needed.

With the ability to allocate responsibilities on daily basis, the EDWA will produce the most accurate and equitable results among all available techniques considering the state of information available at the time of analysis. Nevertheless, developing a more accurate approach of progress monitoring on a daily basis by utilizing recent techniques of progress tracking will reduce disputes among parties and produce more fair results. The EDWA calculations are based on the schedule durations obtained after removing certain combinations of events from the as-built schedule. Thus, the capabilities of the used software to accommodate different types of activities' relationships will dictate the types of relationships which can be used in the model. Further validation of the EDWA on real-life cases is needed, however, this needs collaboration work with delay analysis practitioners considering the confidential nature of construction claims information.

6.2 Conclusions

The developments introduced in this research to improve the current state of delay analysis techniques have led to the following conclusions:

- Having a more accurate and equitable delay analysis tool will help to reduce disputes among project parties.
- Some delay analysis techniques have been misused in literature by ignoring concurrent delays such as But-For and What-If analysis techniques.
- CPM-based delay analysis techniques assess delay responsibility based on its effect on the schedule completion date; it is the analyst's responsibility to obtain the correct results by observing the cause (delay event) and effect (project delay) relationship.

- In the multi-window delay analysis, the smaller the window size the more accurate the results and the higher the analysis resolution. The most accurate analysis can be done using one day window size such as the daily window analysis.
- Schedule acceleration analysis is different from schedule delay analysis. In contrast to delays, using the subtractive techniques, the acceleration responsibility should be assessed with respect to the as-planned schedule, while delays are commonly assessed from the as-built schedule.
- Special care should be given to the different definitions of concurrent delays in practice, analysis results are highly dependent on the adopted definition which will affect

6.3 Contributions

This work enhances the current state of forensic schedule analysis by introducing significant improvements to the widely-used delay analysis techniques (But-For and Windows analysis). The introduced improvements are expected to improve the corrective-action decisions during construction and to fairly review claims and settle disputes during construction and after project completion. It also provides efficient forensic analysis tools to achieve mutual and smooth settlement of claims, and avoid costly and time-consuming disputes. The contributions of this work, for both research and practice, are as follows:

- **Better Understanding of forensic schedule analysis:** This research has provided an extensive literature review on the commonly used delay analysis techniques and its shortcomings. Gaps and areas of potential improvement have been identified.
- **Better Understanding of schedule delay assessment:** This study has resulted in a better understanding of the correct schedule delay assessment based on both cause and effect by

respecting event chronology considering adopted concurrency theory. Selecting the analysis-windows arrangement properly helped to enhance concurrency assessment, avoid wrong assessment of delays, and differentiate between truly concurrent delays and concurrent effects.

- **Concurrency assessment under different concurrency theories:** Concurrent delays have varying definitions by forensic analysis practitioners and industry guidance documents. This reflects the differing viewpoints that originated from the inconsistent application of those definitions in practice. This study has produced a clear understanding and application of delay analysis techniques under the different definitions.
- **Improving traditional But-For analysis:** The common misinterpretation of traditional But-For analysis has been corrected; this resulted in enabling the technique to identify concurrent delays and producing consistent results irrespective of the adopted viewpoint. In addition, the proposed multi-window Bu-For analysis has improved its concurrency assessment under the different theories.
- **Enhanced Modified But-For delay analysis method:** This research has revealed the difference in assessing schedule delays and schedule accelerations. A clear methodology to allocate schedule accelerations has been introduced using the modified But-For analysis method (MBF). The proposed EMBF methodology has the ability to allocate both schedule delays and accelerations among three project parties. The proposed enhancements have significantly improved its concurrency assessment by applying the EBMF multi-window analysis and enabled it to consider baseline updates.

- **Enhanced Daily Windows Analysis technique:** The innovative technique presented in this study has integrated the computational benefits of both the DWA and the EMBF techniques. The EDWA combines the high analysis resolution of the DWA and the accurate allocation of concurrent delays/accelerations of the EMBF method. The proposed methodology has the ability to handle complex cases of extensive simultaneous delay/acceleration events and obtain accurate and equitable results that match the requirements of the industry guidance documents. A step-by-step procedure has been presented with an MS-Project VBA-macro implementation.

6.4 Future Research

Several aspects of forensic schedule analysis could be improved by further studies. The current state of forensic schedule analysis shows the following limitations and areas of improvement:

- In literature researchers have also criticized What-IF analysis for producing inconsistent results when adopting different viewpoints and its inability to identify concurrent delays. Investigating the traditional application of What-If analysis for any common misconceptions to improve its implementation is an area of potential improvement.
- In literature, despite the efforts spent to enhance delay analysis techniques and propose new ones, no clear methodology was found for allocating accelerations among project parties. The clear methodology presented in this research of allocating schedule accelerations using the modified But-For method paves the way to extend ideas introduced to other delay analysis techniques.
- The availability of the information and documents about the project's actual progress may dictate or hinder the delay analysis process and cause disputes. Employing Real-time detection

technologies of any expected delay cause can enhance the prediction of future delays, reduce disputes among parties, and provide the necessary information required for fair allocation of delay/acceleration responsibilities. This area of improvement can be achieved using the available technologies, such as document management systems, time clock, and material and machinery tracking technologies. The utilization of such technologies will increase the resolution and accuracy of project recorded data without spending much more effort.

- Repetitive projects, including the three broad categories of linear, vertical, and scattered projects, have different scheduling needs. The critical path method (CPM) shows several limitations and shortcomings in scheduling repetitive projects. The line of balance (LOB) is the most commonly used method for repetitive projects scheduling and representation. None of the existing research efforts addressed the forensic analysis of repetitive construction projects. Extending delay analysis techniques to repetitive projects will make it applicable to a wide spectrum of projects including the big infrastructure constructions.
- The application of the presented improvements (correct But-For analysis, EMBF, EDWA) to real-life construction claims is an important part of the practical validation of the proposed methodologies.

References

(Ernst & Young LLP). (2017). *Spotlight on Canadian power and utility megaprojects: challenges in financing and delivery*.

AACE RP 10S-90. (2010). Cost Engineering Terminology. In *Practice*. Morgantown, WV, USA.

AACE RP 29R-03. (2011). *Forensic Schedule Analysis*. Morgantown, WV, USA: Association for the Advancement of Cost Engineering International.

Abd El-Razek, M. E., Bassioni, H. A., & Mobarak, A. M. (2008). Causes of Delay in Building Construction Projects in Egypt. *ASCE - Journal Construction Engineering and Management*, *134*(11), 831–841. https://doi.org/10.1061/_ASCE_0733-9364_2008_134:11_831

Al-Gahtani, K. S., Al-Sulaihi, I. A., & Iqupal, A. (2016). Total float management: computerized technique for construction delay analysis. *Canadian Journal of Civil Engineering*, (43), 391–401. https://doi.org/https://doi.org/10.1139/cjce-2015-0434

Al-Gahtani, K. S., & Mohan, S. B. (2007). Total float management for delay analysis. *Cost Engineering (Morgantown, West Virginia)*, *49*(2), 32–37. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-33846968505&partnerID=tZOtx3y1

Al-Khalil, M. I., & Al-Ghafly, M. A. (1999). Important causes of delay in public utility projects in Saudi Arabia. *Construction Management and Economics*, *17*(5), 647–655. https://doi.org/10.1080/014461999371259

Al-Momani, A. (2000). Construction delay: a quantitative analysis. *International Journal of Project Management*, *18*, 51–59. Retrieved from http://www.sciencedirect.com/science/article/B6V9V-40Y4JMT-

7/1/76499cc2c8a18bfa35b740d31975214d

Alizadehsalehi, S., & Yitmen, I. (2019). A Concept for Automated Construction Progress Monitoring: Technologies Adoption for Benchmarking Project Performance Control. *Arabian Journal for Science and Engineering*, *44*(5), 4993–5008. https://doi.org/10.1007/s13369-018-3669-1

Alkass, S., Mazerolle, M., & Harris, F. (1995). Computer Aided Construction Delay Analysis and Claims Preparation. *Construction Management and Economics*, *13*, 335–352.

Alkass, Sabah, Mazerolle, M., & Harris, F. (1996). Construction delay analysis techniques. *Construction Management and Economics*, *14*(5), 375–394. https://doi.org/10.1080/014461996373250

American Arbitration Association (AAA). (2013). *Annual Report & Financial Statements*.

Arcadis. (2020). *Global Construction Disputes Report 2020: Collaborating to achieve project excellence*. Retrieved from https://media.arcadis.com/-/media/project/arcadiscom/com/perspectives/middle-east/2020/global-construction-disputes-report-2020/me-global-construction-disputes-report-2020.pdf?rev=c2e77f1446744f559599cbbb1ddedd2f&hash=BEF742F40F03A7D8B651FF5DA6563045

Arditi, D., Akan, G. T., & Gurdamar, S. (1985). Reasons for delays in public projects in Turkey. *Construction Management and Economics*, *3*(2), 171–181. https://doi.org/10.1080/01446198500000013

Arditi, D., & Pattanakitchamroon, T. (2006). Selecting a delay analysis method in resolving construction claims. *International Journal of Project Management*, *24*, 145–155. https://doi.org/10.1016/j.ijproman.2005.08.005

Arditi, D., & Robinson, M. (1995). Concurrent delays in construction litigation. *Cost Engineering (Morgantown, West Virginia)*, *37*(7), 20–30.

Arif, F., & Morad, A. (2014). Concurrent delays in construction: International Legal Perspective. *ASCE-Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, *6*(1), 1–4. https://doi.org/10.1061/(ASCE)LA.1943-4170.0000134.

ASCE 67-17. (2017). Schedule delay analysis. In *American Society of Civil Engineers*.

Assaf, S. A., & Al-hejji, S. (2006). Causes of delay in large construction projects. *International Journal of Project Management*, *24*, 349–357. https://doi.org/https://doi.org/10.1016/j.ijproman.2005.11.010

Assaf, S. A., Al-Khalil, M., & Al-Hazmi, M. (1995). Causes of Delay in Large Building Construction Projects. *Journal of Management in Engineering*, *11*(2), 45–50. https://doi.org/10.1061/(asce)0742-597x(1995)11:2(45)

Baldwin, J., Manthei, J., Rothbart, H., & Harris, R. (1971). Causes of Delay in the Construction Industry. *Journal of Construction Division*, *97*(2).

Baram, G. E. (1994). Delay Analysis - Issues Not for Granted. *AACE International Transactions*, *DCL*(%), 1–9.

Baram, G. E. (2000). Concurrent Delays — What Are They and How to Deal With Them ? *AACE International Transactions*, 1–8.

Bhih, M., & Hegazy, T. (2019). Concurrent Delays : Comparison among Forensic Analysis Recommended Practices. *CSCE 7th International Construction Conference/Construction Research Congress 2019*, CON117-1–10. Laval, Quebec \, Canada.

Bhih, M., & Hegazy, T. (2020a). Improving But-For Delay Analysis and Concurrency Assessment. In M. Skibniewski & M. Hajdu (Eds.), *Proceedings of the Creative Construction e-Conference* (pp. 125–130). https://doi.org/10.3311/CCC2020-031

Bhih, M., & Hegazy, T. (2020b). Improving Concurrency Assessment and Resolving Misconceptions about the But-For Delay Analysis Technique. *ASCE-Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, *12*(2), 04520007 (1-10). https://doi.org/10.1061/(ASCE)LA.1943-4170.0000378

Bhih, M., & Hegazy, T. (2021a). Enhanced But-For Method (EBFM) to Apportion Net Delays and Accelerations. *ASCE - Journal Construction Engineering and Management*, *147*(7), 06021003 (1-5). https://doi.org/10.1061/(ASCE)CO.1943-7862.0002094

Bhih, M., & Hegazy, T. (2021b). Enhanced Daily Windows Delay-Analysis (EDWA) Technique. *ASCE-Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, *13*(3), 04521012 (1-12). https://doi.org/10.1061/(ASCE)LA.1943-4170.0000475

Bhih, M., & Hegazy, T. (2021c). Multiple-Window Modified But-For Analysis of Project Delays and Accelerations. *International Journal of Construction Management*. https://doi.org/10.1080/15623599.2021.1916708

Braimah, N. (2013). Construction Delay Analysis Techniques—A Review of Application Issues and Improvement Needs. *Buildings*, *3*(3), 506–531. https://doi.org/10.3390/buildings3030506

Bramble, B., & Callahan, M. (1987). *Construction delay claims*. New York (NY): Wiley Law Publications.

Bubshait, B. A. A., & Cunningham, M. (1998a). Determining Schedule Impact: Working Practice. *ASCE - Practical Periodical on Structural Design and Consruction*, *3*(4), 176–179.

Bubshait, B. A. A., & Cunningham, M. J. (1998b). Comparison of Delay Analysis Methodologies. *ASCE - Journal Construction Engineering and Management*, *124*(4), 315–322.

Canadian Construction Association. (2021). Canadian Construction Association website. Retrieved from https://www.cca-acc.com/about-us/value-of-industry/

Chan, D. W. M., & Kumaraswamy, M. M. (1996). An evaluation of construction time performance in the building industry. *Building and Environment*, *31*(6), 569–578. https://doi.org/10.1016/0360-1323(96)00031-5

Chen, G. X., Shan, M., Chan, A. P. C., Liu, X., & Zhao, Y. Q. (2019). Investigating the causes of delay in grain bin construction projects: the case of China. *International Journal of Construction Management*, *19*(1), 1–14. https://doi.org/10.1080/15623599.2017.1354514

Dale, W. S., & D'Onofrio, R. M. (2010). Reconciling Concurrency in Schedule Delay and Constructive Acceleration. *Public Contract Law Journal*, (39), 161–230.

Dale, W. S., & D'Onofrio, R. M. (2017). *Construction Schedule Delays* (2017 ed.). THOMSON REUTERS.

Dlakwa, M., & Culpin, M. (1990). Reasons for overrun in public sector construction projects in Nigeria. *International Journal of Project Management*, *8*(4), 237–241. https://doi.org/10.1016/0263-7863(90)90032-7

Ekanayake, B., Wong, J. K. W., Fini, A. A. F., & Smith, P. (2021). Computer vision-based interior construction progress monitoring: A literature review and future research directions. *Automation in Construction*, *127*(January), 103705. https://doi.org/10.1016/j.autcon.2021.103705

El-Adaway, I. H., Fawzy, S. a., Bingham, R., Clark, P., & Tidwell, T. (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), 02514001. https://doi.org/10.1061/(ASCE)LA.1943-4170.0000145

Faridi, A. S., & El-Sayegh, S. M. (2006). Significant factors causing delay in the UAE construction industry. *Construction Management and Economics*, *24*(11), 1167–1176. https://doi.org/10.1080/01446190600827033

Fawzy, S. a., & El-Adaway, I. H. (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. https://doi.org/10.1061/(ASCE)LA.1943-4170.0000104

Finke, M. (1997). Contemporaneous analyses of excusable delays. *Cost Engineering AACE*, *39*(12), 26–31.

Finke, M. (1999). Window analyses of Compensable Delays. *ASCE - Journal Construction Engineering and Management*, *125*(2), 96–100.

Frimpong, Y., & Oluwoye, J. (2003). Significant Factors Causing Delay and Cost Overruns in Construction of Groundwater Projects in Ghana. *Journal of Construction Research*, *04*(02), 175–187. https://doi.org/https://doi.org/10.1142/S1609945103000418

Fruchtman, E. (2000). Delay Analysis - Eliminaring the Smoke and Mirrors. *AACE International Transactions*, *CDR 06*, 1–4.

Golanaraghi, S., & Alkass, S. (2012). Modified Isolated Delay Type Technique. *Construction Research Congress 2012*, 90–99.

Gothand, K. D. (2003). Schedule delay analysis: Modified windows approach. *Cost Engineering*

(Morgantown, West Virginia), *45*(9), 18–23.

Guévremont, M., & Hammad, A. (2018). Visualization of Delay Claim Analysis Using 4D Simulation. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, *10*(3), 05018002 (1-8). https://doi.org/10.1061/(ASCE)LA.1943-4170.0000267

Guévremont, M., & Hammad, A. (2020). Review and Survey of 4D Simulation Applications in Forensic Investigation of Delay Claims in Construction Projects. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, *12*(3), 04520017 (1-9). https://doi.org/10.1061/(asce)la.1943-4170.0000391

Harris, R. A., & Scott, S. (2001). UK practice in dealing with claims for delay. *Engineering, Construction and Architectural Management*, *8*(5/6), 317–324. https://doi.org/https://doi.org/10.1108/eb021192

Hegazy, T., & Abdel-Monem, M. (2012). Email-based system for documenting construction as-built details. *Automation in Construction*, *24*, 130–137. https://doi.org/https://doi.org/10.1016/j.autcon.2012.02.014

Hegazy, T., Elbeltagi, E., & Zhang, K. (2005). Keeping Better Site Records Using Intelligent Bar Charts. *Journal of Construction Engineering and Management*, *131*(5), 513–521. https://doi.org/https://doi.org/10.1061/(ASCE)0733-9364(2005)131:5(513)

Hegazy, T., & Menesi, W. (2008). Delay Analysis under Multiple Baseline Updates. *Journal of Construction Engineering and Management*, *134*(8), 575–582. https://doi.org/10.1061/(ASCE)0733-9364(2008)134:8(575)

Hegazy, T., & Petzold, K. (2004). *Genetic Optimization for Dynamic Project Control*. *129*(August 2003), 396–404.

Hegazy, T., Said, M., & Kassab, M. (2011). Incorporating rework into construction schedule analysis. *Automation in Construction*, *20*(8), 1051–1059. https://doi.org/10.1016/j.autcon.2011.04.006

Hegazy, T., & Zhang, K. (2005). Daily Windows Delay Analysis. *Journal of Construction Engineering and Management*, *131*(5), 505–512. https://doi.org/https://10.1061/(ASCE)0733-9364(2005)131:5(505)

Ibbs, W., Nguyen, L. D., & Simonian, L. (2011). Concurrent Delays and Apportionment of Damages. *Journal of Construction Engineering and Management*, *137*(2), 119–126. https://doi.org/https://doi.org/10.1061/(ASCE)CO.1943-7862.0000259

Jonathan, B., Shi, J., Cheung, S. O., & Arditi, D. (2001). *Construction Delay Computation Method*. *127*(February), 60–65.

Kao, C. K., & Yang, J. Bin. (2009). Comparison of windows-based delay analysis methods. *International Journal of Project Management*, *27*(4), 408–418. https://doi.org/10.1016/j.ijproman.2008.05.016

Kim, Y., Kim, K., & Shin, D. (2005). Delay analysis method using delay section. *Journal of Construction Engineering and Management*, *131*(11), 1155–1164. https://doi.org/https://doi.org/10.1061/(ASCE)0733-9364(2005)131:11(1155)

Kopsida, M., & Vela, P. A. (2015). *A Review of Automated Construction Progress Monitoring and Inspection Methods Geometric Digital Twin Generation for Railway Infrastructure View project A Review of Automated Construction Progress Monitoring and Inspection Methods*. (June 2016). Retrieved from https://www.researchgate.net/publication/304013510

Koushki, P. A., Al-Rashid, K., & Kartam, N. (2005). Delays and cost increases in the construction of private residential projects in Kuwait. *Construction Management and Economics*, *23*(3), 285–

294. https://doi.org/10.1080/0144619042000326710

Kraiem, Z., & Diekmann, J. (1987). Concurrent delays in construction projects. *ASCE-Journa Construction Engineering and Management*, *113*(4), 591–602.

Le-Hoai, L., Lee, Y. D., & Lee, J. Y. (2008). Delay and cost overruns in Vietnam large construction projects: A comparison with other selected countries. *KSCE Journal of Civil Engineering*, *12*(6), 367–377. https://doi.org/10.1007/s12205-008-0367-7

Lee, J.-S. (2007). Delay Analysis Using Linear Schedule in Construction. *AACE International Transactions*, 1–6.

Lee, J., & Diekmann, J. E. (2011). Delay analysis considering production rate. *Canadian Journal of Civil Engineering*. https://doi.org/https://doi.org/10.1139/L11-006

Levin, P. (1998). *Construction contract claims, changes and dispute resolution* (2nd ed.; A. Press, Ed.). New York (NY).

Lifschitz, B. J., Barba, E. M., Lockshin, A. M., & Compa-, R. R. (2009). A Critical Review of the AACEI Recommended Practice for Forensic Schedule Analysis. *Construction Lawyer*, *29*(4), 1–10.

Livengood, J. (2017). Knowns and Unknowns of Concurrent Delay. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, *9*(3), 1–11. https://doi.org/https://doi.org/10.1061/(ASCE)LA.1943-4170.0000224.

Lo, T. Y., Fung, I. W., & Tung, K. C. (2006). Construction Delays in Hong Kong Civil Engineering Projects. *Journal of Construction Engineering and Management*, *132*(6), 636–649. https://doi.org/10.1061/(asce)0733-9364(2006)132:6(636)

Long, R. J. (2017). *As-Built But-For Schedule Delay Analysis List of Figures*. Retrieved from http://www.long-intl.com/wp-content/uploads/2020/07/Long_Intl_As-Built_But-For_Schedule_Delay_Analysis.pdf

Lovejoy, V. A. (2004). Claims schedule development and analysis: Collapsed as-built scheduling for beginners. *Cost Engineering (Morgantown, West Virginia)*, *46*(1), 27–30.

Magdy, M., & Georgy, M. (2019). Delay Analysis Methodologies Used by Engineering and Construction Firms in Egypt. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, *11*(3). https://doi.org/10.1061/(ASCE)LA.1943-4170.0000293

Mansfield, N., Ugwu, O., & Doran, T. (1994). Causes of delay and cost overruns in Nigerian construction projects. *International Journal of Project Management*, *12*(4), 254–260. https://doi.org/10.1016/0263-7863(94)90050-7

Mbabazi, A. (2004). *Quantification and Analysis of Constraction Claims*. https://doi.org/10.1177/001088048102200214

Mbabazi, A., Hegazy, T., & Saccomanno, F. (2005). Modified But-For Method for Delay Analysis. *Journal of Construction Engineering and Management*, *131*(10), 1142–1144. https://doi.org/10.1061/(ASCE)0733-9364(2005)131:10(1142)

McCullough, R. (1999). CPM Schedules in Construction Claims From the Contractor's Prespective. *AACE International Transactions*, *CDR*(02), 1–4.

Menesi, W. (2007). Construction delay analysis under multiple baseline updates (University of Waterloo). https://doi.org/10.1061/(ASCE)0733-9364(2008)134:8(575)

Nagata, M. (2018). Criticism of the ASCE Schedule Delay Analysis Offsetting Delay Concept. *AACE International 2018 Conference & Expo, Claims and Dispute Resolution*, 1–20. San Diego, CA, USA: AACE International.

Nguyen, L. D., & Ibbs, W. (2008). FLORA: New Forensic Schedule Analysis Technique. *Journal of Construction Engineering and Management*, *134*(7), 483–491. https://doi.org/https://doi.org/10.1061/(ASCE)0733-9364(2008)134:7(483)

Odeh, A. M., & Battaineh, H. T. (2002). Causes of construction delay: Traditional contracts. *International Journal of Project Management*, *20*(1), 67–73. https://doi.org/10.1016/S0263-7863(00)00037-5

Ogunlana, S. O., Promkuntong, K., & Jearkjirm, V. (1996). Construction delays in a fast-growing economy: Comparing Thailand with other economies. *International Journal of Project Management*, *14*(1), 37–45. https://doi.org/10.1016/0263-7863(95)00052-6

Okpala, D., & Aniekwu, A. (1988). CAUSES OF HIGH COSTS OF CONSTRUCTION IN NIGERIA. *ASCE - Journal Construction Engineering and Management*, *114*(2), 233–244.

Oliveros, A. V. O., & Fayek, A. R. (2005). Fuzzy Logic Approach for Activity Delay Analysis and Schedule Updating. *Journal of Construction Engineering and Management*, (January), 42–52.

Omar, T., & Nehdi, M. L. (2016). Data acquisition technologies for construction progress tracking. *Automation in Construction*, *70*, 143–155. https://doi.org/10.1016/j.autcon.2016.06.016

Sambasivan, M., & Soon, Y. W. (2007). Causes and effects of delays in Malaysian construction industry. *International Journal of Project Management*, *25*, 517–526. https://doi.org/https://doi.org/10.1016/j.ijproman.2006.11.007

Sandlin, L. S., Sapple, J. R., & Gautreaux, R. M. (2004). Phased root cause analysis - A distinctive view on construction claims. *Cost Engineering (Morgantown, West Virginia)*, *46*(6), 16–20.

Santoso, D. S., & Soeng, S. (2016). Analyzing Delays of Road Construction Projects in Cambodia: Causes and Effects. *Journal of Management in Engineering*, *32*(6), 05016020.

https://doi.org/10.1061/(asce)me.1943-5479.0000467

Schumacher, L. (1995). Quantifying and apportioning delay on construction projects. *Cost Engineering*, *37*(2), 11–13. Retrieved from https://search.proquest.com/docview/220469967/fulltextPDF/B931D5DFBDBD43CCPQ/1?accountid=45153

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

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

Scott, S. (1990). Keeping better site records. *International Journal of Project Management*, *8*(4), 243–249. https://doi.org/https://10.1016/0263-7863(90)90033-8

Semple, C., Hartman, F. T., & Jergeas, G. (1994). Construction Claims and Disputes: Causes and Cost/Time Overruns. *Journal of Construction Engineering and Management*, *120*(4), 785–795. https://doi.org/10.1061/(asce)0733-9364(1994)120:4(785)

Sgarlata, B. M. A., & Brasco, C. J. (2004). Successful claims resolution through an understanding of the law governing allocation of risk for delay and disruption. *Construction Management e Journal, CMAA*.

Stumpf, G. R. (2000). Schedule delay analysis. *Cost Engineering (Morgantown, West Virginia)*, *42*(7), 32–43.

Sweis, G., Sweis, R., Hammad, A. A., & Shboul, A. (2008). Delays in construction projects : The case of Jordan. *International Journal of Project Management*, *26*, 665–674. https://doi.org/https://doi.org/10.1016/j.ijproman.2007.09.009

Tsai, C., Wang, M., & Chang, L. (2015). Criticality index delay analysis method based on float allocation. *Journal of the Chinese Institute of Engineers*, *38*(7), 887–896. https://doi.org/https://doi.org/10.1080/02533839.2015.1037350

Tsai, M.-K., Yang, J.-B., & Yau, N.-J. (2013). Developing Computer-Based Schedule Delay Analysis Methods Based on Information Flow Analysis: a Case Study. *Journal of Civil Engineering and Management*, *19*(6), 823–835. https://doi.org/10.3846/13923730.2013.801901

Vasilyeva-lyulina, A. (2013). *The Study on Methodological Framework for Comparing Delay Analysis Methods in Construction Contracts*. KYOTO UNIVERSITY.

Wickwire, J., Driscoll, T., & Hurlbut, S. (1991). *Construction scheduling preparation, liability, and claims*. New York (NY): Wiley Law Publications.

Yang, J. Bin, Chu, M.-Y., & Huang, K.-M. (2013). An Empirical Study of Schedule Delay Causes Based on Taiwan's Litigation Cases. *Management*, *44*(3), 21–31. https://doi.org/10.1002/pmj

Yang, J. Bin, & Kao, C.-K. (2009). Review of Delay Analysis Methods: A Process-Based Comparison. *The Open Construction and Building Technology Journal*, *3*(1), 81–89. https://doi.org/10.2174/1874836800903010081

Yang, J. Bin, & Ou, S.-F. (2008). Using structural equation modeling to analyze relationships among key causes of delay in construction. *Canadian Journal of Civil Engineering*, *35*(4), 321–332. https://doi.org/10.1139/L07-101

Yang, J. Bin, & Tsai, M.-K. (2011). Computerizing ICBF Method for Schedule Delay Analysis. *Journal of Construction Engineering and Management*, *137*(8), 583–591. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000338

Yang, J. Bin, & Wei, P.-R. (2010). Causes of Delay in the Planning and Design Phases for Construction Projects. *Journal of Architectural Engineering*, *16*(2), 80–83.

https://doi.org/10.1061/(asce)1076-0431(2010)16:2(80)

Yang, J. Bin, Yang, C. C., & Kao, C. K. (2010). Evaluating schedule delay causes for private participating public construction works under the Build-Operate-Transfer model. *International Journal of Project Management*, *28*(6), 569–579. https://doi.org/10.1016/j.ijproman.2009.10.005

Yang, J. Bin, & Yin, P.-C. (2009). Isolated Collapsed But-For Delay Analysis Methodology. *Journal of Construction Engineering and Management-ASCE*, *135*(7), 570–578. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000016

Yusuwan, N. M., & Adnan, H. (2013). Issues Associated with Extension of Time (EoT) Claim in Malaysian Construction Industry. *Procedia Technology*, *9*, 740–749. https://doi.org/10.1016/j.protcy.2013.12.082

Zack, J. (1999). But-For Schedules - Analysis and Defense. *AACE International Transactions*, 1–6.

Zack, J. (2000). Pacing delays-the practical effect. *Cost Engineering (Morgantown, West Virginia)*, *42(7)*, 23–28.

Zafar, Z. Q. (1996). Construction project delay analysis. *Cost Engineering AACE*, *38*(3), 23–27.

Zhang, C., & Arditi, D. (2013). Automated progress control using laser scanning technology. *Automation in Construction*, *36*, 108–116. https://doi.org/10.1016/j.autcon.2013.08.012

Zhang, K., & Hegazy, T. (2005). Apportioning concurrent delays and accelerations using daily windows. *Construction Research Congress 2005: Broadening Perspectives - Proceedings of the Congress*, (519), 787–796.

Appendix A:

Microsoft-Project VBA-macro to implement EDWA

```
Sub EDWA()

‘ ******************************
‘ EDWA implementation MS VBA-macro
‘ Author: Moneer Bhih, 2020
‘ *******************************

Dim i, D, APD, TD, TA, DE, AC As Integer
Dim T0, T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11 As Integer
Dim R1, R2, R3, R4, R5, R6, R7, R8, R9, R10 As Integer
Dim T As Task
' As-Built Duration
    D = ActiveProject.Duration / ActiveProject.HoursPerDay / 60

' Constructing As-Planned Schedule
    For Each T In ActiveProject.Tasks
       If Not (T Is Nothing) Then
         If (T.Text1 = "O" Or T.Text1 = "C" Or T.Text1 = "N" Or T.Text1 = "O&C") And (T.Flag3 = False)
     Then
         T.Duration = 0
         End If
         If (T.Text1 = "" And T.Flag1 = True) Or (T.Text1 = "" And T.Flag2 = True) Then
         T.Duration = ActiveProject.HoursPerDay * 60
         End If
       End If
    Next T

' As-Planned Duration
    APD = ActiveProject.Duration / ActiveProject.HoursPerDay / 60

' Total Project Delay
    If D > APD Then
      TD = D - APD
      MsgBox "As-Built Duration   = " & D & " days" & Chr(10) & "As-Planned Duration    = " & APD
     & " days" & Chr(10) & "Total Project Delay = " & TD & " days"
      Else: TA = APD - D
      MsgBox "As-Built Duration   = " & D & " days" & Chr(10) & "As-Planned Duration    = " & APD
     & " days" & Chr(10) & "Total Project Acceleration = " & TA & " days"
    End If

' Daily Windows Loop
```

```
For i = 1 To D
DE = 0
AC = 0
T0 = 0
T1 = 0
T2 = 0
T3 = 0
T4 = 0
T5 = 0
T6 = 0
T7 = 0
T8 = 0
T9 = 0
T10 = 0
T11 = 0

' Delay or Acceleration on day i ?
For Each T In ActiveProject.Tasks
   If Not (T Is Nothing) Then
      If T.Critical Then
         If (T.Text1 = "O" Or T.Text1 = "C" Or T.Text1 = "N" Or T.Text1 = "O&C") And T.Number1 = i
    And T.Flag3 = False Then
         DE = DE + 1
         End If
         If (T.Text1 = "O" Or T.Text1 = "C" Or T.Text1 = "O&C") And T.Number1 = i And T.Flag3 = True
    Then
         AC = AC + 1
         End If
      End If
   End If
Next T

If AC > 0 And DE = 0 Then GoTo Line1
If AC > 0 And DE > 0 Then GoTo Line3

' Enhanced Daily Windows - Delay
   ' Construct contemporaneous Schedule for Day i
   For Each T In ActiveProject.Tasks
      If Not (T Is Nothing) Then
         If (T.Text1 = "O" Or T.Text1 = "C" Or T.Text1 = "N" Or T.Text1 = "O&C") And T.Number1 = i
    Then
         T.Duration = ActiveProject.HoursPerDay * 60
         End If
         If ((T.Text1 = "" And T.Flag1 = True) Or (T.Text1 = "" And T.Flag2 = True)) And T.Number1 = i
    And T.Flag3 = True Then
         T.Duration = 0
      End If
   End If
Next T
' As-Built Duraion for Window i T0
T0 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
' AS-Planned duration for Window i T1
For Each T In ActiveProject.Tasks
   If Not (T Is Nothing) Then
      If (T.Text1 = "O" Or T.Text1 = "C" Or T.Text1 = "N" Or T.Text1 = "O&C") And T.Number1 = i
 Then
      T.Duration = 0
      End If
    End If
Next T
T1 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
' AS-Built duration without O for Window i T2
For Each T In ActiveProject.Tasks
   If Not (T Is Nothing) Then
      If (T.Text1 = "C" Or T.Text1 = "N" Or T.Text1 = "O&C") And T.Number1 = i Then
      T.Duration = 60 * ActiveProject.HoursPerDay
      End If
    End If
Next T
T2 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
' AS-Built duration without C for Window i T3
For Each T In ActiveProject.Tasks
   If Not (T Is Nothing) Then
      If T.Text1 = "O" And T.Number1 = i Then
      T.Duration = 60 * ActiveProject.HoursPerDay
      End If
      If T.Text1 = "C" And T.Number1 = i Then
      T.Duration = 0
      End If
    End If
Next T
T3 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
' AS-Built duration without N for Window i T4
For Each T In ActiveProject.Tasks
   If Not (T Is Nothing) Then
      If (T.Text1 = "C" Or T.Text1 = "O&C") And T.Number1 = i Then
      T.Duration = 60 * ActiveProject.HoursPerDay
      End If
      If T.Text1 = "N" And T.Number1 = i Then
      T.Duration = 0
      End If
    End If
```

```
Next T
T4 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
' AS-Built duration without O & C for Window i T5
For Each T In ActiveProject.Tasks
   If Not (T Is Nothing) Then
      If T.Text1 = "N" And T.Number1 = i Then
      T.Duration = 60 * ActiveProject.HoursPerDay
      End If
      If (T.Text1 = "O" Or T.Text1 = "C" Or T.Text1 = "O&C") And T.Number1 = i Then
      T.Duration = 0
      End If
   End If
Next T
T5 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
' AS-Built duration without O & N foe Window i T6
For Each T In ActiveProject.Tasks
   If Not (T Is Nothing) Then
      If (T.Text1 = "C" Or T.Text1 = "O&C") And T.Number1 = i Then
      T.Duration = 60 * ActiveProject.HoursPerDay
      End If
      If T.Text1 = "N" And T.Number1 = i Then
      T.Duration = 0
      End If
   End If
Next T
T6 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
' AS-Built duration without C & N for Window i T7
For Each T In ActiveProject.Tasks
   If Not (T Is Nothing) Then
      If T.Text1 = "O" And T.Number1 = i Then
      T.Duration = 60 * ActiveProject.HoursPerDay
      End If
      If T.Text1 = "C" And T.Number1 = i Then
      T.Duration = 0
      End If
   End If
Next T
T7 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
' Reconstruct contemporaneous Schedule for day i
For Each T In ActiveProject.Tasks
   If Not (T Is Nothing) Then
      If (T.Text1 = "O" Or T.Text1 = "C" Or T.Text1 = "N" Or T.Text1 = "O&C") And T.Number1 = i
 Then
      T.Duration = 60 * ActiveProject.HoursPerDay
      End If
      If ((T.Text1 = "" And T.Flag1 = True) Or (T.Text1 = "" And T.Flag2 = True)) And T.Number1 = i
```

```
     And T.Flag3 = True Then
        T.Duration = 0
        End If
       End If
    Next T
    ' Showing results for day i
    MsgBox "Delay responsabilities for window of day " & i & Chr(10) & "OO       = " & T0 - T2 & "
     days" & Chr(10) & "OC        = " & T0 - T3 & " days" & Chr(10) & "ON        = " & T0 - T4 & " days"
     & Chr(10) & "O&C      = " & T2 + T3 - T0 - T5 & " days" & Chr(10) & "O&N      = " & T2 + T4 - T0 -
     T6 & " days" & Chr(10) & "C&N       = " & T3 + T4 - T0 - T7 & " days" & Chr(10) & "O&C&N = " &
     T0 + T5 + T6 + T7 - T1 - T2 - T3 - T4 & " days"
    GoTo Line2


Line1:
  ' Enhanced Daily Windows - Acceleration

    ' Construct contemporaneous Schedule for Day i
    For Each T In ActiveProject.Tasks
      If Not (T Is Nothing) Then
        If ((T.Text1 = "" And T.Flag1 = True) Or (T.Text1 = "" And T.Flag2 = True)) And T.Number1 = i
     Then
        T.Duration = 0
        End If
        If (T.Text1 = "O" Or T.Text1 = "C" Or T.Text1 = "N" Or T.Text1 = "O&C") And T.Number1 = i
     And T.Flag3 = False Then
        T.Duration = ActiveProject.HoursPerDay * 60
        End If
      End If
    Next T
    ' As-Built Duraion T8
       T8 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
    ' AS-Planned duration T9
    For Each T In ActiveProject.Tasks
      If Not (T Is Nothing) Then
        If ((T.Text1 = "" And T.Flag1 = True) Or (T.Text1 = "" And T.Flag2 = True)) And T.Number1 = i
     Then
        T.Duration = 60 * ActiveProject.HoursPerDay
        End If
        If (T.Text1 = "O" Or T.Text1 = "C" Or T.Text1 = "N" Or T.Text1 = "O&C") And (T.Number1 = i
     And T.Flag3 = False) Then
        T.Duration = 0
        End If
      End If
    Next T
      T9 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
    ' AS-Built duration without O T10
    For Each T In ActiveProject.Tasks
      If Not (T Is Nothing) Then
        If (T.Text1 = "" And T.Flag1 = True) And T.Number1 = i Then
        T.Duration = 0
        End If
      End If
    Next T
      T10 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
    ' AS-Built duration without C T11
      For Each T In ActiveProject.Tasks
        If Not (T Is Nothing) Then
          If (T.Text1 = "" And T.Flag1 = True) And T.Number1 = i Then
          T.Duration = 60 * ActiveProject.HoursPerDay
          End If
          If (T.Text1 = "" And T.Flag2 = True) And T.Number1 = i Then
          T.Duration = 0
          End If
         End If
      Next T
      T11 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
    ' Reconstruct contemporaneous Schedule for Day i
    For Each T In ActiveProject.Tasks
      If Not (T Is Nothing) Then
        If ((T.Text1 = "" And T.Flag1 = True) Or (T.Text1 = "" And T.Flag2 = True)) And T.Number1 = i
     And T.Flag3 = True Then
        T.Duration = 0
        End If
        If (T.Text1 = "O" Or T.Text1 = "C" Or T.Text1 = "N" Or T.Text1 = "O&C") And T.Number1 = i
     Then
        T.Duration = ActiveProject.HoursPerDay * 60
        End If
      End If
    Next T
  ' Showing results for day i
  MsgBox "Acceleration responsabilities for window of day " & i & Chr(10) & "OOA       = " & T9 -
     T11 & " days" & Chr(10) & "OCA        = " & T9 - T10 & " days" & Chr(10) & "OA&CA     = " & T10
     + T11 - T8 - T9 & " days"

Line2:
    ' Accumulating Delay Responsibilities
      R1 = R1 + (T0 - T2)
      R2 = R2 + (T0 - T3)
      R3 = R3 + (T0 - T4)
      R4 = R4 + (T2 + T3 - T0 - T5)
      R5 = R5 + (T2 + T4 - T0 - T6)
      R6 = R6 + (T3 + T4 - T0 - T7)
      R7 = R7 + (T0 + T5 + T6 + T7 - T1 - T2 - T3 - T4)
      R8 = R8 + (T9 - T11)
      R9 = R9 + (T9 - T10)
      R10 = R10 + (T10 + T11 - T8 - T9)
      GoTo Line4

Line3:
  ' Simultaneous delay and acceleration events

    ' Delay Subcase

    ' Construct contemporaneous Schedule for Day i (For Delay Subcase)
    For Each T In ActiveProject.Tasks
      If Not (T Is Nothing) Then
        If (T.Text1 = "O" Or T.Text1 = "C" Or T.Text1 = "N" Or T.Text1 = "O&C") And (T.Number1 <= i
     And T.Flag3 = False) Then
        T.Duration = ActiveProject.HoursPerDay * 60
        End If
        If (T.Text1 = "" And T.Flag1 = True And T.Number1 <= i) Or (T.Text1 = "" And T.Flag2 = True
     And T.Number1 <= i) Then
        T.Duration = ActiveProject.HoursPerDay * 60
        End If
      End If
    Next T
    ' As-Built Duraion for Window i T0
    T0 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
    ' AS-Planned duration for Window i T1
    For Each T In ActiveProject.Tasks
      If Not (T Is Nothing) Then
        If (T.Text1 = "O" Or T.Text1 = "C" Or T.Text1 = "N" Or T.Text1 = "O&C") And (T.Number1 = i
     And T.Flag3 = False) Then
        T.Duration = 0
        End If
       End If
    Next T
    T1 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
    ' AS-Built duration without O for Window i T2
    For Each T In ActiveProject.Tasks
      If Not (T Is Nothing) Then
        If (T.Text1 = "C" Or T.Text1 = "N" Or T.Text1 = "O&C") And (T.Number1 = i And T.Flag3 =
     False) Then
        T.Duration = 60 * ActiveProject.HoursPerDay
        End If
       End If
    Next T
```

```
T2 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
' AS-Built duration without C for Window i T3
For Each T In ActiveProject.Tasks
   If Not (T Is Nothing) Then
      If T.Text1 = "O" And (T.Number1 = i And T.Flag3 = False) Then
      T.Duration = 60 * ActiveProject.HoursPerDay
      End If
      If T.Text1 = "C" And (T.Number1 = i And T.Flag3 = False) Then
      T.Duration = 0
      End If
    End If
Next T
T3 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
' AS-Built duration without N for Window i T4
For Each T In ActiveProject.Tasks
   If Not (T Is Nothing) Then
      If (T.Text1 = "C" Or T.Text1 = "O&C") And (T.Number1 = i And T.Flag3 = False) Then
      T.Duration = 60 * ActiveProject.HoursPerDay
      End If
      If T.Text1 = "N" And (T.Number1 = i And T.Flag3 = False) Then
      T.Duration = 0
      End If
    End If
Next T
T4 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
' AS-Built duration without O & C for Window i T5
For Each T In ActiveProject.Tasks
   If Not (T Is Nothing) Then
      If T.Text1 = "N" And (T.Number1 = i And T.Flag3 = False) Then
      T.Duration = 60 * ActiveProject.HoursPerDay
      End If
      If (T.Text1 = "O" Or T.Text1 = "C" Or T.Text1 = "O&C") And (T.Number1 = i And T.Flag3 =
 False) Then
      T.Duration = 0
      End If
   End If
Next T
T5 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
' AS-Built duration without O & N foe Window i T6
For Each T In ActiveProject.Tasks
   If Not (T Is Nothing) Then
      If (T.Text1 = "C" Or T.Text1 = "O&C") And (T.Number1 = i And T.Flag3 = False) Then
      T.Duration = 60 * ActiveProject.HoursPerDay
      End If
      If T.Text1 = "N" And (T.Number1 = i And T.Flag3 = False) Then
      T.Duration = 0
```

```
         End If
      End If
   Next T
   T6 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
   ' AS-Built duration without C & N for Window i T7
   For Each T In ActiveProject.Tasks
      If Not (T Is Nothing) Then
         If T.Text1 = "O" And (T.Number1 = i And T.Flag3 = False) Then
         T.Duration = 60 * ActiveProject.HoursPerDay
         End If
         If T.Text1 = "C" And (T.Number1 = i And T.Flag3 = False) Then
         T.Duration = 0
         End If
      End If
   Next T
   T7 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60


' Acceleration Subcase

   ' Construct contemporaneous Schedule for day i (for acceleration subcase)
   For Each T In ActiveProject.Tasks
      If Not (T Is Nothing) Then
         If (T.Text1 = "O" Or T.Text1 = "C" Or T.Text1 = "N" Or T.Text1 = "O&C") And (T.Number1 = i
    And T.Flag3 = False) Then
         T.Duration = 0
         End If
         If (T.Text1 = "" And T.Flag1 = True) Or (T.Text1 = "" And T.Flag2 = True) And T.Number1 = i
    Then
         T.Duration = 0
         End If
       End If
   Next T
   ' As-Built Duraion T8
       T8 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
   ' AS-Planned duration T9
      For Each T In ActiveProject.Tasks
         If Not (T Is Nothing) Then
            If (T.Text1 = "" And T.Flag1 = True) Or (T.Text1 = "" And T.Flag2 = True) And T.Number1 = i
    Then
            T.Duration = 60 * ActiveProject.HoursPerDay
            End If
         End If
      Next T
      T9 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
   ' AS-Built duration without O T10
```

```
  For Each T In ActiveProject.Tasks
    If Not (T Is Nothing) Then
      If (T.Text1 = "" And T.Flag1 = True) And T.Number1 = i Then
      T.Duration = 0
      End If
    End If
  Next T
  T10 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
' AS-Built duration without C T11
  For Each T In ActiveProject.Tasks
    If Not (T Is Nothing) Then
      If (T.Text1 = "" And T.Flag1 = True) And T.Number1 = i Then
      T.Duration = 60 * ActiveProject.HoursPerDay
      End If
      If (T.Text1 = "" And T.Flag2 = True) And T.Number1 = i Then
      T.Duration = 0
      End If
     End If
  Next T
  T11 = ActiveProject.Duration / ActiveProject.HoursPerDay / 60
' Reconstruct As-Built Schedule
  For Each T In ActiveProject.Tasks
    If Not (T Is Nothing) Then
      If (T.Text1 = "" And T.Flag1 = True) Or (T.Text1 = "" And T.Flag2 = True) And T.Number1 = i
 Then
      T.Duration = 0
      End If
      If (T.Text1 = "O" Or T.Text1 = "C" Or T.Text1 = "N" Or T.Text1 = "O&C") And (T.Number1 =
 i And T.Flag3 = False) Then
      T.Duration = 60 * ActiveProject.HoursPerDay
      End If
    End If
  Next T

' Showing results for day i
MsgBox "Project responsabilities for day " & i & Chr(10) & "This day involve simultaneous delay
 and acceleration events! " & Chr(10) & "Results of the basic subcases before float sharing " &
 Chr(10) & "Subcase of delay " & Chr(10) & "OO       = " & T0 - T2 & " days" & Chr(10) & "OC
 = " & T0 - T3 & " days" & Chr(10) & "ON        = " & T0 - T4 & " days" & Chr(10) & "O&C     = " &
 T2 + T3 - T0 - T5 & " days" & Chr(10) & "O&N     = " & T2 + T4 - T0 - T6 & " days" & Chr(10) &
 "C&N      = " & T3 + T4 - T0 - T7 & " days" & Chr(10) & "O&C&N = " & T0 + T5 + T6 + T7 - T1 - T2
 - T3 - T4 & " days" & Chr(10) & "Subcase of acceleration " & Chr(10) & "OOA       = " & T9 - T11
 & " days" & Chr(10) & "OCA        = " & T9 - T10 & " days" & Chr(10) & "OA&CA     = " & T10 +
 T11 - T8 - T9 & " days"

' Float sharing on day i
```

```
If T0 > T2 And T9 > T10 Then
MsgBox "Final responsabilities for day " & i & "after float sharing" & Chr(10) & "Delay
 responsibilities " & Chr(10) & "OO       = " & ((T0 - T2) + (0.5 * (T9 - T10))) & " days" & Chr(10) &
 "OC        = " & T0 - T3 & " days" & Chr(10) & "ON        = " & T0 - T4 & " days" & Chr(10) & "O&C
 = " & T2 + T3 - T0 - T5 & " days" & Chr(10) & "O&N      = " & T2 + T4 - T0 - T6 & " days" & Chr(10)
 & "C&N      = " & T3 + T4 - T0 - T7 & " days" & Chr(10) & "O&C&N = " & T0 + T5 + T6 + T7 - T1 -
 T2 - T3 - T4 & " days" & Chr(10) & "Acceleration responsibilities " & Chr(10) & "OOA       = " &
 T9 - T11 & " days" & Chr(10) & "OCA        = " & 0.5 * (T9 - T10) & " days" & Chr(10) & "OA&CA
 = " & T10 + T11 - T8 - T9 & " days"

   ' Accumulating Delay Responsibilities
     R1 = R1 + ((T0 - T2) + (0.5 * (T9 - T10)))
     R2 = R2 + (T0 - T3)
     R3 = R3 + (T0 - T4)
     R4 = R4 + (T2 + T3 - T0 - T5)
     R5 = R5 + (T2 + T4 - T0 - T6)
     R6 = R6 + (T3 + T4 - T0 - T7)
     R7 = R7 + (T0 + T5 + T6 + T7 - T1 - T2 - T3 - T4)
     R8 = R8 + (T9 - T11)
     R9 = R9 + (0.5 * (T9 - T10))
     R10 = R10 + (T10 + T11 - T8 - T9)
     GoTo Line4

Else
  If T0 > T3 And T9 > T11 Then
  MsgBox "Final responsabilities for day " & i & " after float sharing" & Chr(10) & "Delay
 responsibilities " & Chr(10) & "OO       = " & (T0 - T2) & " days" & Chr(10) & "OC        = " & ((T0 -
 T3) + (0.5 * (T9 - T11))) & " days" & Chr(10) & "ON        = " & T0 - T4 & " days" & Chr(10) &
 "O&C      = " & T2 + T3 - T0 - T5 & " days" & Chr(10) & "O&N      = " & T2 + T4 - T0 - T6 & " days"
 & Chr(10) & "C&N      = " & T3 + T4 - T0 - T7 & " days" & Chr(10) & "O&C&N = " & T0 + T5 + T6 +
 T7 - T1 - T2 - T3 - T4 & " days" & Chr(10) & "Acceleration responsibilities " & Chr(10) & "OOA
 = " & (0.5 * (T9 - T11)) & " days" & Chr(10) & "OCA        = " & 0.5 * (T9 - T10) & " days" &
 Chr(10) & "OA&CA      = " & T10 + T11 - T8 - T9 & " days"

   ' Accumulating Delay Responsibilities
     R1 = R1 + (T0 - T2)
     R2 = R2 + ((T0 - T3) + (0.5 * (T9 - T11)))
     R3 = R3 + (T0 - T4)
     R4 = R4 + (T2 + T3 - T0 - T5)
     R5 = R5 + (T2 + T4 - T0 - T6)
     R6 = R6 + (T3 + T4 - T0 - T7)
     R7 = R7 + (T0 + T5 + T6 + T7 - T1 - T2 - T3 - T4)
     R8 = R8 + (0.5 * (T9 - T11))
     R9 = R9 + (T9 - T10)
     R10 = R10 + (T10 + T11 - T8 - T9)
     GoTo Line4
```

```
      Else
      MsgBox "Final responsabilities for day " & i & "after float sharing" & Chr(10) & "Delay
     responsibilities " & Chr(10) & "OO       = " & (T0 - T2) & " days" & Chr(10) & "OC        = " & (T0 -
     T3) & " days" & Chr(10) & "ON        = " & T0 - T4 & " days" & Chr(10) & "O&C     = " & T2 + T3 -
     T0 - T5 & " days" & Chr(10) & "O&N     = " & T2 + T4 - T0 - T6 & " days" & Chr(10) & "C&N      =
     " & T3 + T4 - T0 - T7 & " days" & Chr(10) & "O&C&N = " & T0 + T5 + T6 + T7 - T1 - T2 - T3 - T4 &
     " days" & Chr(10) & "Acceleration responsibilities " & Chr(10) & "OOA       = " & (T9 - T11) & "
     days" & Chr(10) & "OCA        = " & 0.5 * (T9 - T10) & " days" & Chr(10) & "OA&CA     = " & T10 +
     T11 - T8 - T9 & " days"
      End If
    End If

     ' Accumulating Delay Responsibilities
      R1 = R1 + (T0 - T2)
      R2 = R2 + (T0 - T3)
      R3 = R3 + (T0 - T4)
      R4 = R4 + (T2 + T3 - T0 - T5)
      R5 = R5 + (T2 + T4 - T0 - T6)
      R6 = R6 + (T3 + T4 - T0 - T7)
      R7 = R7 + (T0 + T5 + T6 + T7 - T1 - T2 - T3 - T4)
      R8 = R8 + (T9 - T11)
      R9 = R9 + (T9 - T10)
      R10 = R10 + (T10 + T11 - T8 - T9)

Line4:
    Next i
    MsgBox "Enhanced Daily Windows Analysis EDWA" & Chr(10) & "Project delay responsibilities"
     & Chr(10) & "OO       = " & R1 & " days" & Chr(10) & "OC        = " & R2 & " days" & Chr(10) &
     "ON        = " & R3 & " days" & Chr(10) & "O&C     = " & R4 & " days" & Chr(10) & "O&N     = " &
     R5 & " days" & Chr(10) & "C&N      = " & R6 & " days" & Chr(10) & "O&C&N = " & R7 & " days"
     & Chr(10) & "Project acceleration responsibilities" & Chr(10) & "OOA      = " & R8 & " days" &
     Chr(10) & "OCA       = " & R9 & " days" & Chr(10) & "OA&CA     = " & R10
End Sub
```