# Criticality of Schedule Constraints – Classification and Identification

**Qui T. Nguyen[1] and David K. H. Chua[2]**

## Abstract

In construction scheduling, constraints among activities are vital as they govern the schedule solution. Understanding their criticality is essential for better schedule management. This paper presents a systematic method to identify and classify the criticality of schedule constraints for the schedule management from the constraint perspective. In terms of criticality, schedule constraints can be grouped into four types: project-critical, activity-critical, sequence-critical and non-critical. Project-critical constraints are those which govern start/finish time of critical activities and the project end time. Activity-critical constraints define the start/finish time of non-critical activities, and sequence-critical constraints are those whose existence affect the start/finish time of some activities or the project end time. Constraints belonging to any of these groups are vital to a schedule as they cannot be removed from the constraint collection. Non-critical constraints, on the other hand, do not govern either start/finish time of any activity or the project end time. Accordingly, non-critical constraints are redundant and can be removed from the constraint collection without causing any change to the schedule solution. The method proposed was applied to a case example for further interpretation. The proposed classification scheme could shed light on a more in-depth understanding of the nature of criticality and the role of constraints in a schedule, and thus better schedule management may be achieved.

**Keywords:** constraint criticality; constraint flexibility; constraint management; schedule management

## Introduction

In a construction project, schedule constraints represent the construction requirements that a schedule must satisfy. They define the precedence relationships among activities as well as the sequences that construction processes may follow (Chua and Yeoh 2011). Each constraint exhibits different influence to the schedule according to its characteristic and/or the activities involved. A constraint could be of no significance, locally significant to an activity or globally crucial for the entire schedule. In some cases, a constraint, when exists, could be of no significance to the schedule, yet removing it from the constraint collection could lead to changes in activity's times and/or sequence.

Constraint management is an essential task of schedule management. The major aim is to identify and prioritize the "key" or critical constraints that govern the overall schedule (Chua and Shen 2005). Generally, a critical constraint could be any that control the project duration, the start and/or finish times of an activity or the sequence currently defined in the schedule. In other words, a critical constraint cannot be removed from the constraint collection, since such a removal will lead to changes in the schedule. Accordingly, the definition of critical constraints is likely broader than that of critical activities. While critical activities are those explicitly shown in the critical path(s), a critical constraint could be

---

[1] PhD Candidate, Department of Civil Engineering, National University of Singapore,
10, Kent Ridge Crescent, 119260, Singapore, Tel: +65-6-5164643, Fax: +65-6-7791635, E-mail: ntqui@nus.edu.sg.

[2] Associate Professor, Department of Civil Engineering, National University of Singapore,
10, Kent Ridge Crescent, 119260, Singapore, Tel: +65-6-5162195, Fax: +65-6-7791635, E-mail: ceedavid@nus.edu.sg.

between two non-critical activities. It also means that there could be different types of criticality dependent on how a constraint affects a certain activity or the overall schedule. Thus, understanding and classifying the nature of constraint criticality is necessary for constraint management as well as schedule controlling.

The concept of criticality already has been introduced since the formation of the Critical Path Method (CPM) in the 1950's. CPM allows planners to identify critical paths and critical activities, from which critical constraints can be implicitly inferred as those connecting two critical activities. However, as CPM has limitations in representing non-precedence constraints such as work/resource continuity or process concurrency/ overlap/ disjunction (El-Bibany 1997), inferring critical constraints from critical paths could generally be inadequate.

Previous researchers have put much effort to develop the criticality concept in resource-constrained scheduling problems (Bowers 1995; Lu and Li 2003; Rivera and Duran 2004; Wiest 1981; Woodworth and Shanahan 1988). Their major focus is to identify resource-constrained critical paths. Critical constraints, either precedence or resource constraints, could also be determined from critical paths as they are the constraints connecting critical activities. The idea of critical constraints is therefore still restricted to those that affect the final project end time, and constraints among non-critical activities could be intuitively considered non-critical.

Schedule constraints in a construction project can be of any types, not only precedence or resource constraints. They impose the conditions that an activity can start, process and finish. In this context, a critical constraint may not necessarily be between two critical activities. Chua and Shen (2005) proposed a methodology to identify key information and resource constraints in a delayed project. In their model, information and resource constraints are modeled as unary temporal constraints of activities, and the impact of constraints to the overall project performance is measured using constraint float. This method helps planner highlight the hidden bottleneck constraints so that appropriate policies can be utilized to lessen the delay.

According to the above review, although many methodologies have been proposed to develop the concept of criticality in construction schedules, it is found that there is still room for improvement. Firstly, critical constraints could not be restricted to those between critical activities. From the construction viewpoint, not only the project duration but also the start/ finish times of all activities are of importance to contractors, as they may affect their overall working plan among different projects. While each activity may be involved in different constraints, it is regular that only some of them actually control the activity's times. Thus, these constraints should also have higher priority for better management. Secondly, schedule constraints generally can be of any types, such as unary or binary, and with minimal-lag, maximal-lag or non-lag requirements. Thus, a generic and systematic approach which can be applied to all constraint types is necessary. Lastly, critical constraints should be identified as early as in the planning phase, so that better management strategies can be applied for better project performance.

This paper aims to investigate the criticality in construction schedules from a constraint viewpoint. It presents a systematic methodology for classifying and identifying critical schedule constraints. Both unary and binary with minimal-lag, maximal-lag and non-lag requirements are examined. In the context of this paper, schedule constraints are defined as temporal interval constraints and captured using the PDM++ model (Chua and Yeoh 2011), which is briefly summarized in the next section. Then, a detailed description of the proposed methodology is presented, followed by the demonstration of its application via an illustrative example schedule. Subsequently, a short comparison between constraint criticality and activity criticality is presented to highlight the differences and the advantages of identifying constraint criticality. By categorizing and determining constraint criticality in a systematic way, the proposed methodology could help provide a deeper understanding about their role to the overall schedule, so that better schedule performance could be achieved.

## Modeling Schedule Constraints

Schedule constraints represent construction requirements that a schedule must satisfy. They can exist in different forms such as functional requirements, resource or safety constraints (Nguyen et al. 2009). For scheduling purpose, all constraints need to be eventually converted into temporal unary/binary relationships between activities. Temporal constraints can be represented in either a point-to-point format as in CPM/PDM models, or an interval-to-interval format (Allen 1984). Despite its simplicity and capability of modeling lag-time requirements, the point-to-point format has been found to be inadequate in representing complex constraints such as work/resource continuity or disjunction, and process concurrency/overlapping. In contrast, the Allen's representation format could provide greater flexibility and a richer semantic context to explicitly describe the precedence, coincidence, concurrency, and disjunction constraints between two time intervals. However, this format lacks a mechanism to capture lag time requirements.

| **Unary Constraint** | | **Non-lag Type Binary Constraint** | |
|---|---|---|---|
| Description | Mathematical Definition | Non-lag | Mathematical Definition |
| $X$ Due-Before($m$) | $X^- + d_X \leq m$ | $X$ Meets $Y$ | $X^- + d_X = Y^-$ |
| $X$ Due-After($m$) | $X^- + d_X \geq m$ | $Y$ Met-by $X$ | |
| $X$ Start-Before($m$) | $X^- \leq m$ | $X$ Contains $Y$ | ($X^- \leq Y^-$) AND<br>($X^- + d_X \leq Y^- + d_Y$) |
| $X$ Start-After($m$) | $X^- \geq m$ | $Y$ Contained-by $X$ | |
| | | $X$ Disjoint $Y$ | ($X^- + d_X \leq Y^-$) OR<br>($Y^- + d_Y \leq X^-$) |

| **Minimal and Maximal-Lag Type Binary Constraint** | | | |
|---|---|---|---|
| Minimal-lag | Mathematical Definition | Maximal-lag | Mathematical Definition |
| $X$ Before($m$) $Y$ | $X^- + d_X + m \leq Y^-$ | $X$ Before(~$m$) $Y$ | $X^- + d_X + m \geq Y^-$ |
| $Y$ After($m$) $X$ | | $Y$ After(~$m$) $X$ | |
| $X$ Starts($m$) $Y$ | $X^- + m \leq Y^-$ | $X$ Starts(~$m$) $Y$ | $X^- + m \geq Y^-$ |
| $Y$ Started-by($m$) $X$ | | $Y$ Started-by(~$m$) $X$ | |
| $X$ Finishes($m$) $Y$ | $X^- + d_X + m \leq Y^- + d_Y$ | $X$ Finishes(~$m$) $Y$ | $X^- + d_X + m \geq Y^- + d_Y$ |
| $Y$ Finished-by($m$) $X$ | | $Y$ Finished-by(~$m$) $X$ | |
| $X$ Overlaps($m$) $Y$ | ($X^- + d_X \geq Y^- + m$) AND | $X$ Overlaps(~$m$) $Y$ | ($X^- + d_X \leq Y^- + m$) OR |
| $Y$ Overlaped-by($m$) $X$ | ($Y^- + d_Y \geq X^- + m$) | $Y$ Overlaped-by(~$m$) $X$ | ($Y^- + d_Y \leq X^- + m$) |
| $X$ Start-Finish($m$)$Y$ | $X^- + m \leq Y^- + d_Y$ | $X$ Start-Finish(~$m$)$Y$ | $X^- + m \geq Y^- + d_Y$ |

Figure 1. PDM++ temporal interval constraints

The PDM++ model (Chua and Yeoh 2011) integrates the advantages of the two modeling paradigms. It extends the traditional PDM model by incorporating two basic logical operators "AND" ($\wedge$) and "OR" ($\vee$) with the enriched syntax inspired by the Artificial Intelligence developed by Allen. Hence, it could capture a wider range of schedule constraints. In this paper, schedule constraints are defined as interval temporal relationships between activities using the PDM++ model. Accordingly, both unary and binary constraints with different lag-time requirements are analyzed. A summarized description of these constraints is presented in Figure 1, where $X^-$ and $d_X$ respectively denote the start time and duration of activity X, and $m$ ($m \geq 0$) denotes the lag-time requirement.

## Criticality of Schedule Constraints

### Definition

In this paper, a schedule constraint is considered critical if its existence affects the project duration, activity's start/finish times or the sequence between activities. In other words, any change or removal of a critical constraint will lead to variations to the schedule. In contrast, non-critical constraints are redundant ones, and deleting such constraints results in no change to either schedule times or sequence.

### Classification

Although critical, schedule constraints could have different impacts to the schedule, due to their nature and the activities involved. Thus, a classification of constraint criticality would be useful to further distinguish the significance of constraints to the schedule. This paper categorizes constraint criticality into four groups, termed as: project-critical, activity-critical, sequence-critical, and non-critical.

### *Project-critical Constraint*

A constraint is project-critical when it governs the project duration. Since project duration is defined by start/finish times of critical activities, a project-critical constraint is the one that directly controls the start/finish times of a critical activity. By this definition, it is apparent that there is a correlation between a critical activity path and a project-critical constraint path. More precisely, any critical activity path has an associated project-critical constraint path, which passes through the constraints governing the start/finish times of the critical activities involved. As a result, project-critical constraints can be implicitly derived from critical activity paths.

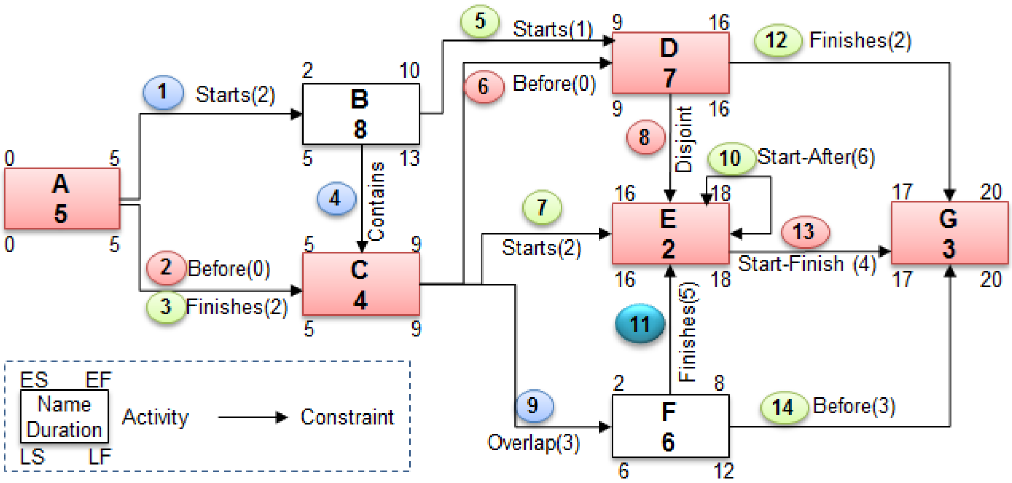


Figure 2. Example schedule network

Figure 2 depicts a simple schedule network with 7 activities and 14 constraints, with a total duration of 20 days. The modeling syntax of constraints follows the legend shown in Figure 1. The critical activity path is A-C-D-E-G. This path also includes four project-critical constraints named (2), (6), (8), and (13). These constraints directly define the start/finish times of activities C, D, E, and G respectively. For instance, although there are two constraints between A and C (named (2) and (3)), the start time of C is defined by Constraint (2), which is mathematically expressed as: $A^{-} + d_A + 0 \leq C^{-}$. When Constraint (2) is modified to *Before(1)*, start and finish times of activity C also change to new values as 6 and 10 respectively, resulting in a new project duration of 21 days.

### *Activity-critical Constraint*

Similar to critical activities, the start/finish times of every non-critical activity are also controlled by at least one constraint. These times can be changed due to any change or deletion of such a constraint. Although it may not be critical to project duration, constraints of this type are also vital to the schedule. In addition, an activity-critical constraint becomes project-critical if the activities involved are critical. Alternatively, it is possible to state that project-critical constraint is a subclass of activity-critical constraint which defines a relationship between two critical activities.

In the example schedule shown in Figure 2, B is a non-critical activity and its start/finish times are controlled by two constraints (1) and (4). In detail, constraint (1) defines its early start/finish times while constraint (4) governs its late start/finish times. In the case that the constraint has some change, the controlled times are also affected, while the project duration is not be influenced. For example, if Constraint (1) is changed to *Starts(1)*, early start/finish times of B will change to 1 and 9 respectively. However, the project duration is kept unchanged as 20 days.

### *Sequence-critical Constraint*

When a constraint does not control start/finish times if any activity, it is commonly considered “non-critical”. Consequently, it is easily to be intuitively treated as a redundant constraint, which means that any change or removal of such a constraint is considered not to cause any change to the schedule. However, it possibly happens that when a non-critical constraint is deleted, the sequence among activities can also be changed to achieve a shorter project duration.

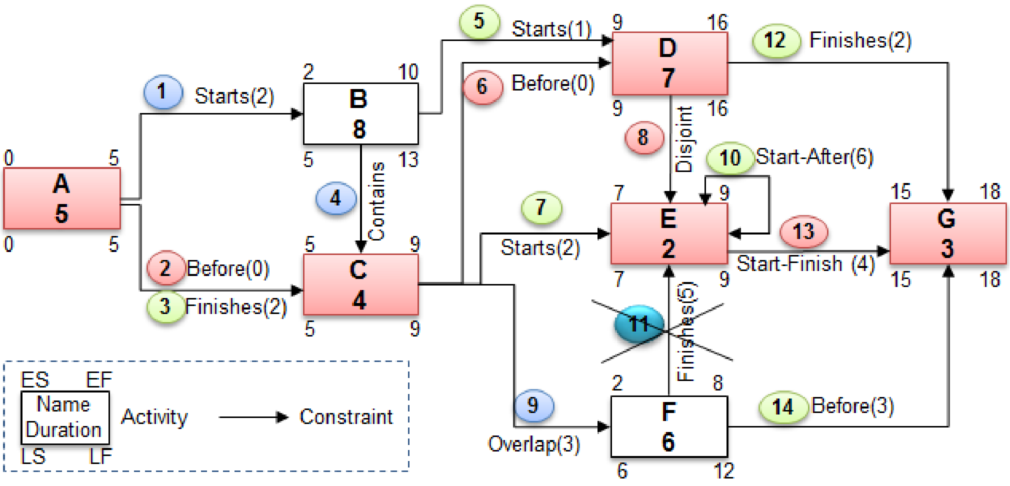
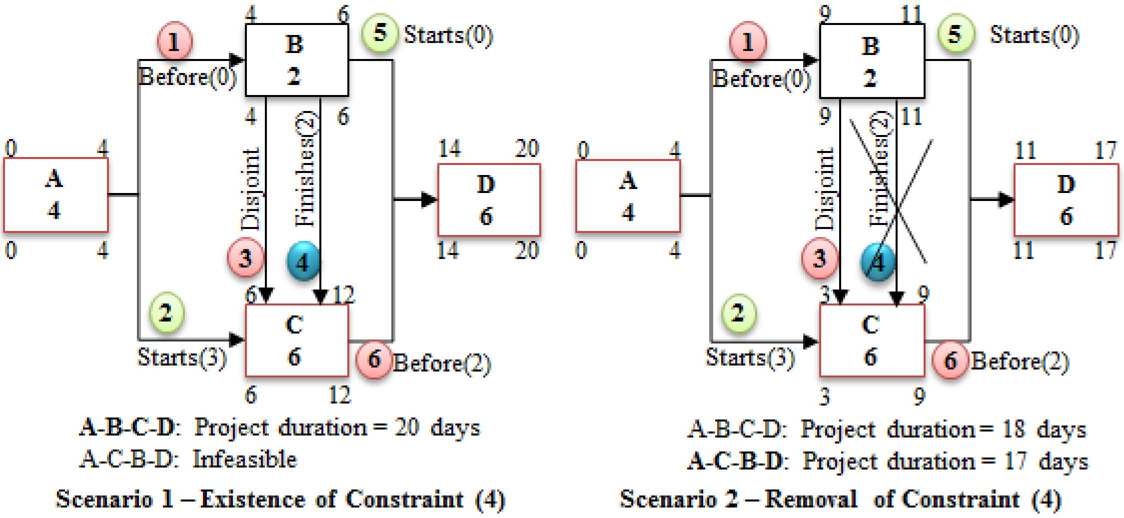


Figure 3. Example schedule network - Removal of Constraint (11)

Constraint (11) in Figure 2 is an example of this situation. In the current schedule, activity D is scheduled before activity E as this sequence provides better project duration. (In the other sequence where D starts after E due to the disjoint constraint, the project duration is 22 days). Constraint (11) can be considered non-critical as it does not control any activity’s times. Yet, if it is omitted, the preferable sequence will switch to A-C-E-D-G, with shorter project duration of 18 days as shown in Figure 3. (If D starts before E, project duration can be easily calculated as 20 days).



Scenario 1 – Existence of Constraint (4)

Scenario 2 – Removal of Constraint (4)

Figure 4. Example schedule network of sequence-critical constraint

It is apparently that there may be some variation in activity sequence and/or project duration when a "non-critical" constraint is deleted. It could also happen that removing a "non-critical" constraint may allow infeasible sequence become feasible as illustrated in Figure 4. In other words, such a constraint, when exists, does contribute some impact to the overall schedule solution and cannot be consider "non-critical". This class of constraints is defined as "sequence-critical". It refers to those constraints whose existence affects the feasibility of possible alternative sequences or the most preferable sequence (providing shortest project duration). Due to this distinctive characteristic, from the management viewpoint, sequence-critical constraints thus could not be treated as redundant constraints.

### *Non-critical Constraint*

The last category of constraint criticality is "non-critical". This class refers to those constraints that do not control the start/finish times of any activity and project duration. Additionally, any removal of non-critical constraints will cause no change to the schedule. They could thus be considered redundant constraints.

## Identification Methodology

Identifying the criticality type of a constraint seems to be simple in small schedules with a small number of activities and constraints. However, as construction projects commonly involve tens or hundreds of activities and constraints, and manually checking each constraint for its criticality is clearly time-consuming and probably impossible. Therefore, a systematic methodology for identifying constraint criticality is obviously a necessity. For management purpose, a criticality indicator is also essential for constraint comparison and evaluation.

The criticality of a constraint closely relates to how it affects the schedule when its characteristics and/or its existence status change. From the management perspective, examining a constraint is required only when changes would adversely affect the schedule or when it is removed from the constraint collection. Prolonging project duration, delaying or reducing the feasible range of an activity's start/finish times can be referred to as adverse impacts of a constraint's variation. According to the general nature of constraints and the overall objective of scheduling which aims to minimize the project duration, adverse schedule changes could be caused when a constraint become "tighter" or "stricter". This situation happens when increasing or decreasing lag-time requirement of a minimal or maximal-lag type constraint. Constraint criticality is also reflected by how much it can be tightened. Thus, the present methodology utilizes the tightening degree, or tightening time, as an indicator of constraint criticality.

The tightening time of a constraint can be obtained by examining how much it can adversely vary without causing changes to the activities involved. Alternatively, it can be determined by identifying how much an activity can be flexibly moved backward or forward without violating that particular constraint. The flexibility of an activity *X* of a constraint *C*, denoted as $f_{XC}$, is the minimal value between moving backward and forward durations (denoted as $f_{XC}^{BW}$ and $f_{XC}^{FW}$ respectively), expressed as:

$$f_{XC} = Min(f_{XC}^{BW}, f_{XC}^{FW}) \tag{1}$$

A constraint cannot be further tightened when its involved activities cannot vary or have no flexibility. Thus, the tightening time of a constraint *C* between two activities *X* and *Y* (denoted as $T_C$) is the minimal flexibility of *X* and *Y*, shown as:

$$T_C = Min(f_{XC}, f_{YC}) \tag{2}$$

Tightening time represents how much a constraint can varies without affecting an activity's times and/or project time. Thus, $T_C = 0$ indicates that constraint *C* cannot be tightened anymore, and thus it is either project-critical (when linking two critical activities) or activity-critical otherwise. When $T_C > 0$, constraint *C* still have room for tightening and thus it is either sequence-critical or non-critical. In this case, the schedule needs to be further analyzed by re-computing it without the existence of *C*. If there no change in activity's times or sequence, *C* is non-critical; otherwise, it is a sequence-critical constraint. A summary of the proposed criticality identifier is presented in Table 1.

Table 1. Indicator of constraint criticality

| Criticality Type | Impact of Variation or Removal | Indicator |
|---|---|---|
| Project-critical | Change of critical activities' times and/or project duration | - Between critical activities<br>- $T_C = 0$ |
| Activity-critical | Change of non-critical activities' times | - Not project critical<br>- $T_C = 0$ |
| Sequence-critical | Change of activities' times, project times and/or sequence when removed | - $T_C = 0$<br>- There exist at least one change of activity times or sequence when removed |
| Non-critical | No impact | - $T_C = 0$<br>- No change of activity times or sequence when removed |

## Constraint Evaluation

Criticality can be used as a criterion for evaluating schedule constraints. Directly affecting project duration, project-critical constraints are apparently the most crucial since unfulfilling these constraints may cause delay of the whole project. Secondly, activity-critical constraints are also important for planners as they define the activities' start/finish times. Such constraints also need to be well-managed in order to maintain activity's times as planned. Sequence-critical and non-critical constraints exhibit less significance to the schedule compared to the two aforementioned types. Although not crucial, they yield different impacts when removed from the constraint collection. While removing a non-critical constraint causes no change to activities' times or sequence, variation will happen when a sequence-critical constraint is removed. As such, sequence-critical constraints are more significant than non-critical constraints and should receive more attention.

### Illustrative Example

The schedule example shown in Figure 2 is used to illustrate the implementation of the proposed method. Despite its simplicity compared with real-scaled projects, according to the preliminarily analysis in the previous section, this small schedule involves all criticality types. Thus using this example could demonstrate the capability of the present approach.

The general procedure of identifying the criticality of a constraint *C* consists of five steps as follows:

1. Calculate the flexibility of activities involved, $f_{XC}$ and $f_{YC}$, using their start/finish times following Equation (1)
2. Compute the tightening time $T_C$ using Equation (2)
3. Classify *C* as project-critical, activity-critical or non-critical based on the criticality of the activities involved
4. If *C* is non-critical, remove *C* from the constraint collection and reschedule the project
5. If there exist any change in activity's times or sequence, then *C* is sequence-critical; otherwise, *C* is non-critical

This procedure was sequentially applied to each constraint of the example schedule. The result obtained is presented in Table 2.

Table 2. Result of constraint criticality

| **Constraint** | **Description** | **Tightening Time** | **Criticality Type** |
|---|---|---|---|
| 1 | *A* Starts(2) *B* | 0 | Activity-critical |
| 2 | *A* Before(0) *C* | 0 | Project-critical |
| 3 | *A* Finishes(2) *C* | 2 | Non-critical |
| 4 | *B* Contains *C* | 0 | Activity-critical |
| 5 | *B* Starts(1) *D* | 6 | Non-critical |
| 6 | *C* Before(0) *D* | 0 | Project-critical |
| 7 | *C* Starts(2) *E* | 9 | Non-critical |
| 8 | *C* Disjoint *D* | 0 | Project-critical |
| 9 | *C* Overlaps(3) *G* | 0 | Activity-critical |
| 10 | *E* Start-After(6) | 10 | Non-critical |
| 11 | *F* Finishes(5) *E* | 5 | Sequence-critical |
| 12 | *D* Finishes(2) *G* | 2 | Non-critical |
| 13 | *E* Start-Finish(4) *G* | 0 | Project-critical |
| 14 | *F* Before(3) *G* | 12 | Non-critical |

The result obtained showed that constraints (2), (6), (8), and (13) are project-critical. They respectively govern the start/finish times of critical activities C, D, E, and G. Constraints (1), (4), and (9) are activity-critical and control the start/finish times of activities B and F. Changing these constraints may not affect project duration, but it will cause variation to start/finishes time of B and F. Constraint (11) is sequence-critical while the rest are non-critical with positive tightening time. Among non-critical constraints, constraint (14) can be considered the least significant as it have the largest tightening time of 12 days. It means that this constraint can be more tightened up to 12 days without causing any change to the schedule.

### Constraint Criticality vs. Activity Criticality

The concept of criticality plays a vital role for schedule management. This concept is traditionally applied from the activity perspective. The major focus is to determine the most crucial or critical activities that have significant impact to the overall schedule. From that, planners could produce a suitable management strategy to reduce the adverse impact of activity changes. Activity criticality is helpful to manage uncertainties at the activity level such as uncertain durations or disruptions. However, this concept could not provide planners with information about which constraints (and which construction requirements in a broader view) could affect an activity, or how an activity could be impact if a certain constraint has variations.

Constraint criticality concept, on the other hand, concentrates on the role of a constraint to activities' times and project duration. It could allow planners to identify which constraints are the key bottlenecks that could have adverse impact to the schedule or which constraints directly govern the sequence among activities. As schedule constraints are generated from construction requirements which commonly vary along the project lifetime, besides critical activities, determining essentially crucial constraints are also necessary for planners to place high priority at the right places, and thus the overall schedule performance could be improved.

## Conclusion

Schedule constraints play a vital role for schedule planning and controlling. They define the temporal relationships and sequences among activities. However, constraints could yield different impact on activities' times and project duration. As constraint criticality differ from one constraint to another, identifying constraint criticality is vital for schedule management.

This paper introduces a classification schema of constraint criticality. In a schedule, a constraint could be project-critical, activity-critical, sequence-critical or non-critical depending on how it could affect activities' start/finishes times and/or project duration. A systematic methodology for identifying constraint criticality is also presented. The proposed method uses the tightening time of constraints and the characteristic of the activities involved as an indicator of criticality. Its application was demonstrated via a simple schedule example. The result obtained showed that different criticality types can be systematically determined.

A key advantage of the present methodology is that it could provide a deeper understanding about the criticality of schedule constraints and how they could affect the schedule performance when changes happen. As a result, planners could choose the most appropriate management policy for each constraint to achieve better project performance. However, although correlation among schedule constraints possibly exists, the current research did not examine this situation. Future research should attempt to investigate the impact of constraint dependencies to constraint criticality. Such a research would provide more insight knowledge on the nature of constraint criticality.

## References

Allen, J. F. (1984). "Towards a general theory of action and time." *Artificial Intelligence*, 23(2), 123-154.

Bowers, J. A. (1995). "Criticality in resource constrained networks." *The Journal of Operational Research Society*, 46(1), 80.

Chua, D. K. H., and Shen, L. J. (2005). "Key constraints analysis with integrated production scheduler." *Journal of Construction Engineering and Management*, 131(7), 753-764.

Chua, D. K. H., and Yeoh, K. W. (2011). "PDM++: Planning framework from a construction requirements perspective." *Journal of Construction Engineering and Management*, 137(4), 266-274.

El-Bibany, H. (1997). "Parametric Constraint Management in Planning and Scheduling: Computational Basis." *Journal of Construction Engineering and Management*, 123(3), 348-353.

Lu, M., and Li, H. (2003). "Resource-activity critical-path method for construction planning." *Journal of Construction Engineering and Management*, 129(4), 412-420.

Nguyen, T. Q., Chua, D. K. H., and Yeoh, K. W. (Year). "Functional requirement oriented modeling framework for schedule generation." *The 6th International Conference on Innovative in Architecture, Engineering & Construction (AEC)*, State College, Pennsylvania.

Rivera, F. A., and Duran, A. (2004). "Critical clouds and critical sets in resource-constrained projects." *International Journal of Project Management*, 22(6), 489-497.

Wiest, J. D. (1981). "Precedence diagramming method: some unsual characteristics and their implications for project managers." *journal of operations management*, 1(3), 121.

Woodworth, B. M., and Shanahan, S. (1988). "Identifying the critical sequence in a resource constrained project." *International Journal of Project Management*, 6(2), 89-96.