# CONTROLLING SEQUENCE AND CONTROLLING ACTIVITIES IN THE REPETITIVE SCHEDULING METHOD

By Photios G. Ioannou[1] and I-Tung Yang[2]

## ABSTRACT

The duration of target schedules produced by the Repetitive Scheduling Method (RSM) for both discrete and continuous repetitive projects is determined by a new concept, the controlling sequence, a continuous chain of controlling activities that span the project from start to finish. RSM scheduling and the identification of the controlling sequence are performed through the new concepts of controlling points and controlled points. The Trace Algorithm is introduced to (i) identify all controlling sequences and (ii) assign algebraic signs to controlling activities to connect them to (a) push-pull lean concepts, (b) project duration, (c) the RSM paradox, and (d) three new RSM scheduling principles. Proposed new bounds, Lo and Hi, define (a) production line segments belonging to multiple controlling sequences and (b) a new algorithm for identifying all controlling activities directly without tracing possibly billions of controlling sequences. The application of RP2, a program developed by the authors to identify controlling activities and critical activities in graphical and tabular reports, is illustrated by scheduling a complex of apartment buildings.

## KEYWORDS:

planning, scheduling, repetitive project, repeating activities, continuous resource utilization, resource work continuity, unforced idleness, LOB, RSM, RP2, linear construction, vertical construction, horizontal construction, discrete projects, continuous projects, target schedule, controlling sequence, controlling activities, critical path, critical activities, CPM, control links, control points, algorithm, bounds, push-pull, RSM paradox.

## CORRESPONDING AUTHOR:

Photios G. Ioannou, Professor, Civil and Environmental Engineering Dept,
University of Michigan, 2350 G.G. Brown Bldg, Ann Arbor, MI 48109-2125,
Tel:734-764-3369, Fax: 734-764-4292,
email: photios@umich.edu

---

[1] Prof., Dept. of Civ. and Envir. Engrg., Univ. of Michigan, Ann Arbor, MI 48109-2125. email: photios@umich.edu

[2] Prof., Dept. of Civil and Constr. Engrg., Nat. Taiwan Univ. of Science and Tech., Taipei, Taiwan. E-mail: ityang@mail.ntust.edu.tw

## Introduction

Construction projects with repeating units, such as multi-story buildings, housing developments, highways, pipelines, etc., present unique scheduling challenges, the most important of which is to ensure that crews performing repetitive tasks can work continuously from one unit or location to the next (Ballard and Tommelein 1999, Riley and Savindo 1997). Several line-of-balance (LOB) variants have been developed over the years as alternatives to the critical path method (CPM) and are described in Harris and Ioannou (1998b). The Repetitive Scheduling Method (RSM) was proposed in (Harris and Ioannou 1998a, 1998b) as a general methodology that unifies all LOB variants into one repetitive scheduling framework that applies to all types of repetitive projects, including those with discrete repetitive units (e.g., high-rise buildings) as well as those repeating over continuous space (e.g., highways).

Two sets of activities are of primary importance in an RSM target schedule:

- Prior to construction and while still planning and scheduling a repetitive project, attention should be focused on the *controlling sequence*, the chain of *controlling activities* that determine the duration of the target schedule. By fine-tuning these activities, planners can adjust project duration to meet contractual obligations and managerial objectives.
- Once a target schedule has been developed and construction begins, attention should shift to *critical* activities, i.e., those activities in the target schedule that, if delayed *during construction*, will delay the actual project past the target completion time.

This paper presents the results of significant research and development that continues to extend and enhance RSM (Ioannou and Yang 2015). In particular, it concentrates on the definition of the *controlling sequence*, presents new bounds and algorithms for identifying the activities on the controlling sequence, and introduces new principles for adjusting the duration of the target schedule. These fundamental extensions to RSM are explained and illustrated through discrete and continuous example projects. An example application of RSM to a real building project— scheduled using RP2, an RSM-based computerized scheduling system developed by the authors—is used to illustrate the proposed new methodology in practice.

## RSM Key Concepts

RSM is a scheduling technique for projects where the same activities repeat over similar units of work. The objective of RSM is to minimize project duration while ensuring that crews (resources) can work continuously from one repeating unit or location to the next.

Important facts about RSM (as currently extended) can be summarized as follows:

- RSM applies to all types of projects with repeating work, including those with discrete work units (such as multistory buildings, housing developments, machine installations, etc.) and projects with continuous work (such as highways, pipelines, canals, etc.).
- The key result of planning and scheduling with RSM is a *target schedule* in the form of a graphical production diagram to be followed during construction.
- Repetitive tasks are modeled using three types of activities: *line* activities (production lines), *block* activities, and *bar* activities (Vorster et al 1992, Ioannou and Yang 2015).
- Repetitive tasks that require resource-work continuity (to avoid crew comeback delays and unproductive waiting) are explicitly defined and are scheduled to ensure the continuous workflow for the associated crews.
- RSM is a *resource-restrained scheduling technique* that ensures resource-work continuity. Thus, it may result in *longer* project durations relative to early-start schedules that are not constrained to provide continuous work for each crew.
- RSM is also a *resource-driven scheduling technique* where planners control the durations of activities through the associated *resource production rates (e.g.*, $m^2$/day for a crew of carpet installers), which depend on (a) the type of work, (b) the method selected, and (c) the crew size and composition. Planners can adjust project duration by fine-tuning the resources and methods that control activity durations (Harris and Ioannou 1998a,1998b).
- RSM is also driven by *work quantities* (e.g., $m^2$/floor), which can vary by repeating unit, and which, along with *resource production rates*, control the *unit production rates* (slopes) of activities (e.g., $m^2$/day / $m^2$/floor = floors/day = slope of production line). Hence, a single RSM production line can have multiple slopes that vary from unit to unit.
- RSM activities may repeat only in some units but not in others, and may also have discontinuities (work breaks, buffers) if so desired. Moreover, production lines may apply only over a subset of repetitive units, may need to represent out-of-sequence unit work, and may need to switch the direction of work (Harris and Ioannou 1998a, 1998b).

Work progress in an RSM production diagram can be measured using a *uniform counting scale* (where each *unit* may have a *non-uniform* work quantity) or a *dominant-attribute scale* (such as $m^2$ or km) that may have *non-uniform repetitive unit intervals* (work segments). In all cases, the *type* and *amount of work* in each repetitive unit do not need to be uniform. A detailed discussion of measuring work progress for both discrete and continuous projects, along with a

description of all RSM modeling elements, can be found in (Ioannou and Yang 2015).

## Control Links and Control Points

The scheduling positions of line, block, and bar activities in an RSM target schedule are determined by *controlling points* and *controlled points* that mark the start and end of *control links*. These key scheduling concepts are best illustrated by example.

Fig. 1 shows the RSM target schedule for a simple project with five activities that repeat over five identical units (such as floors in a building). Following the convention for discrete repetitive projects, work progress is shown on the ordinate using a counting scale from 1 to 5. Time in days is measured on the abscissa using a continuous scale from 0 to 48. Repeating work is represented by five production lines, A, B, C, D, E, each performed by a separate crew. The *slope* of each line activity in Fig. 1 is given by its *unit production rate* (*upr*) in *units per day*: (A) 1/2, (B) 1/5, (C) 1/3, (D) 1, (E) 1/4. Activities are performed serially in alphabetical order. Hence, in each repetitive unit *i*, work must be performed in the order $A_i$, $B_i$, $C_i$, $D_i$, $E_i$.

Clearly, the five continuous production lines in Fig. 1 ensure that the corresponding crews can work continuously from one unit to the next without interruptions. For example, the crew for activity C can start work in unit 1 at time 15, move to unit 2 at time 18, and continue through the rest of the units uninterrupted until it completes unit 5 at time 30. Because activity C is scheduled as one continuous production line, the *resource availability precedence links* that connect task $C_i$ to task $C_{i+1}$ in successive repetitive units have zero lag, $\text{Lag}(C_i, C_{i+1})=0$, i.e., zero length, and are hidden by the small circles on line activity C.

Line activity A has no predecessors and is scheduled to start in unit 1 at time $t = 0$. Because its *upr* is 1/2, it completes all five units in 10 days. Activity B succeeds activity A and has a *upr* of 1/5. It is slower than activity A and thus the production lines for A and B diverge. Based on the RSM divergence principle, the position of the production line for activity B is determined by *control link* $(A_1,B_1)$ from its predecessor A in the *first* repetitive unit as indicated by the vertical dashed arrow that joins points cgp(AB) and cdp(AB) in Fig. 1. Control link $(A_1,B_1)$ has zero lag, $\text{Lag}(A_1,B_1) = 0$. Moreover, because of divergence, it is the only *control link* between line activities A and B. The tail of the control link ($A_1$, B1) is called the controlling point, cgp(AB), and represents the completion of activity A in unit 1. The tip of the control link $(A_1,B_1)$ is called the *controlled point*, cdp(AB), and represents the start of activity B in unit 1.

These fundamental RSM scheduling concepts can be summarized and formalized as follows:

- A *control link* $(X,Y)$ is a precedence link with *zero* lag, $\text{Lag}(X,Y)=0$, between predecessor

line activity *X* and successor line activity *Y*.

- A control link (*X*,*Y*) is represented by a dashed arrow that is perpendicular to the time axis so that its projection on that axis, which equals the link lag, is zero.
- The tail of a control link (*X*,*Y*) lies on the *controlling predecessor* line activity *X* and is called the *controlling point*. The end of a *finishing activity (e.g., activity E in Fig. 1) is also a controlling point, as it determines* the duration of the target schedule.
- The tip of the control link (*X*,*Y*) lies on the successor line activity *Y* and is called the *controlled point*. The *controlled point* for a *start* activity (i.e., an activity with no predecessors) is located at its start at t = 0.
- Every activity *Y* in an RSM target schedule *must* have at least one *incoming control link* that ends on activity *Y*'s *controlled point*. This is the precedence link that determines the position of the production line for activity *Y* in the RSM target schedule.
- A single activity *X* may have multiple controlling points by which it controls the placement in the target schedule of one or more successors, *Y*, *Z*, *W*, etc.
- Some activities in an RSM target schedule may not control the placement of any successor activities in the schedule and thus may have zero *controlling points*.
- A single activity *Y* may have multiple controlled points, at different locations, which result in the *same* placement of the production line for activity *Y* in the target schedule.
- A predecessor line activity *X* and a successor line activity *Y* with parallel production lines—i.e., with the same unit production rates (slopes)—have multiple control links between them, one in each repetitive unit.

The positions of the production lines for activities A, B, C, D, E, in the target schedule shown in Fig. 1 are determined through the corresponding controlling and controlled points by the four control links (A,B), (B,C), (C,D) and (D,E), indicated by the vertical dashed arrows. The positions of these control links follow the RSM *divergence* and *convergence* rules. For example, activities A and B diverge, so the control link (A,B) is located in the first repetitive unit, Unit 1. Activities B and C converge, so the control link (B,C) is located in the *last* unit, Unit 5.

Control points and control links are the elements that determine the positions of activities in the target schedule. As illustrated below, the production diagrams for actual projects in practice can be so large and complicated that it is extremely difficult, if not impossible, to decipher the scheduling logic behind the final placement of production lines in the target schedule if control points and control links are not shown explicitly.

## Controlling Sequence and Controlling Activities

The duration of every RSM target schedule is determined by the *controlling sequence*, a continuous chain of *controlling activities* connected by control links with zero lag. The term "*controlling sequence*" was first introduced in (Harris and Ioannou 1998b) and further explained in (Ioannou and Harris 2003, Ioannou and Yang 2003, 2004). The term "*controlling activities*" is introduced in this article to describe a new algorithm, illustrated below, for directly identifying controlling activities *without* tracing first the multiple controlling sequences that an actual project may often have in practice.

Controlling activities are defined at the *repetitive unit level*, e.g., A1, B4, and C5, such that:

- Controlling activities on the *same* controlling sequence form a *continuous chain* that spans the entire target schedule from *project start* to *project finish*.
- To form a *continuous chain*, the links between successive controlling activities on the same controlling sequence must be *control links*, i.e., they must have zero lag.
- The first controlling activity on a controlling sequence must start at the beginning of the target schedule, i.e., at time zero, such as A1 in Fig. 1.
- The last controlling activity on a controlling sequence must finish the project, and its finish time must establish the duration of the target schedule, such as E5 in Fig. 1.

These requirements are the basis for tracing the potentially multiple controlling sequences that a target schedule may have and for identifying the controlling activities as described below.

It should be noted that a controlling sequence is primarily determined by the need to achieve resource-work continuity, which requires that some activities (at the repetitive unit level) be scheduled deliberately to start *later* than their CPM early-start times. This simple fact makes the controlling sequence *different* from a critical path. As shown below, a CPM critical path is in fact a *special case* of a controlling sequence. As all examples in this article illustrate, an RSM target schedule may not even have a traditional critical path that spans the entire project.

Moreover, as shown in Fig. 1, there are *four* types of activities in an RSM target schedule: (a) controlling activities that are also critical, (b) *controlling activities that are not critical*, (c) *critical activities that are not controlling activities*, and (d) neither controlling nor critical activities. It is essential to note that not all controlling activities are *critical*. This means that if these controlling activities are delayed during construction, they will *not* delay the project past its target duration but may cause a break in workflow.

## Trace Algorithm for Identifying the Controlling Sequence

The easiest way to identify the controlling sequence is to trace the continuous chain of controlling activities through the RSM target schedule, starting at the end of the project and moving *backwards* to the start of the project at time zero.

The proposed algorithm for tracing the controlling sequence in this manner will be called the *Trace Algorithm* and is described below. Fig.1 is used to provide a step-by-step example.

Step 1. Identify the activity that completes the project and determines its duration. Start the trace at the controlling point located at the end of the production line for that activity.

Step 2. Trace the production line for the *current activity* from its *controlling point* to its *controlled point*.

Step 3. If the current activity is not a starting activity, trace the incoming *control link* backwards from the *controlled point* to the *controlling point* on the production line of the controlling predecessor activity. Go to step 2.

Step 4. If the current activity is a starting activity, then stop.

## Example Application of the Trace Algorithm to Identify the Controlling Sequence

The application of the Trace Algorithm is illustrated using the example project in Fig. 1.

*Step* 1. Line activity E completes the project. Start the trace at the end of E at t=47.

*Step* 2. Trace the entire line activity E from the controlling point at its end (t=47) to its controlled point cdp(DE) at the start of E (t=27).

*Step* 3. Trace link (DE) backwards from the controlled point cdp(DE) on line activity E to the controlling point cgp(DE) on line activity D.

*Repeat Step* 2: Trace line activity D from its controlling point cgp(DE) to its controlled point cdp(CD). I.e., trace D2-D3-D4, from t=27 to t=30, over units 2, 3, and 4, in that order.

(Note: Even though the controlling sequence is *traced backwards*, from t=47 to t=0, the portion over line activity D is *traced forward in time*, from t=27 to t=30. This *reversal* in tracing direction, relative to the project time axis, when tracing activities D2-D3-D4 has important consequences as discussed below.)

*Repeat Step* 3: Trace link (CD) backwards from the controlled point cdp(CD) on line activity D to the controlling point cgp(CD) on line activity C.

*Repeat Step* 2: Trace line activity C from its controlling point cgp(CD) to its controlled point cdp(BC). This traces only C5 from t=30 to t=27.

*Repeat Step* 3: Trace link (BC) backwards from the controlled point cdp(BC) on line activity

C to the controlling point cgp(BC) on line activity B.

*Repeat Step* 2: Trace line activity B from its controlling point cgp(BC) to its controlled point cdp(AB). This traces the entire line activity B from t=27 to t=2.

*Repeat Step* 3: Trace link (AB) backwards from the controlled point cdp(AB) on line activity B to the controlling point cgp(AB) on line activity A.

*Repeat Step* 2: Trace line activity A from its controlling point cgp(AB) to its controlled point at its start at t=0. This traces A1 from t=2 to t=0.

*Step 4.* The trace has reached the start of the project at time t=0. Stop.

The *controlling sequence* consists of the following *controlling activities* traced *at the repetitive unit level*, from the end of the project to the start of the project (grouped by line activity, from the start to the end of the target schedule):

A1, (B1, B2, B3, B4, B5), C5, (D4, D3, D2), (E1, E2, E3, E4, E5)

To adjust project duration, planners should concentrate on these controlling activities and fine-tune their durations by adjusting the crews or the construction methods.

## Project Duration and Algebraic Signs of Controlling Activities

The Trace Algorithm can be augmented to provide a direct connection between *the duration of the target schedule* and *the durations of the controlling activities* as follows:

1. Perform the Trace Algorithm and identify the controlling activities.
   For the target schedule in Fig. 1, start at t=47, and trace the controlling sequence to t=0.
2. Mark as *positive* those controlling activities that are traced *backward* in time (relative to the project time axis). Those activities are scheduled in *push* mode.
   In Fig. 1, the activities with positive durations, in the order traced, are
   (E5, E4, E3, E2, E1), C5, (B5, B4, B3, B2, B1), A1.
3. Mark as *negative* those controlling activities that are traced *forward* in time (relative to the project time axis). Those activities are scheduled in *pull* mode.
   In Fig. 1, activities with negative durations, in the order traced, are (D2, D3, D4).
4. The algebraic sum, Σ, of the durations of all controlling activities on the same controlling sequence, using the algebraic signs determined in steps 2 and 3, gives the duration of the target schedule.

For the example in Fig. 1, the algebraic sum, Σ, of the durations of the controlling activities (organized in order, from the start of the controlling sequence to the end of the project) is:

$$\Sigma = 2+(5+5+5+5+5)+3+(-1-1-1)+(4+4+4+4+4) = 47 \text{ days}$$

As expected, the sum, Σ, matches the duration of the target schedule, i.e., 47 days.

The concept of algebraic signs in RSM is novel and directly connects it with *push-pull* concepts in lean manufacturing. Activities with positive (negative) durations are scheduled in *push* (*pull*) mode by predecessors (successors). Clearly, a traditional CPM critical path is a special case of a controlling sequence where all activities are scheduled in *push* mode, i.e., they are traced backward in time and are assigned positive signs (i.e., none have negative durations). Hence, the sum of the durations of all activities on a CPM critical path gives the project duration.

## RSM Paradox

The above procedure for determining the duration of a target schedule, as the algebraic sum of the durations of the activities on the controlling sequence, can also be used to revisit the RSM paradox, first described in Harris and Ioannou (1998a, 1998b).

> Simply put, the RSM paradox occurs when a *decrease* in the resources allocated to an activity in an RSM target schedule (a) *increases* the duration of the activity, but (b) benefits the project by *decreasing* (instead of increasing) the duration of the target schedule. (The RSM paradox can also work in the opposite direction when an increase in the resources allocated to an activity shortens the activity but increases project duration.)

As an example, consider taking away resources from line activity D in Fig. 1 to slow D down and possibly save money. Traditional CPM thinking would indicate that increasing the duration of any activity that controls project duration would also increase the duration of the target schedule. Yet, the opposite occurs. *Increasing* the unit duration of D and *then rescheduling the project to maintain resource-work continuity*, *decreases* the duration of the target schedule.

As a numerical example, suppose that slowing down activity D will increase its duration from 1 day/unit to 3 days/unit. The resulting target schedule is shown in Fig. 2 and has a duration of 41 days. The RSM paradox in this case is that taking away resources and slowing down activity D ends up speeding up the target schedule by 6 days, from 47 days down to 41 days.

The RSM paradox can be explained easily by considering the algebraic sum, Σ. In this case, increasing the unit duration of activity D *decreases* the duration of the target schedule (from 47 to 41 days), because activities D2-D3-D4 have a *negative* contribution to the algebraic sum, Σ.

$$\Sigma = 2+(5+5+5+5+5)+3+(-3-3-3)+(4+4+4+4+4) = 41 \text{ days}$$

In general, the RSM paradox can be explained using algebraic signs as follows:

> *Increasing (decreasing) the duration of controlling activities that have a negative sign in the sum, Σ, has the opposite effect and decreases (increases) project duration.*

The RSM paradox can also be explained graphically. Slowing down activity D to 3 days/unit causes its production line in Fig. 1 to rotate *clockwise* around its *controlled point* cdp(CD) until it becomes parallel with line activity C, as shown in Fig. 2. The *controlling point* cgp(DE) on line activity D and the *controlled point* cdp(DE) on line activity E can now occur earlier by 6 days (from t=27 to t=21), which allows line activity E to finish earlier by 6 days. And this reduces the duration of the target schedule by 6 days, from t=47 to t=41.

## Scheduling Principles based on the Algebraic Signs of Controlling Activities

The assignment of algebraic signs to the durations of controlling activities and the RSM paradox lead to the following RSM scheduling principles:

- Slowing down controlling activities with negative contributions to the algebraic sum Σ decreases the duration of the target schedule and may also reduce costs. Thus, the controlling activities on the same production line that have negative contributions to the algebraic sum Σ should have their durations increased (to reduce project duration) by (a) reducing their resources and/or (b) switching to a slower and less expensive method.
- When *three* consecutive production lines whose slopes are "slow-fast-slow" follow each other on the same controlling sequence (i.e., when *convergence* is followed by *divergence*), the controlling activities on the "fast" production line in the middle have negative signs in the algebraic sum Σ. Thus, it is beneficial to slow down a production line that is faster than both its controlling predecessor and its controlled successor on the same controlling sequence (e.g., activity D in Fig. 1) as this will reduce the duration of the target schedule and may also reduce costs.
- In general, it is beneficial to select or adjust the slope of a production line to be *between* the slope of a controlling predecessor and the slope of a controlling successor (e.g., in Fig. 1, select {1/2 ≥ slope of B ≥ 1/3}, and/or {1/3 ≥ slope of D ≥ 1/4}). The resulting production line slopes then follow one of two progressions, either "slowest-faster-fastest" or "fastest-slower-slowest". These slope progressions reduce project duration and also eliminate controlling activities with negative signs from the algebraic sum Σ.

Clearly, the last two principles apply when all three production lines have continuous straight lines with constant slopes. A production line with different amounts of work in each repetitive unit has multiple slopes and must be adjusted using the first principle.

## Parallel Line Activities and Multiple Controlling Sequences

A comparison of the target schedules in Fig. 1 and Fig. 2 shows that when line activity D rotates clockwise and becomes parallel with line activity C, it increases the control links between C and D from just one in Unit 5, to five, one in each repetitive unit. As discussed below, multiple control links between parallel line activities create *multiple controlling sequences*.

Consider tracing backwards the controlling sequence for the target schedule in Fig. 2. When the Trace Algorithm reaches the *controlling point* cgp(DE) on line activity D, it can proceed next to any one of five *controlled points* cdp(CD)$_i$ on activity D. Having selected one of these five controlled points, the trace continues backwards through the incoming control link ($C_i$,$D_i$) to the corresponding *controlling point* cgp(CD)$_i$ on line activity C and from there to the single *controlled point* cdp(BC) on activity C. The trace continues to the start of the project as before.

Thus, the target schedule in Fig. 2 has five different controlling sequences that can be differentiated by the control link ($C_i$,$D_i$) they go through. Notice that activities A1 and the entire line activities B and E belong to all five controlling sequences. The individual activities $C_i$ and $D_i$ on each controlling sequence vary as indicated below.

The five controlling sequences and the algebraic sum of their activity durations (where only activities $C_i$ and $D_i$ have negative durations) can be summarized as follows:

1. A1, (B), C4, C3, C2, D1, (E): $\Sigma = 2+(25)-3-3-3+3+(20) = 41$ days
2. A1, (B), C4, C3 (E): $\Sigma = 2+(25)-3-3+(20) = 41$ days
3. A1, (B), C4, D2, (E): $\Sigma = 2+(25)-3-3+(20) = 41$ days
4. A1, (B), D3, D2, (E): $\Sigma = 2+(25)-3-3+(20) = 41$ days
5. A1, (B), C5, D4, D3, D2, (E): $\Sigma = 2+(25)+3-3-3-3+(20) = 41$ days

As expected, all five controlling sequences give the same algebraic sum $\Sigma = 41$ days, which equals the duration of the target schedule in Fig. 2.

## Bounds for Activities on Multiple Controlling Sequences

The Trace Algorithm for identifying the activities on a controlling sequence backwards enters each line activity at one of its *controlling points* and traces that line activity until it exits at one of its *controlled points*. In Fig. 2, for example, the trace enters activity D at its controlling point cgp(DE) and traces activity D from there to one of its five controlled points, cdp(CD)$_i$.

In general, the Trace Algorithm can *enter* a generic line activity *Y* at one of its controlling points cgp(*YZ*)$_j$ and trace a segment of activity *Y* from there to one of its controlled points cdp(*XY*)$_i$. Each controlling point cgp(*YZ*)$_j$ on activity *Y* and the set of controlled points

$\{\text{cdp}(XY)_i\}$, on activity $Y$ can have one of three possible arrangements with respect to their time coordinate, $t$:

$$\begin{aligned} \text{cgp}(YZ)_j &\le \min\{\text{cdp}(XY)_i\} \le \max\{\text{cdp}(XY)_i\} \\ \min\{\text{cdp}(XY)_i\} &\le \text{cgp}(YZ)_j \le \max\{\text{cdp}(XY)_i\} \\ \min\{\text{cdp}(XY)_i\} &\le \max\{\text{cdp}(XY)_i\} \le \text{cgp}(YZ)_j \end{aligned}$$

In all three arrangement possibilities, when the traces of *all* controlling sequences that *enter* through the *same* controlling point $\text{cgp}(YZ)_j$ are considered together, activity $Y$ is traced between the following two bounds with respect to the time coordinate, $t$:

$$\min\{\ \text{cgp}(YZ)_j\ ;\ \min\{\text{cdp}(XY)_i\}\}$$
$$\max\{\ \text{cgp}(YZ)_j\ ;\ \max\{\text{cdp}(XY)_i\}\}$$

For example, a trace that enters activity D at its controlling point cgp(DE) and traces activity D from there to $\text{cdp}(CD)_5$ marks activities D2, D3, and D4. A trace that enters activity D at the same controlling point cgp(DE) and traces activity D from there to $\text{cdp}(CD)_1$ marks activity D1. Together, these two traces mark activity D from t=18 to t=30 and identify all controlling activities $D_i$ (i.e., D1, D2, D3, D4) that belong to a controlling sequence, as shown in Fig. 2.

In the case *of multiple controlling sequences*, a trace can *enter* the generic line activity $Y$ at any one of *several* controlling points $\text{cgp}(YZ)_j$. Thus, activity $Y$ is marked, by *all* traces taken together, between the following two general bounds with respect to the time coordinate, $t$:

$$\text{Lo} = \min\{\ \min\{\text{cgp}(YZ)_j\}\ ;\ \min\{\text{cdp}(XY)_i\}\} = \min\{\ \text{cgp}(YZ)_j\ ;\ \text{cdp}(XY)_i\}$$
$$\text{Hi} = \max\{\ \max\{\text{cgp}(YZ)_j\}\ ;\ \max\{\text{cdp}(XY)_i\}\} = \max\{\ \text{cgp}(YZ)_j\ ;\ \text{cdp}(XY)_i\}$$

Note that the controlling points $\text{cgp}(YZ)_j$ on $Y$ include *only* those that can be *traced backward* from successor line activities $Z$ that *are on a controlling sequence*. All other controlling points $\text{cgp}(YW)_k$ on activity $Y$ that lead to successors $W$ not on a controlling sequence are ignored.

For line activity C in Fig. 2 the backward trace can *enter* at one of five controlling points $\text{cgp}(CD)_i$ and proceed to a single controlled point cdp(BC) = 27. Thus, the two bounds for C are:

$$\text{Lo} = \min\{\ 18, 21, 24, 27, 30\ ;\ 27\ \} = 18$$
$$\text{Hi} = \max\{\ 18, 21, 24, 27, 30\ ;\ 27\ \} = 30$$

This means that activity C is traced by all five controlling sequences between t=18 and t=30, and as a result, C2, C3, C4, and C5 belong to at least one controlling sequence.

The controlling point on activity D in Fig. 2 is cgp(DE) = 21 and the Lo and Hi bounds are:

$$\text{Lo} = \min\{\ 21\ ;\ 18, 21, 24, 27, 30\ \} = 18$$
$$\text{Hi} = \max\{\ 21\ ;\ 18, 21, 24, 27, 30\ \} = 30$$

Hence, the combined effect of tracing all five controlling sequences is to mark line activity D between t=18 and t=30, which means that D1, D2, D3, and D4 belong to at least one controlling sequence. Notice that the Lo and Hi bounds, as stated, do not reflect the direction in which each activity $D_i$ is traced and cannot be used to assign algebraic signs to controlling activities directly.

The bounds Lo and Hi are new and play an important role in RSM theory:

- The existence of the bounds Lo and Hi proves that the *union* of all controlling activities on the *same* production line that are traced by *multiple* controlling sequences forms *one continuous segment* (i.e., they do not form several discontinuous segments).
- The bounds Lo and Hi allow the direct identification of *all* controlling activities (at the repetitive unit level) that belong to one or more controlling sequences, *without having to trace each individual controlling sequence* through the target schedule.

Thus, the bounds Lo and Hi are the foundation for an efficient algorithm that identifies all controlling activities directly, without first having to trace all controlling sequences.

## The Direct Identification of Controlling Activities

The application of the Trace Algorithm to identify the controlling sequence is straightforward and provides a clear understanding of how the controlling sequence, through the algebraic durations of the controlling activities, determines the duration of the target schedule.

However, it is not uncommon for large and complex real-life projects to have many parallel production lines and thus multiple controlling sequences. For example, the target schedule for the apartment building complex described below has more than 8 billion controlling sequences. Tracing each controlling sequence would overwhelm even the fastest computers, especially if it is done in real-time to support interactive decision-making during planning and scheduling.

The following LoHi Algorithm, which uses the bounds Lo and Hi to directly identify all controlling activities on all controlling sequences, is much less demanding in terms of effort.

## The LoHi Algorithm for the Direct Identification of Controlling Activities

Step 1. Identify the sequence step for each line activity as described in (Harris 1978) and in (Harris and Ioannou 1998a, 1998b). Place in a processing queue the line activities in the last sequence step whose finish times tie and equal the duration of the target schedule.

Step 2. If the processing queue is empty, stop. Otherwise, remove the next line activity from the *front* of the processing queue and make it the *current line activity*, *Y*.

Step 3. For the *current line activity*, *Y*, identify the time coordinates for each controlled point,

cdp($XY$)$_i$ and each controlling point, cgp($YZ$)$_j$, that leads to a *controlling* activity $Z$.

Step 4. Mark the segment of the current line activity, $Y$, between the time bounds Lo and Hi as belonging to a controlling sequence:

$$\text{Lo} = \min\{\ \text{cgp}(YZ)_j\ ;\ \text{cdp}(XY)_i\ \}$$

$$\text{Hi} = \max\{\ \text{cgp}(YZ)_j\ ;\ \text{cdp}(XY)_i\ \}$$

Step 5. For the currently selected line activity, $Y$, examine all incoming control links ($X$,$Y$) and place all controlling predecessors, $X$, into the processing queue (if not already there) sorted so that activities with the largest sequence step are in the front. Go to step 2

Step 6. If the processing queue is empty, stop.

## Example Application of the LoHi Algorithm for Identifying Controlling Activities

The application of the HiLo Algorithm is illustrated using the example target schedule with five controlling sequences shown in Fig. 2.

*Step* 1. Line activities A, B, C, D, E are in sequence steps 1, 2, 3, 4, 5, respectively. Only activity E is in sequence step 5. Place activity E in the processing queue.

*Step* 2. Remove activity E from the processing queue and make it the current line activity.

*Step* 3. E has a controlled point at its start, at t=21, and a controlling point at its end, at t=41.

*Step* 4. For activity E, Lo = min(21,41) = 21 and Hi = max(21,41) = 41. Hence, mark the entire line activity E from t=21 to t=41 as belonging to a controlling sequence.

*Step* 5. Line activity E is controlled by line activity D, which is placed in the processing queue.

*Repeat Step* 2. Remove line activity D and make it the current line activity.

*Repeat Step* 3. D has 1 controlling point at t=21 and 5 controlled points, t=18, 21, 24, 27, 30.

*Repeat Step* 4. Mark line activity D between Lo = min(18, 21, 24, 27, 30) = 18 and Hi = max(18, 21, 24, 27, 30) = 30 as belonging to a controlling sequence.

*Repeat Step* 5. D is controlled by its predecessor C, which is placed in the processing queue.

*Repeat Step* 2. Line activity C becomes the current activity.

*Repeat Step* 3. C has 5 controlling points at t=18, 21, 24, 27, 30, and 1 controlled point, t=27.

*Repeat Step* 4. Mark line activity C between Lo = min(18, 21, 24, 27, 30) = 18 and Hi = max{18, 21, 24, 27, 30} = 30 as belonging to a controlling sequence.

*Repeat Step* 5. C is controlled by its predecessor B which is placed in the processing queue.

*Repeat Step* 2. Line activity B becomes the current activity.

*Repeat Step* 3. B has 1 controlling point at t=27, and 1 controlled point at t=2.

*Repeat Step* 4. Mark line activity B between Lo = min(2,27) = 2 and Hi = max{2,27} = 27 as belonging to a controlling sequence.

*Repeat Step* 5. B is controlled by its predecessor A, which is placed in the processing queue.

*Repeat Step* 3. A has 1 controlling point at t=2, and 1 controlled point at t=0.

*Repeat Step* 4. Mark line activity A between Lo = min(0,2) = 0 and Hi = max{0,2} = 2 as belonging to a controlling sequence.

*Step* 6. A does not have any predecessors, and the processing queue is empty, so stop.

The activities identified as belonging to a controlling sequence are the same as those marked by the Tracing Algorithm and indicated in Fig. 2.

It should be noted that the LoHi Algorithm for the direct identification of controlling activities considers *each line activity only once*, in reverse sequence-step order, and *marks all controlling activities associated with that line activity in a single step*. Thus, it does not determine the *number* of multiple controlling sequences, nor does it need *to trace all controlling sequences* (which could be in the billions) to identify the controlling activities. As a result, the LoHi Algorithm is efficient, and, as indicated below, its computer implementation can quickly identify all controlling activities, even for a project with billions of controlling sequences. This enables the identification of controlling activities interactively and indicates to management where to focus attention for fine-tuning the duration of the target schedule during planning and scheduling.

## Multiple Controlling Sequences

The number of controlling sequences for a target schedule with *p consecutive pairs* of parallel line activities and *n* repetitive units is an exponential function of *p*,

$$\text{Number of controlling sequences} =\sim n^p.$$

Thus, even modest values of *n* and *p* can easily result in an extremely large number of controlling sequences (e.g., in the billions), making the Trace Algorithm impractical. In contrast, the LoHi Algorithm for the direct identification of controlling activities is not affected by the number of parallel line activities, *p*, because it considers each line activity only once.

The theoretical number of controlling sequences in a target schedule becomes very large because it includes all *n* control links between each pair of parallel line activities. In practice, however, only the first and last of the *n* control links (i.e., those with the earliest and latest times) are needed for sensitivity analysis. For example, of the five control links between parallel line activities C and D in Fig. 2, only links ($C_1$,$D_1$) and ($C_5$,$D_5$) are important. As explained earlier,

these two control links define the controlled points $cdp(CD)_1$ and $cdp(CD)_5$, which serve as pivot points around which line activity D rotates when resources are added or removed to change its unit production rate (slope). In most practical cases, the remaining controlled points, $cdp(CD)_2$, $cdp(CD)_3$, and $cdp(CD)_4$, along with the associated controlling sequences, can be ignored.

## Continuous Repetitive Projects

RSM can be used to develop target schedules for both projects where work repeats over discrete units or over continuous locations. Fig. 3 shows an example application of RSM on a project with continuous work locations. This project has three line activities, A, B, and C, that follow each other and repeat over the entire continuous workspace of 32 stations. A distance buffer of 8 stations is required between line activities at all locations to prevent interference between crews. Following the convention for continuous repetitive projects, work progress is shown on the abscissa using stations (which can represent any convenient unit of length), and time is measured on the ordinate in days. The three activities are performed by separate crews. Activities A and C have a unit production rate (slope) of 4 stations per day, whereas activity B has a higher unit production rate of 8 stations per day. Activities A, B, and C are scheduled using the convergence and divergence rules to establish their controlled and controlling points, as shown in the target schedule in Fig. 3. The resulting project duration is 14 days.

## Application of the Trace Algorithm to a Continuous Repetitive Project

The Trace Algorithm for identifying the controlling sequence for the target schedule in Fig. 3 starts at the end of line activity C, at s=32 and t=14, which by convention is its controlling point. The trace continues to the controlled point of C, cdp(BC), and continues backwards through link (B,C) to the controlling point on line activity B, cgp(BC). The process is then repeated. The trace continues to the controlled point of B, cdp(AB), and then backwards through link (A,B) to the controlling point on line activity A, cgp(AB). The trace follows activity A to its controlled point at the start of the project, where it stops. The result is the controlling sequence indicated on Fig. 3, which includes activity A, followed by the segment B2 on activity B, and ends with activity C.

The algebraic sum Σ of activity durations is as follows (where only the segment B2 has a negative duration), and matches the duration of the target schedule in Fig. 3, i.e., 14 days:

$$\Sigma = [\ t(\mathrm{A}) - t(\mathrm{B2}) + t(\mathrm{C})\ ] = 8 - 2 + 8 = 14 \text{ days}$$

## Application of the LoHi Algorithm to a Continuous Repetitive Project

The LoHi Algorithm can also directly identify the activity segments that belong to the

controlling sequence. The finishing line activity C becomes the current activity, and its segment on the controlling sequence is set by the bounds Lo=min{6,14}=6 and Hi=max{6,14}=14. Thus, the entire line activity C is on a controlling sequence. Its controlling predecessor, line activity B, is placed in the processing queue and then immediately removed to be examined next. Its bounds are Lo=min{6,8}=6 and Hi=max{6,8}=8. Hence, only segment B2 belongs to a controlling sequence. Line activity A is the controlling predecessor of B and is examined next. Its bounds are Lo=min{0,8}=0 and Hi=max{0,8}=8. Thus, the entire line activity A is on a controlling sequence. At this point, the processing queue is empty, and the algorithm stops. The result is the same controlling sequence as identified above and indicated on Fig. 3, which includes activity A, followed by segment B2, and ending with activity C.

The important observation from the above examples is that the same RSM concepts concerning control points, control links, the controlling sequence, its relationship to project duration, and the two algorithms for identifying activities on the controlling sequence apply equally well to both discrete and continuous projects.

## Repetitive Project Planner—RP2

The RSM methodology for scheduling projects with repetitive activities subject to resource work continuity has been implemented in RP2, a computer program developed by the authors to validate the proposed scheduling approach. RP2 has been used in practice to schedule both continuous and discrete projects, including the apartment building complex presented below. The graphical RSM schedules, as well as the tabular reports for the apartment building complex, were generated by RP2 and provide complete details for schedule development and sensitivity analysis, allowing someone knowledgeable about the field to repeat the study and verify the results. RP2 identified and marked the controlling activities as shown in the target schedules by using the LoHi Algorithm for the direct identification of controlling activities.

## Example RSM Schedule—Four-Story Residential Apartment Building Complex

The target schedule for a complex of residential apartment buildings is used to illustrate the above RSM concepts and algorithms in practice. This project was part of a land development plan for low-income housing, involving the construction of 50 four-story apartment buildings. Due to space limitations, the RSM target schedule presented below includes the work for only 4 of the 50 buildings. Fig. 4 shows a photograph of the project under construction.

Each apartment building had four identical floors, with each floor divided into 2 three-

bedroom apartments. The foundation for each building consisted of 21 concrete piles. Each group of 3 piles supported one of 7 cast-in-place concrete walls that ran parallel to each other from the front to the back of the building. The 4 cast-in-place concrete slabs and the roof for each building were constructed by the same concrete slab crew. After each floor slab was completed, 7 concrete wall crews used 7 metal forms to construct all 7 walls for the same floor simultaneously. All concrete work was performed by the general contractor.

Concrete construction was followed by interior activities. Except for final cleaning, all interior activities were performed by subcontractors, with the general contractor responsible for material and equipment. All activities, including those with similar work, used separate crews. For example, there were two tile crews, one for laying wall and floor tiles in bathrooms, and another for laying floor tiles in living rooms and bedrooms.

The natural unit of progress for this project was that of a single floor. Buildings were designated as A, B, C, and D, and the four floors in each building were numbered 1, 2, 3, and 4. Table 1 shows the complete activity list, along with the scheduled start and finish times generated by RP2. The activity ID's of structure activities start with "S" and for interior activities with "I". Precedence relationships between activities and their attributes are shown in Table 2.

Fig. 5 shows the graphical RSM target schedule generated by RP2. All activities were modeled as *line* activities, except for the casting of the roof slabs for the four buildings, which were modeled as four *bar* activities at the end of units A-4, B-4, C-4, and D-4, respectively.

The work for this project followed four different working sequences:

- Pile fabrication and driving progressed by building {A, B, C, D}.
- Concrete slab and concrete wall activities were performed first by building and then by floor {A-1, B-1, C-1, D-1, A-2, B-2, C-2, D-2, A-3, B-3, C-3, D-3, A-4, B-4, C-4, D-4}.
- The four roof activities were performed once for each building, but at *specific locations*, i.e., the *end* of units A-4, B-4, C-4, and D-4, and are shown as *bar* activities.
- Interior activities progressed first by floor and then by building {A-1, A-2, A-3, A-4, B-1, B-2, B-3, B-4, C-1, C-2, C-3, C-4, D-1, D-2, D-3, D-4}.

## Example RSM Target Schedule — Billions of Controlling Sequences

For the target schedule in Fig. 5, there were $n$=16 *control links* between each pair of parallel predecessor and successor production lines that are on at least one controlling sequence. In total, there are $p$=8 such pairs of parallel predecessor-successor line activities. Moreover, all control

sequences that go through activity I-25 could alternatively go through activity I-30, as they have the same predecessor and the same successor. Consequently, the total number of controlling sequences in Fig. 5 is very large, more than 8 billion, and equals $2n^p = 2x16^8 = 8,589,934,592$.

## Practical Limitations for the Trace Algorithm

To identify each controlling sequence for the target schedule in Fig. 5, the Trace Algorithm starts at the end of line activity I-75 in unit D-4. This is the activity that finishes the target schedule and establishes the project duration at 64.5 days. From there, the trace follows line activity I-75 backwards until it reaches one of its 16 controlled points and continues through the corresponding incoming control link to the matching controlling point on I-70.

The process is then repeated for line activity I-70. Starting at the current controlling point on I-70, the trace follows I-70 to one of its 16 controlled points (moving either *backward* or *forward* in time relative to the time axis) and through the incoming control link it continues backwards to the corresponding controlling point on I-65.

Activity I-65 has 16 controlling points, but only one controlled point, which is located at the start of unit D-4. Thus, irrespective of which of the 16 controlling points was used by the controlling sequence trace to enter activity I-65, the next step is to trace to the single controlled point and through the incoming control link to the controlling point at the end of I-60.

The trace continues to the single controlled point at the start of I-60, to the controlling point at the end of I-40 in A-1, to the controlled point at the start of I-40 in A-1, and then to the controlling point at the end of I-35 in A-1.

From there on, the tracing possibilities are too many to track individually.

## Direct Identification of Controlling Activities on Multiple Controlling Sequences

In contrast to the Trace Algorithm which must trace more than 8 billion controlling sequences to identify all controlling activities, the LoHi Algorithm for the direct identification of controlling activities considers each of the 29 production lines just once, in reverse-sequence-step-order, and identifies directly the segments on each production line (and thus the controlling activities) that are on at least one controlling sequence.

The LoHi Algorithm for identifying all controlling activities directly starts with line activity I-75, which finishes the target schedule and establishes its duration as 64.5 days. Line activity I-75 has one controlling point at t=64.5 and 16 controlled points from t=48.5 to 63.5. Thus, Lo = min{48.5, …, 63.5, 64.5} = 48.5 and Hi = max{48.5, …, 63.5, 64.5} = 64.5. These bounds

indicate that the entire line activity I-75 belongs to at least one controlling sequence.

From I-75 the algorithm continues with its controlling predecessor, line activity I-70, where the situation is almost identical and results in the same Lo and Hi bounds. Thus, all of the line activity I-70 belongs to at least one controlling sequence, also.

The controlling predecessor of I-70 is activity I-65 which has one controlled point at t=61.5 and 16 controlling points from t=47.5 to 62.5. Hence, Lo = min{47.5, …, 61.5, 62.5} = 47.5 and Hi = max{47.5, …, 61.5, 62.5} = 62.5. As a result, almost all of activity I-65, except for the segment in unit A-1, belongs to at least one controlling sequence.

From this point on, the procedure repeats as above. Each controlling predecessor line activity is removed from the front of the processing queue in reverse sequence-step order and considered next. The segment between its Lo and Hi bounds is marked as belonging to at least one controlling sequence. The LoHi Algorithm has been programmed in RP2 and marked the activities on the controlling sequence using thick lines (solid and dashed) as shown in Fig. 5. Table 3 was also produced by RP2, and for each activity shows the controlling predecessor, the control relationship, and the location of the control links.

## Sensitivity Analysis for the Example RSM Schedule

The identification of controlling activities indicated to management which activities to focus on to adjust the duration of the target schedule while still planning and scheduling the work. For brevity in presenting the sensitivity analysis, the activities of the target schedule in Fig. 5 have been aggregated into four groups with distinct characteristics.

The first group includes all concrete structure activities, S-05 to S-35. These activities are already scheduled at the minimum limits allowed for concrete curing and cannot be expedited further to shorten the project duration.

The second group includes the interior work activities, I-05 to I-45, which have *parallel* production lines. These activities were performed sequentially, except for I-25 and I-30, which were preceded by I-20 and followed by I-35, and were performed concurrently. The third group includes *parallel* activities I-60, I-70, and I-75. The comments below apply to both groups.

Because the activities in these two groups were sequential and have *parallel* production lines, any change to the unit production rate of *only one* of the activities in each group (I-05 to I-45) or (I-60 and I-70), either to speed it up (increase its slope) or to slow it down (decrease its slope), will violate the third scheduling principle presented above and will *increase* the duration of the target schedule (i.e., it is undesirable).

This is illustrated by the target schedule in Fig. 6 which shows examples of both, i.e., slowing down and speeding up a *single* activity in *each* group. For illustration purposes only, the unit production rate (slope) of activity I-10 in the second group has been *decreased* from 1 unit/day to 2/3 units/day, thus *increasing* its unit duration from 1 day/unit to 1.5 days/unit. As expected, the additional 0.5 day/unit over 16 units *increases* the duration of the target schedule by 8 days (*undesirable*). Also, the slope of activity I-70 in the third group has been *increased* from 1 unit/day to 2 units/day. Thus, its unit duration *decreased* from 1 day/unit to 0.5 days/unit. Because of the RSM paradox, however, this *reduction* in duration by 0.5 days/unit over 14 units has the *undesirable* effect of *increasing* the duration of the target schedule by 7 days.

Notice that both changes violate the third general principle that the slope of a production line should be between that of a controlling predecessor and that of a controlling successor. Thus, these two changes, slowing down I-10 and speeding up I-70, *increase* the duration of the target schedule (by 8 and 7 days, respectively) for a total of 15 days, from 64.5 to 79.5 days.

These simple examples lead to the following practical corollary for the third principle:

> *Chains of parallel line activities that are also part of at least one controlling sequence are best changed as a group (i.e., slowed down or expedited) to remain parallel.*

For example, if the *only* activities that could be expedited are I-05 to I-45, *and no other*, then increasing the slope of *all* by an equal amount so that they remain parallel would be the most effective way to reduce the duration of the target schedule.

The last group of activities includes I-40, I-50, I-55, and I-60, and is arguably the most illustrative. Since these were some of the slowest activities, it follows that perhaps they should be the first to be expedited. The line activity notation in Fig. 5, however, makes it obvious that even though activities, I-40, I-50, and I-55, are slow, they are *not* controlling activities. Only activity I-60 is on all controlling sequences (more than 8 billion), and as a result, expediting I-60 will shorten all controlling sequences, thus decreasing the duration of the target schedule.

At this point, it would be appropriate to point out the usefulness of the special line notation used by RP2 to mark activities, as shown in Fig. 5. The alternative, i.e., a target schedule that shows all activities using plain lines without markings, would not provide any indication as to which activities are important and why. For example, it would not make it evident that even though they may appear promising, expediting one of the activities I-40, I-50, and I-55 would have no impact on the duration of the target schedule.

The identification of controlling activities in Fig. 5 also indicates the pivot points for the line

activities and the extent to which each activity should be expedited. Speeding up I-60, for example, causes it to rotate counterclockwise around its controlled point at t = 33.5. Thus, decreasing its unit duration from 1.75 days/floor to 1.5 days/floor will have four important effects: (a) project duration will be reduced from 64.5 days to 60.5 days (b) I-60 will become parallel to its predecessor I-40; (c) the production lines for I-60 and I-55 (both of which are preceded by I-40 and I-45, and succeeded by I-65) will become parallel and overlap perfectly; (d) activity I-55 will also join the activities on a controlling sequence and will play exactly the same role as I-60.

Expediting line activity I-60 to have a duration shorter than 1.5 days/floor will have no effect on the duration of the target schedule because activity I-55 (whose unit duration is currently 1.5 days/floor) will take over the controlling role of I-60 and prevent any further reduction in project duration (unless I-55 is expedited along with I-60).

Clearly, the above sensitivity analysis was made practical by the LoHi Algorithm for the direct identification of controlling activities. This algorithm enabled RP2 to quickly reschedule the target schedule each time a what-if change was made, thus eliminating the need to repeatedly trace more than 8 billion controlling sequences.

## Summary and Conclusion

RSM is a scheduling technique for projects where similar work repeats over discrete units or continuous locations, producing target schedules in the form of production diagrams to be followed during construction. RSM is both a resource-restrained and a resource-driven technique that schedules activities to minimize project duration while ensuring resource-work continuity.

During planning and scheduling, the duration of an RSM target schedule is determined by the controlling sequence, a continuous chain of controlling activities that span the project from start to finish. The identification of controlling activities indicates which activities to adjust so that project duration can be fine-tuned to meet contractual and managerial objectives.

Control links, controlling points, and controlled points are the concepts that (a) determine the placement of line, block, and bar activities in an RSM target schedule, and (b) identify the controlling activities on the controlling sequence.

The Trace Algorithm has been introduced to identify all controlling sequences in an RSM target schedule and assign a positive (negative) algebraic sign to each controlling activity traced backward (forward) relative to the time axis. This leads to a new result where the algebraic sum of controlling activity durations equals the duration of the target schedule. Algebraic signs also

provided (a) a new connection to *push-pull* concepts from lean manufacturing, as controlling activities with positive (negative) signs are scheduled in push (pull) mode by their predecessors (successors); (b) a new explanation of the RSM paradox: lengthening (shortening) controlling activities with durations marked negative produces the opposite and beneficial (detrimental) result, i.e., shortens (lengthens) the target schedule; and (c) three new scheduling principles concerning the slopes of line activities.

Repetitive projects in practice often involve several sequential or parallel production lines connected by multiple control links at various locations. It has been shown that this results in extremely large numbers of controlling sequences (often in the billions), which makes the identification of controlling activities impractical by first tracing all controlling sequences.

New bounds Lo and Hi have been introduced to demonstrate, for the first time, that all controlling activities on each production line form a single continuous segment (and not several discontinuous segments), even if these controlling activities belong to multiple controlling sequences.

The LoHi Algorithm for the direct identification of controlling activities has been introduced, which considers each production line only once and uses the bounds Lo and Hi to mark the beginning and end of all controlling activities on a production line in one step, thus eliminating the onerous task of tracing billions of controlling sequences to identify the controlling activities.

A simple discrete project, a simple continuous project, and an actual project for constructing an apartment building complex illustrate the application of the new concepts and algorithms discussed above. The graphical RSM target schedules for the apartment building complex, as well as all tabular reports, were generated by RP2, a computer program developed by the authors to implement the new RSM methodology and concepts presented above. RP2 demonstrates the usefulness of identifying both controlling activities and critical activities in graphical and tabular reports to support management decisions. The identification of controlling activities indicates to planners which activities to consider for sensitivity analysis to adjust the duration of the target schedule. It is hoped that the concepts, bounds, and algorithms presented in this article will spur the development of repetitive scheduling systems with similar capabilities and their use in practice.

## Acknowledgements

The various plans and schedules for the residential apartment building project presented in this paper were developed in cooperation with Mr. Luis Riquezes, Project Manager for the work,

whose contributions are gratefully acknowledged.

## References

Ballard, G. and Tommelein, I.D. (1999). "Aiming for Continuous Flow." LCI White Paper-3, Lean Construction Institute, March 5, 1999.

Harris, R.B. (1978). “Precedence and Arrow Networking Techniques for Construction", John Wiley and Sons, New York, NY.

Harris, R.B. and Ioannou, P.G. (1998a). *"Scheduling projects with repetitive activities."*. UMCEE Report No. 98-35. Civ. and Envir. Engrg. Dept., University of Michigan, Ann Arbor.

Harris, R.B. and Ioannou, P.G. (1998b). "Scheduling projects with repeating activities." *J. Constr. Engrg. and Mgmt.*, ASCE, 124(4), 269-278.

Ioannou, P.G. and Harris R.B. (2003). “Discussion of ‘Algorithm for Determining Controlling Path Considering Resource Continuity’ by Mohammed A. Ammar and Emad Elbeltagi”, *J. Constr. Engrg. and Mgmt.*, ASCE, Vol. 17:68-70

Ioannou, P.G. and Yang I.T. (2003). “Discussion of ‘Algorithm for Determining Controlling Path Considering Resource Continuity’ by Mohammed A. Ammar and Emad Elbeltagi”, *J. Constr. Engrg. and Mgmt.*, ASCE, Vol. 17:68-70

Ioannou, P.G. and Yang I.T. (2004). “Discussion of ‘Comparison of Linear Scheduling Model and Repetitive Scheduling Method’ by Kris G. Mattila and Amy Park”, *J. Constr. Engrg. and Mgmt.*, ASCE, Vol. 130:461-463

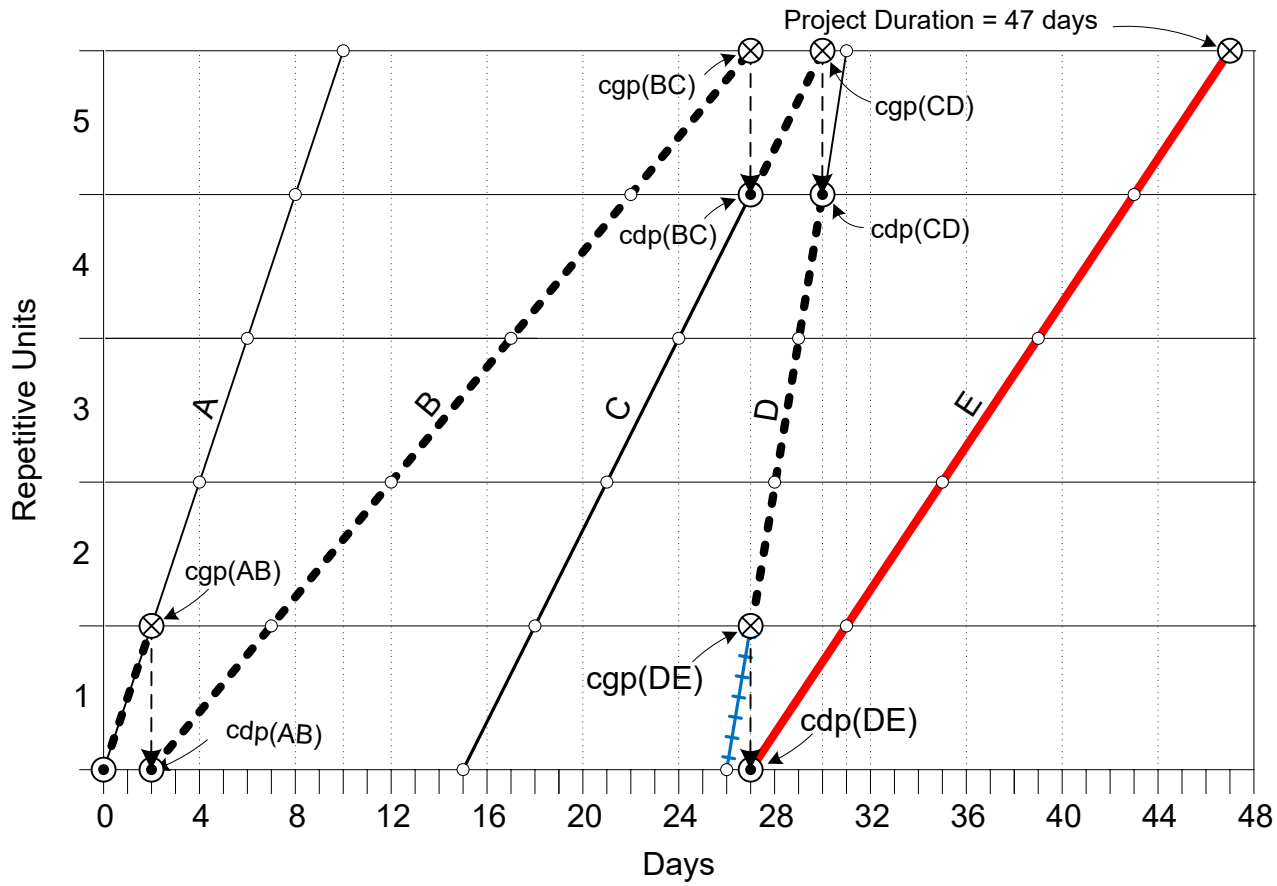
Ioannou, P.G. and Yang I.T. (2015). “Repetitive Scheduling Method—Requirements, modeling and implementation”, *J. Constr. Engrg. and Mgmt.*, ASCE, in press.

Riley, D., and Savindo, V. (1997). "Space planning for mechanical, electrical, plumbing, and fire protection trades in multi-story buildings construction." *Proc., Constr. Congr. V*, S. Anderson, ed., ASCE, New York, 102-109.

Vorster, M. C., Beliveau, Y. J., and Bafna, T. (1992). "Linear scheduling and visualization." *Transportation Research Record*. 1351, pp. 32-39.

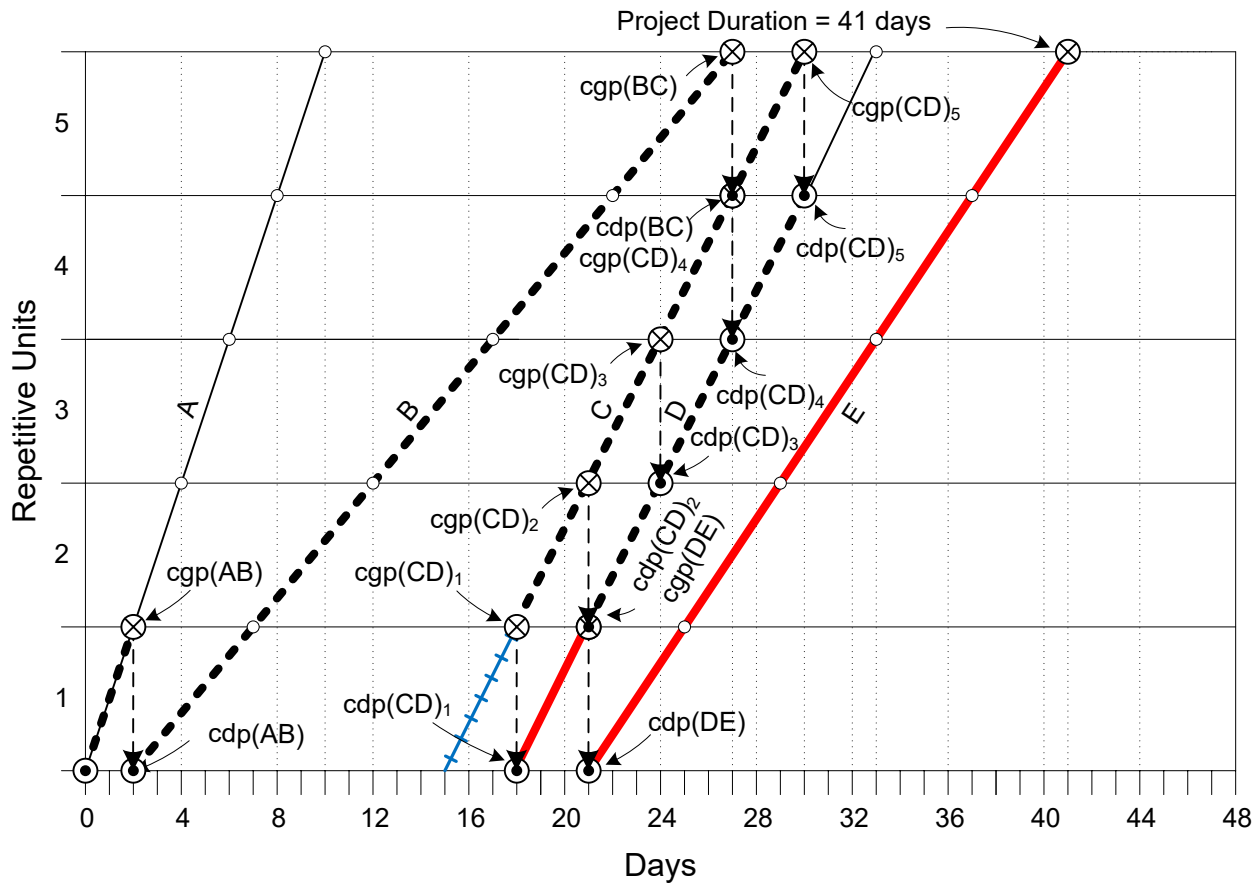
## LIST OF FIGURES

**Fig. 1.** Example RSM target schedule for discrete project ..... 26
**Fig. 2.** Example RSM target schedule with parallel activities ..... 27
**Fig. 3.** Example RSM target schedule for continuous project ..... 28
**Fig. 4.** Apartment Building Project During Construction (photograph courtesy of Luis Riquezes) ..... 29
**Fig. 5.** RSM Schedule for Residential Apartment Building Project ..... 30
**Fig. 6.** RSM Schedule — Effects of slowing I-10 & expediting I-70 ..... 31



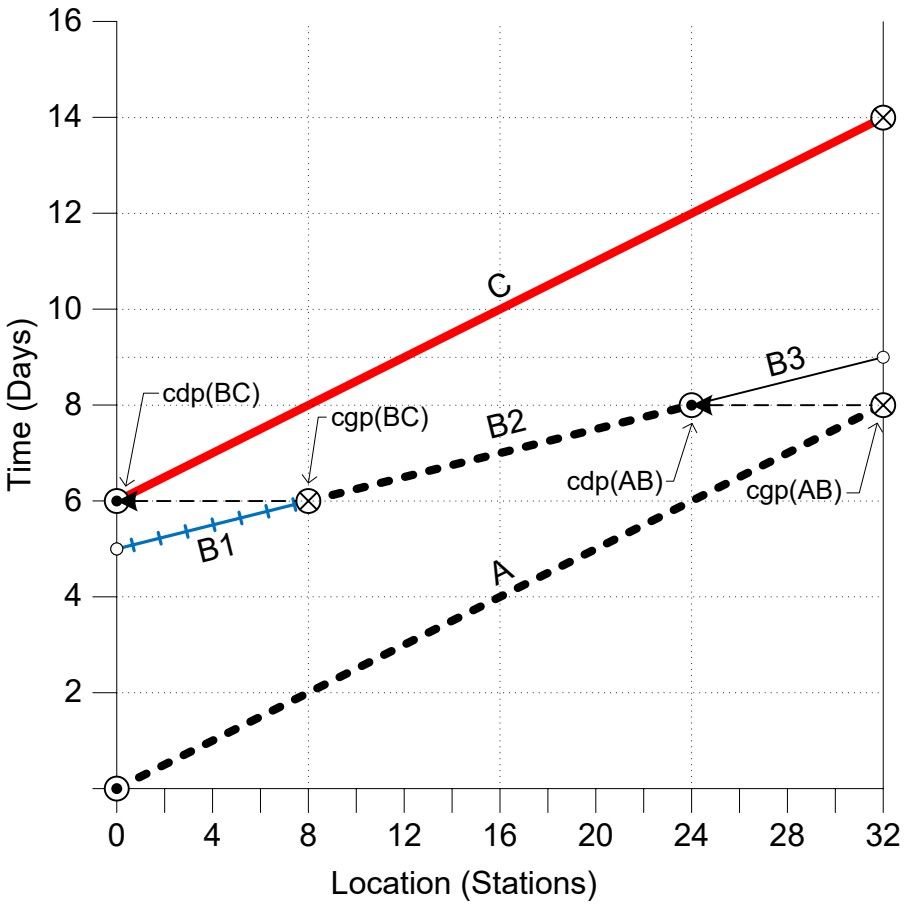

| LEGEND | | | | |
|---|---|---|---|---|
| Line activity | | On Controlling Sequence? Yes | On Controlling Sequence? No | ⊗ - - - - → ⊙ control link arrow |
| Critical? | Yes | | | ⊗ cgp(XY): controlling point for control link (XY) (arrow tail) |
| | No | | | ⊙ cdp(XY): controlled point for control link (XY) (arrow tip) |

**Fig. 1.** Example RSM target schedule for discrete project




| LEGEND | | | | |
|---|---|---|---|---|
| Line activity | | On Controlling Sequence? Yes | On Controlling Sequence? No | ⊗ - - - - → ⊙ control link arrow |
| Critical? | Yes | ▬ | ▬ | ⊗ cgp(XY): controlling point for control link (XY) (arrow tail) |
| | No | - - - | ▬ | ⊙ cdp(XY): controlled point for control link (XY) (arrow tip) |

**Fig. 2.** Example RSM target schedule with parallel activities




| LEGEND | | | | |
|---|---|---|---|---|
| Line activity | | On Controlling Sequence? Yes | On Controlling Sequence? No | control link arrow |
| Critical? | Yes | | | cgp(XY): controlling point for control link (XY) (arrow tail) |
| Critical? | No | | | cdp(XY): controlled point for control link (XY) (arrow tip) |

**Fig. 3.** Example RSM target schedule for continuous project

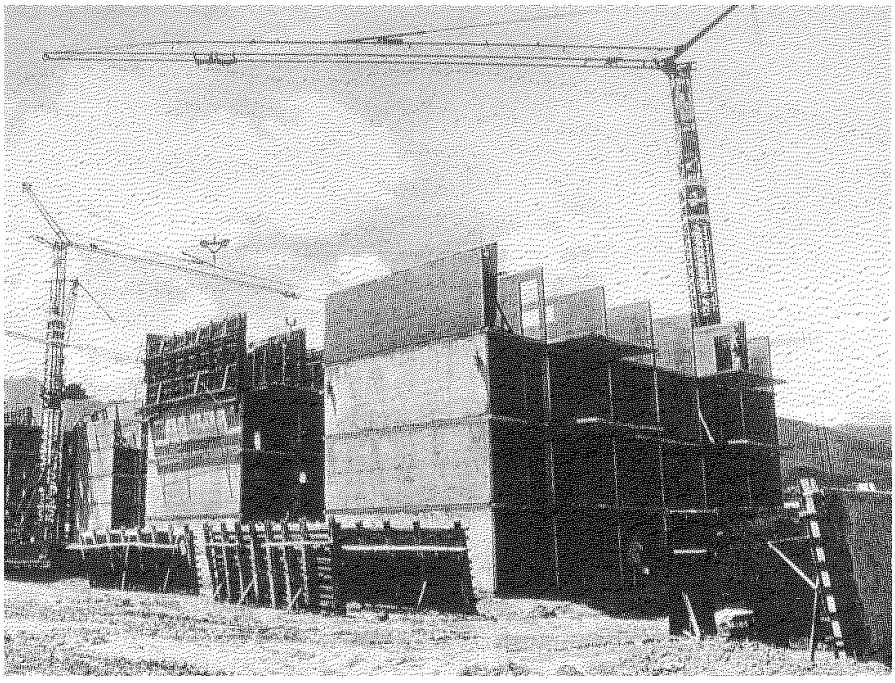


**Fig. 4.** Apartment Building Project During Construction
(photograph courtesy of Luis Riquezes)

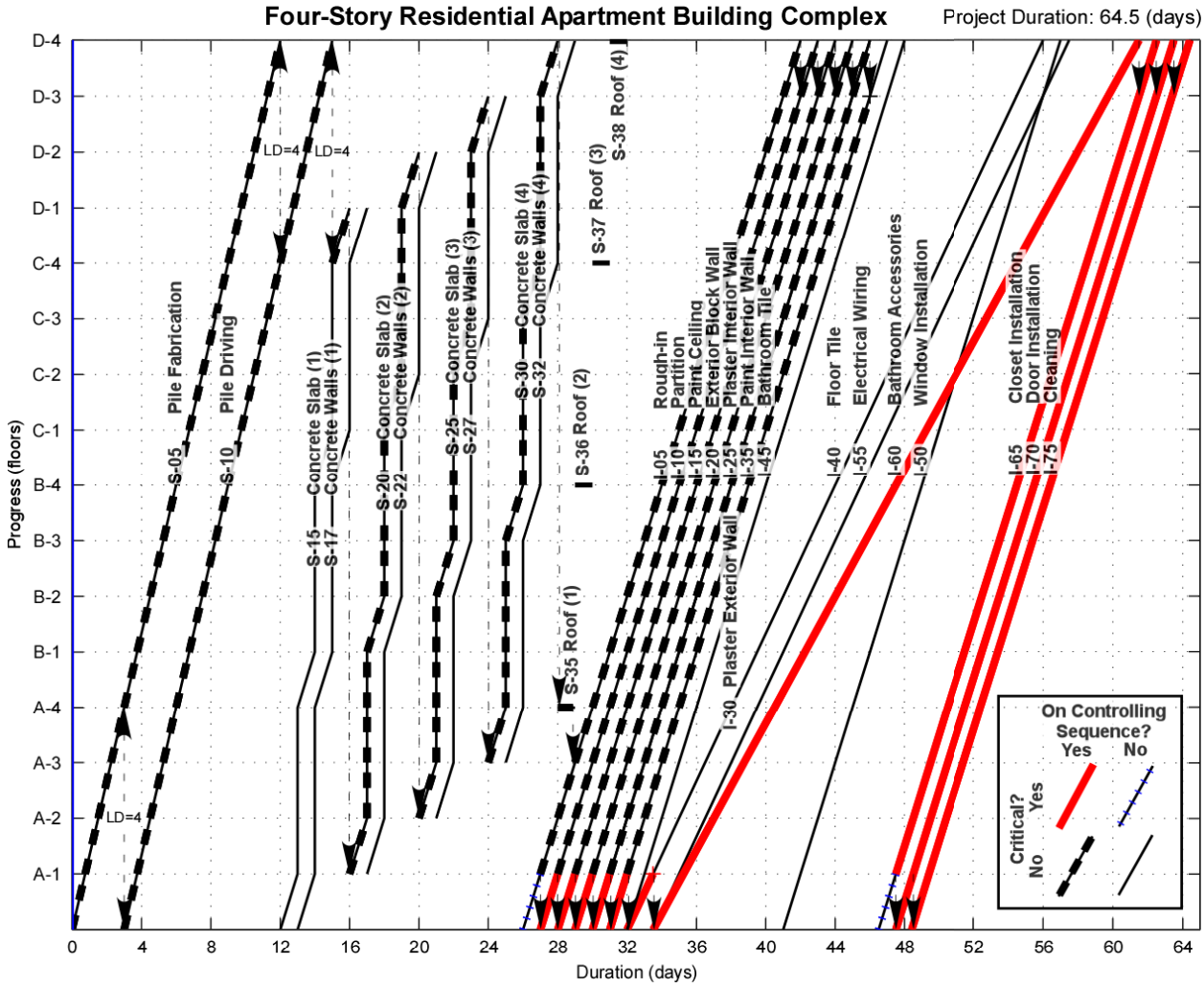



**Fig. 5.** RSM Schedule for Residential Apartment Building Project

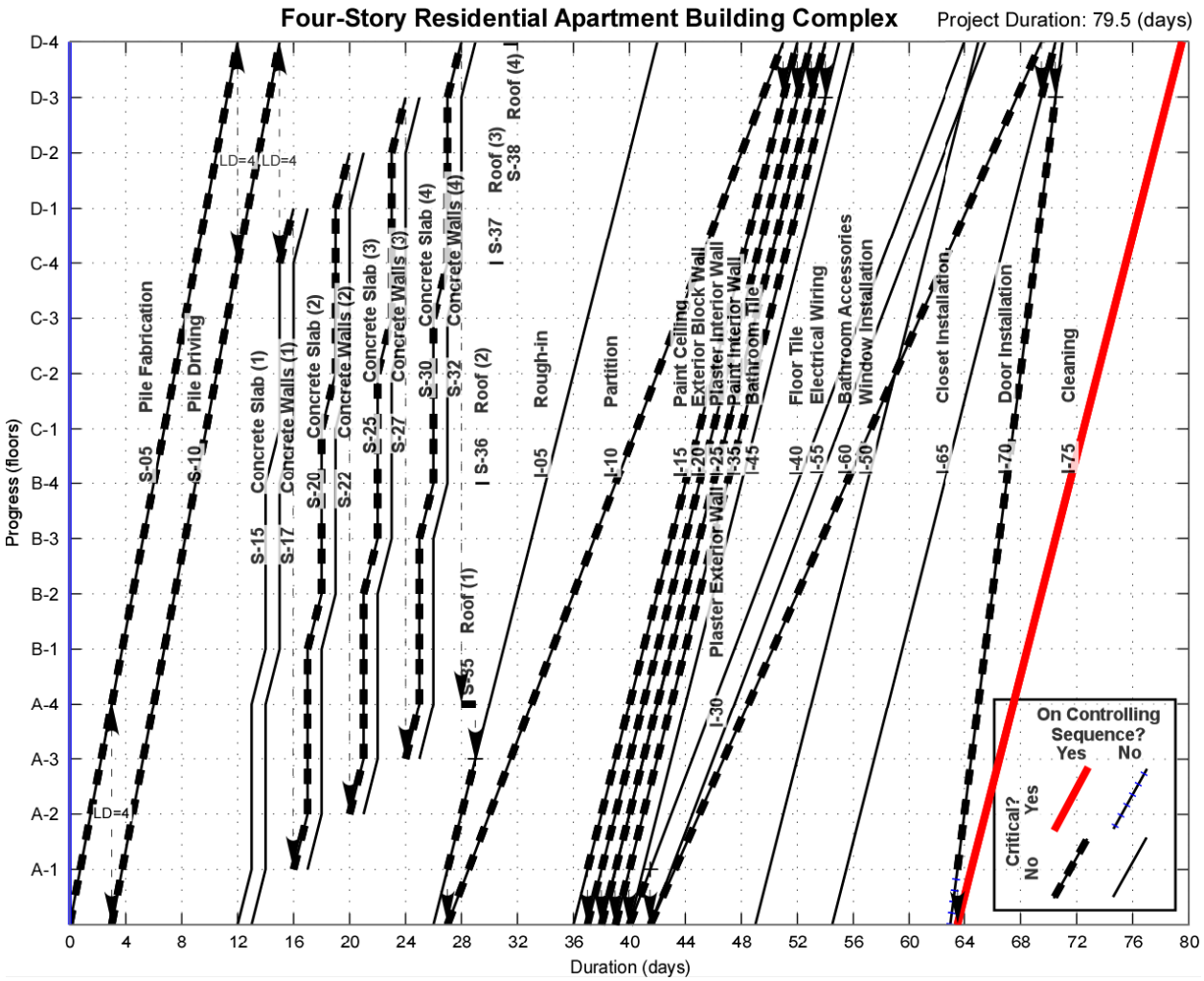



**Fig. 6.** RSM Schedule — Effects of slowing I-10 & expediting I-70

## LIST OF TABLES

**Table 1.** Activity Data and RSM Schedule for Apartment Building Project ..... 33

**Table 2.** Activity Precedence Relationships for Apartment Building Project..... 34

**Table 3.** Activity Control Relationships for Apartment Building Project..... 35

**Table 1.** Activity Data and RSM Schedule for Apartment Building Project

| Act. ID | Activity Description | Act. Type | Start Unit (floor) | Finish Unit (floor) | Start Time (days) | Finish Time (days) | Unit Dur (days) |
|---|---|---|---|---|---|---|---|
| S-05 | Pile Fabrication | Line | A-1 | D-4 | 0 | 12 | 0.75 |
| S-10 | Pile Driving | Line | A-1 | D-4 | 3 | 15 | 0.75 |
| S-15 | Concrete Slab (1) | Line | A-1 | D-1 | 12 | 16 | 1[a] |
| S-17 | Concrete Walls (1) | Line | A-1 | D-1 | 13 | 17 | 1[a] |
| S-20 | Concrete Slab (2) | Line | A-2 | D-2 | 16 | 20 | 1[a] |
| S-22 | Concrete Walls (2) | Line | A-2 | D-2 | 17 | 21 | 1[a] |
| S-25 | Concrete Slab (3) | Line | A-3 | D-3 | 20 | 24 | 1[a] |
| S-27 | Concrete Walls (3) | Line | A-3 | D-3 | 21 | 25 | 1[a] |
| S-30 | Concrete Slab (4) | Line | A-4 | D-4 | 24 | 28 | 1[a] |
| S-32 | Concrete Walls (4) | Line | A-4 | D-4 | 25 | 29 | 1[a] |
| S-35 | Roof (1) | Bar | After A-4 | Before B-1 | 28 | 29 | 1 |
| S-36 | Roof (2) | Bar | After B-4 | Before C-1 | 29 | 30 | 1 |
| S-37 | Roof (3) | Bar | After C-4 | Before D-1 | 30 | 31 | 1 |
| S-38 | Roof (4) | Bar | After D-4 | | 31 | 32 | 1 |
| I-05 | Rough-in | Line | A-1 | D-4 | 26 | 42 | 1 |
| I-10 | Partition | Line | A-1 | D-4 | 27 | 43 | 1 |
| I-15 | Paint Ceiling | Line | A-1 | D-4 | 28 | 44 | 1 |
| I-20 | Exterior Block Wall | Line | A-1 | D-4 | 29 | 45 | 1 |
| I-25 | Plaster Interior Wall | Line | A-1 | D-4 | 30 | 46 | 1 |
| I-30 | Plaster Exterior Wall | Line | A-1 | D-4 | 30 | 46 | 1 |
| I-35 | Paint Interior Wall | Line | A-1 | D-4 | 31 | 47 | 1 |
| I-40 | Floor Tile | Line | A-1 | D-4 | 32 | 56 | 1.5 |
| I-45 | Bathroom Tile | Line | A-1 | D-4 | 32 | 48 | 1 |
| I-50 | Window Installation | Line | A-1 | D-4 | 41 | 57 | 1 |
| I-55 | Electrical Wiring | Line | A-1 | D-4 | 33.5 | 57.5 | 1.5 |
| I-60 | Bathroom Accessories | Line | A-1 | D-4 | 33.5 | 61.5 | 1.75 |
| I-65 | Closet Installation | Line | A-1 | D-4 | 46.5 | 62.5 | 1 |
| I-70 | Door Installation | Line | A-1 | D-4 | 47.5 | 63.5 | 1 |
| I-75 | Cleaning | Line | A-1 | D-4 | 48.5 | 64.5 | 1 |

[a]Unit durations are entered in RP2 as: "1,0,0,0,1,0,0,0,1,0,0,0,1"

**Table 2.** Activity Precedence Relationships for Apartment Building Project

| Act. ID | Activity Description | Pred ID | Predecessor Description | Relationship Type | Linking Unit |
|---|---|---|---|---|---|
| S-05 | Pile Fabrication | N/A | N/A | N/A | N/A |
| S-10 | Pile Driving | S-05 | Pile Fabrication | Dist Ctrl[a] | A-1 ~ D-4 |
| S-15 | Concrete Slab (1) | S-10 | Pile Driving | Dist Ctrl[a] | A-1 ~ D-1 |
| S-17 | Concrete Walls (1) | S-15 | Concrete Slab (1) | G-FTS-TC[b] | A-1 ~ D-1 |
| S-20 | Concrete Slab (2) | S-15 | Concrete Slab (1) | Continuity | D-1 -- A-2 |
| | | S-17 | Concrete Walls (1) | G-FTS-TC[b] | A-2 ~ D-1 |
| S-22 | Concrete Walls (2) | S-20 | Concrete Slab (2) | G-FTS-TC[b] | A-2 ~ D-2 |
| | | S-17 | Concrete Walls (1) | Continuity | D-1 -- A-2 |
| S-25 | Concrete Slab (3) | S-20 | Concrete Slab (2) | Continuity | D-2 -- A-3 |
| | | S-22 | Concrete Walls (2) | G-FTS-TC[b] | A-3 ~ D-2 |
| S-27 | Concrete Walls (3) | S-25 | Concrete Slab (3) | G-FTS-TC[b] | A-3 ~ D-3 |
| | | S-22 | Concrete Walls (2) | Continuity | D-2 -- A-3 |
| S-30 | Concrete Slab (4) | S-25 | Concrete Slab (3) | Continuity | D-3 -- A-4 |
| | | S-27 | Concrete Walls (3) | G-FTS-TC[b] | A-4 ~ D-3 |
| S-32 | Concrete Walls (4) | S-30 | Concrete Slab (4) | G-FTS-TC[b] | A-4 ~ D-4 |
| | | S-27 | Concrete Walls (3) | Continuity | D-3 -- A-4 |
| S-35 | Roof (1) | S-30 | Concrete Slab (4) | Continuity | D-4 -- A-4 |
| | | S-32 | Concrete Walls (4) | G-FTS-TC[b] | After A-4 |
| S-36 | Roof (2) | S-35 | Roof (1) | Continuity | A-4 -- B-4 |
| | | S-32 | Concrete Walls (4) | G-FTS-TC[b] | After B-4 |
| S-37 | Roof (3) | S-36 | Roof (2) | Continuity | B-4 -- C-4 |
| | | S-32 | Concrete Walls (4) | G-FTS-TC[b] | After C-4 |
| S-38 | Roof (4) | S-37 | Roof (3) | Continuity | C-4 -- D-4 |
| | | S-32 | Concrete Walls (4) | G-FTS-TC[b] | After D-4 |
| I-05 | Rough-in | S-35 | Roof (1) | L-FTS-TC[b] | A-4 -- A-4 |
| I-10 | Partition | I-05 | Rough-in | G-FTS-TC[b] | A-1 ~ D-4 |
| I-15 | Paint Ceiling | I-10 | Partition | G-FTS-TC[b] | A-1 ~ D-4 |
| I-20 | Exterior Block Wall | I-15 | Paint Ceiling | G-FTS-TC[b] | A-1 ~ D-4 |
| I-25 | Plaster Interior Wall | I-20 | Exterior Block Wall | G-FTS-TC[b] | A-1 ~ D-4 |
| I-30 | Plaster Exterior Wall | I-20 | Exterior Block Wall | G-FTS-TC[b] | A-1 ~ D-4 |
| I-35 | Paint Interior Wall | I-30 | Plaster Exterior Wall | G-FTS-TC[b] | A-1 ~ D-4 |
| | | I-25 | Plaster Interior Wall | G-FTS-TC[b] | A-1 ~ D-4 |
| I-40 | Floor Tile | I-35 | Paint Interior Wall | G-FTS-TC[b] | A-1 ~ D-4 |
| I-45 | Bathroom Tile | I-35 | Paint Interior Wall | G-FTS-TC[b] | A-1 ~ D-4 |
| I-50 | Window Installation | I-40 | Floor Tile | G-FTS-TC[b] | A-1 ~ D-4 |
| I-55 | Electrical Wiring | I-45 | Bathroom Tile | G-FTS-TC[b] | A-1 ~ D-4 |
| | | I-40 | Floor Tile | G-FTS-TC[b] | A-1 ~ D-4 |
| I-60 | Bathroom Accessories | I-45 | Bathroom Tile | G-FTS-TC[b] | A-1 ~ D-4 |
| | | I-40 | Floor Tile | G-FTS-TC[b] | A-1 ~ D-4 |
| I-65 | Closet Installation | I-50 | Window Installation | G-FTS-TC[b] | A-1 ~ D-4 |
| | | I-55 | Electrical Wiring | G-FTS-TC[b] | A-1 ~ D-4 |
| | | I-60 | Bathroom Accessories | G-FTS-TC[b] | A-1 ~ D-4 |
| I-70 | Door Installation | I-65 | Closet Installation | G-FTS-TC[b] | A-1 ~ D-4 |
| I-75 | Cleaning | I-70 | Door Installation | G-FTS-TC[b] | A-1 ~ D-4 |

[a]Dist Ctrl: Distance Controlled (Lead Distance = 4 floors)
[b]G-FTS-TC: Global Finish-to-Start Time-Controlled
[c]L-FTS-TC: Local Finish-to-Start Time-Controlled

**Table 3.** Activity Control Relationships for Apartment Building Project

| Act. ID | Activity Description | Ctrl Pred ID | Controlling Predecessor Description | Control Relation | Controlled Unit |
|---|---|---|---|---|---|
| S-05 | Pile Fabrication | N/A | N/A | N/A | N/A |
| S-10 | Pile Driving | S-05 | Pile Fabrication | Dist Ctrl[a] | A-1, A-2, A-3, A-4, B-1, B-2, B-3, B-4, C-1, C-2, C-3, C-4, D-1 |
| S-15 | Concrete Slab (1) | S-10 | Pile Driving | Dist Ctrl[a] | D-1 |
| S-17 | Concrete Walls (1) | S-15 | Concrete Slab (1) | G-FTS-TC[b] | A-1, B-1, C-1, D-1 |
| S-20 | Concrete Slab (2) | S-15 | Concrete Slab (1) | Continuity | A-2 |
| S-22 | Concrete Walls (2) | S-17 | Concrete Walls (1) | Continuity | A-2 |
| | | S-20 | Concrete Slab (2) | G-FTS-TC[b] | A-2, B-2, C-2, D-2 |
| S-25 | Concrete Slab (3) | S-20 | Concrete Slab (2) | Continuity | A-3 |
| S-27 | Concrete Walls (3) | S-22 | Concrete Walls (2) | Continuity | A-3 |
| | | S-25 | Concrete Slab (3) | G-FTS-TC[b] | A-3, B-3, C-3, D-3 |
| S-30 | Concrete Slab (4) | S-25 | Concrete Slab (3) | Continuity | A-4 |
| S-32 | Concrete Walls (4) | S-27 | Concrete Walls (3) | Continuity | A-4 |
| | | S-30 | Concrete Slab (4) | G-FTS-TC[b] | A-4, B-4, C-4, D-4 |
| S-35 | Roof (1) | S-30 | Concrete Slab (4) | Continuity | AfterA-4 |
| S-36 | Roof (2) | S-35 | Roof (1) | Continuity | AfterB-4 |
| S-37 | Roof (3) | S-36 | Roof (2) | Continuity | AfterC-4 |
| S-38 | Roof (4) | S-37 | Roof (3) | Continuity | AfterD-4 |
| I-05 | Rough-in | S-35 | Roof (1) | L-FTS-TC[b] | A-4 |
| I-10 | Partition | I-05 | Rough-in | G-FTS-TC[b] | A-1, A-2, A-3, A-4, B-1, B-2, B-3, B-4, C-1, C-2, C-3, C-4, D-1, D-2, D-3, D-4 |
| I-15 | Paint Ceiling | I-10 | Partition | G-FTS-TC[b] | A-1, A-2, A-3, A-4, B-1, B-2, B-3, B-4, C-1, C-2, C-3, C-4, D-1, D-2, D-3, D-4 |
| I-20 | Exterior Block Wall | I-15 | Paint Ceiling | G-FTS-TC[b] | A-1, A-2, A-3, A-4, B-1, B-2, B-3, B-4, C-1, C-2, C-3, C-4, D-1, D-2, D-3, D-4 |
| I-25 | Plaster Interior Wall | I-20 | Exterior Block Wall | G-FTS-TC[b] | A-1, A-2, A-3, A-4, B-1, B-2, B-3, B-4, C-1, C-2, C-3, C-4, D-1, D-2, D-3, D-4 |
| I-30 | Plaster Exterior Wall | I-20 | Exterior Block Wall | G-FTS-TC[b] | A-1, A-2, A-3, A-4, B-1, B-2, B-3, B-4, C-1, C-2, C-3, C-4, D-1, D-2, D-3, D-4 |
| I-35 | Paint Interior Wall | I-25 | Plaster Interior Wall | G-FTS-TC[b] | A-1, A-2, A-3, A-4, B-1, B-2, B-3, B-4, C-1, C-2, C-3, C-4, D-1, D-2, D-3, D-4 |
| | | I-30 | Plaster Exterior Wall | G-FTS-TC[b] | A-1, A-2, A-3, A-4, B-1, B-2, B-3, B-4, C-1, C-2, C-3, C-4, D-1, D-2, D-3, D-4 |
| I-40 | Floor Tile | I-35 | Paint Interior Wall | G-FTS-TC[b] | A-1 |
| I-45 | Bathroom Tile | I-35 | Paint Interior Wall | G-FTS-TC[b] | A-1, A-2, A-3, A-4, B-1, B-2, B-3, B-4, C-1, C-2, C-3, C-4, D-1, D-2, D-3, D-4 |
| I-50 | Window Installation | I-40 | Floor Tile | G-FTS-TC[b] | D-4 |
| I-55 | Electrical Wiring | I-40 | Floor Tile | G-FTS-TC[b] | A-1, A-2, A-3, A-4, B-1, B-2, B-3, B-4, C-1, C-2, C-3, C-4, D-1, D-2, D-3, D-4 |
| I-60 | Bathroom Access. | I-40 | Floor Tile | G-FTS-TC[b] | A-1 |
| I-65 | Closet Installation | I-60 | Bathroom Access. | G-FTS-TC[b] | D-4 |
| I-70 | Door Installation | I-65 | Closet Installation | G-FTS-TC[b] | A-1, A-2, A-3, A-4, B-1, B-2, B-3, B-4, C-1, C-2, C-3, C-4, D-1, D-2, D-3, D-4 |
| I-75 | Cleaning | I-70 | Door Installation | G-FTS-TC[b] | A-1, A-2, A-3, A-4, B-1, B-2, B-3, B-4, C-1, C-2, C-3, C-4, D-1, D-2, D-3, D-4 |

[a]Dist Ctrl: Distance Controlled (Lead Distance = 4 floors)
[b]G-FTS-TC: Global Finish-to-Start Time-Controlled
[c]L-FTS-TC: Local Finish-to-Start Time-Controlled