HVLS: Horizontal and Vertical Logic Scheduling for Multistory Projects

By Walid Y. Thabet[1] and Yvan J. Beliveau,[2] Members, ASCE

Abstract: Multistory buildings are characterized by many repetitive units or typical floors, which involve identical activities repeated consecutively from one floor to the next. The scheduling of these activities is controlled by the logical sequence between activities executed on the same floor, horizontal logic, and between activities executed on different floors, vertical logic. Linear scheduling techniques have been developed for scheduling repetitive work in multistory projects. Although these techniques consider horizontal and vertical constraints in their scheduling approaches, they do not provide a formalized methodology for incorporating such constraints to generate schedules. This paper describes a structured procedure to incorporate horizontal and vertical constraints to schedule repetitive work. The procedure defines "critical floors" at which these constraints need to be checked to ensure that all other floors are satisfactory. The critical floors are grouped in a set of scheduling cases. For every activity, the procedure identifies applicable cases from the set of scheduling cases defined. To validate the constraints, the procedure only checks the critical floors defined in the selected scheduling cases and makes decisions to sequence activity.

INTRODUCTION

Multistory buildings mainly consist of a considerable number of typical floors with identical activities that are repeated consecutively from one floor to the next. These repetitive activities advance within the building in two directions: a horizontal direction through the floor, and a vertical direction from one floor to the next.

The sequencing of these activities is controlled by constraints between activities executed on the same floor, horizontal constraints, and constraints between activities executed on different floors, vertical constraints. To incorporate these constraints in the schedule, the scheduling process must ensure the constraints are satisfactory for all activities on every floor.

Since the early 1970s, several scheduling techniques have been developed to schedule activities with a repetitive nature. Each technique includes a multitude of variations, and most incorporated combinations of a network technique, a graphical technique, and/or an analytical technique. These combined approaches were named differently, but are generally referred to as "linear scheduling methods." Examples include: the line of balance (Arditi 1986; Johnston 1981; Carr and Meyer 1974), the vertical production method (VPM) (O'Brien 1975), the time space scheduling (Stradal and Cacha 1982), the linear scheduling method (LSM) (Chrazanwski 1986), the repetitive project model (RPM) (Reda 1990), and the time chainage charts (Mawdesley et al. 1990).

Linear scheduling techniques have been used for scheduling work with a repetitive nature, including multistory projects. Although these techniques allow for the utilization of horizontal and vertical constraints, they do not

[1] Asst. Prof., Dept. of Civ. Engrg., Union Coll., Schenectady, NY 12308.

[2] Assoc. Prof., Dept. of Civ. Engrg., Virginia Tech., Blacksburg, VA 24061.

Note. Discussion open until May 1, 1995. To extend the closing date one month, a written request must be filed with the ASCE Manager of Journals. The manuscript for this paper was submitted for review and possible publication on April 29, 1993. This paper is part of the ***Journal of Construction Engineering and Management***, Vol. 120, No. 4, December, 1994. ©ASCE, ISSN 0733-9364/94/0004-0875/$2.00 + $.25 per page. Paper No. 6086.

provide a formalized methodology for incorporating the constraints to generate schedules.

This paper describes a structured procedure to incorporate horizontal and vertical constraints to schedule repetitive work in multistory projects. The proposed procedure is entitled horizontal and vertical logic scheduling (HVLS). The HVLS procedure defines a set of scheduling cases to validate horizontal and vertical constraints for any activity. There are 10 scheduling cases defined for horizontal constraints and one for vertical constraints. Each scheduling case identifies a number of "critical floors." On each critical floor, horizontal and vertical constraints are checked to insure the constraints are satisfactory on all other floors.

To satisfy constraints for any activity, the HVLS procedure identifies the applicable scheduling cases and critical floors from the set of 11 cases defined. The procedure also includes scheduling decisions responsible for validating these constraints on the selected floors. The scheduling decisions consider three activity parameters: work continuity, progress rate or duration, and vertical workflow direction.

The HVLS procedure has been integrated with other resource-based and space-based scheduling procedures to develop SCaRC (Space Constrained and Resource Constrained) (Thabet 1992). SCaRC is a knowledge-based scheduling system for scheduling repetitive floors in multistory buildings. The system was developed at Virginia Tech as part of a broader research project and is currently operational. The SCaRC system is not discussed in this paper. However, the system is mentioned to show that the concepts developed and presented in this paper have been implemented.

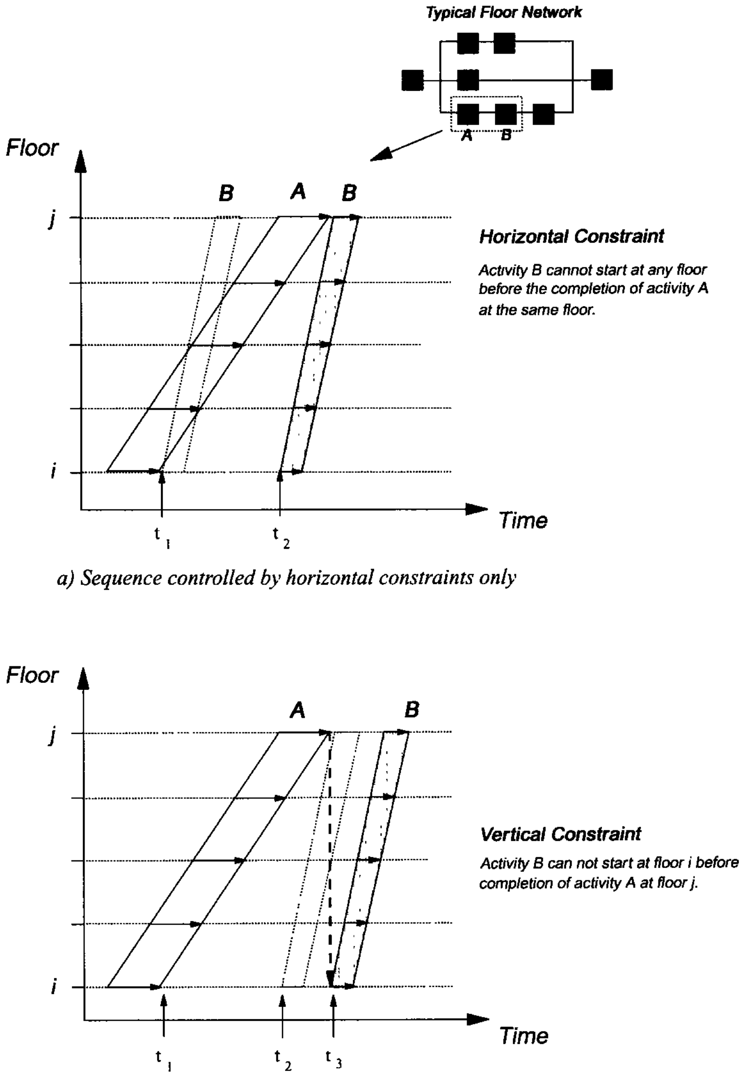
HORIZONTAL VERSUS VERTICAL LOGIC CONSTRAINTS

Two broad types of constraints control the logical sequence among activities in multistory projects—horizontal constraints, and vertical constraints. Horizontal logic constraints control the logical sequence between activities on the same floor. These constraints are defined by the precedence links among the activities on each floor, as established in a typical floor logic network. Based on such constraints, any activity cannot be scheduled unless all its preceding activities are scheduled.

Horizontal constraints are a result of many factors, such as technological dependencies among activities, weather constraints, imposed dates, safety constraints, and so on. An example of such constraints is the technological dependency imposed on the start of the floor slab by the completion of prior floor columns, girders, and beams.

As shown in Fig. 1(*a*), activity B cannot be scheduled to start on any floor until the preceding activity A has been scheduled on the same floor. This precedence dependency or horizontal logic is defined in the typical floor network. To satisfy the horizontal constraint imposed by activity A on B, the starting date of B at the first floor may be shifted from time t_1 to t_2, as shown. Another alternative (not shown) is to modify the production rate of activity B and/or A to satisfy the condition. These alternatives are discussed in the following sections.

The second type of constraints, vertical logic constraints, are based on logical links or dependencies among activities on different floors. These constraints are associated with the vertical construction of multistory projects. In this type of construction, the execution of some activities on some floors controls the execution of other activities on different floors. As a result, the start or completion dates of an activity on a particular floor may


Typical Floor Network
A
B
Floor
j
i
B
A
B
Horizontal Constraint
Activity B cannot start at any floor before the completion of activity A at the same floor.
Time
t_1
t_2
Floor
j
i
A
B
Vertical Constraint
Activity B can not start at floor i before completion of activity A at floor j.
Time
t_1
t_2
t_3

a) Sequence controlled by horizontal constraints only

b) Sequence controlled by horizontal and vertical Constraints

FIG. 1. Horizontal versus Vertical Constraints

become dependent on the controlling activity executed on a different floor. This establishes a vertical dependency between the two activities. This dependency defines a vertical constraint.

The general concept of vertical constraints may be illustrated by the example in Fig. 1(*b*). The condition defined in the figure assumes that activity B, on floor i cannot start at time t_2 before the completion of activity A on floor j at time t_3. This constitutes a vertical constraint between the two activities. As a result, the start of activity B needs to be shifted from time t_2 to time t_3.

An example of vertical constraints is the constructability constraint imposed on the external cladding work by the execution of the structural framework. In multistory buildings, external cladding requires the scaffolding to be erected along the building's external side. The key to the erection process is to suspend the scaffolds from outriggers tied to the building frame. Each outrigger would support the scaffolds for a certain number of floors, say 10 floors. That means that scaffolding on the first typical floor will not be ready for cladding work until the outrigger on the 10th floor is erected. This in turn dictates that the structural frame on the 10th floor must be completed to allow the erection of the outriggers. Similarly, scaffolding on the 11th floor cannot start until the erection of outriggers on the 20th floor is completed, and so on. This constructability constraint implies that a vertical lag or dependency of 10 floors should be maintained between the start of the cladding work and the completion of the structural frame and the construction of the outriggers.

Another example of vertical constraints results from restraints due to weather-tight requirements. For activity "Sheetrock" to proceed, the building has to be significantly enclosed. Horizontal constraints defined in the floor network insures that glazing and brickwork activities are completed before sheetrock installation can begin on the floor. This eliminates the problem of water entering horizontally through the building. The other problem is that of water leaking vertically through the building from the upper floors where glazing and waterproofing are not completed. As a result, the glazing and waterproofing work must be ahead of the sheetrock work by a number of floors to prevent the vertical flow of water from damaging the sheetrock. This time lag, expressed in floors, establishes a vertical constraint between the two activities.

ACTIVITY PARAMETERS

Several activity parameters are considered in the proposed HVLS procedure to define the scheduling cases and to formalize the scheduling decisions to sequence each activity. These parameters are work continuity, progress rate or duration, and vertical workflow direction. The parameters are discussed in the following sections.

Work Continuity

Because of the repetitive nature of the work in multistory buildings, activities should be continuously scheduled to allow crews to work without stopping, from one typical floor to the next. Maintaining continuity in the schedule reduces idle waiting intervals of equipment and manpower and minimizes extra effort associated with work interruptions (e.g., setup time, temporary storage of tools/ materials, and so on). Improving the continuity of the work also maximizes the learning curve effects. This results in considerable savings in time and cost.

To achieve maximum time and cost savings, the splitting of certain activities should not be allowed or should be minimized. This is considered essential for major trades such as structural framing, blockwork, external cladding, heating, ventilation, and air conditioning (HVAC) ductworks, and so on. Where continuity is not a significant issue, the splitting of the activity should be permitted.

The HVLS procedure takes into consideration work continuity for each activity. Activities are categorized into two groups or classes: continuous or intermittent. A continuous activity is scheduled with no splitting from the first typical floor to the last. To schedule the activity, horizontal and vertical constraints must be satisfactory on all floors, otherwise the activity is delayed. This may result in delaying succeeding activities and is considered a disadvantage. Another alternative to satisfy the constraints is to modify the duration of activity. This is discussed in the next subsection.

Intermittent activities are scheduled in floor segments. Each segment consists of one or more typical floors and is individually considered for scheduling. To schedule an activity segment, the horizontal and vertical constraints must be satisfactory only along the critical floors of that segment.

If the constraints are satisfactory and splitting is not necessary, the activity segment is not delayed beyond its initial starting date. This allows the scheduling of the segment continuously, with the previous segment(s). As a result, an intermittent activity may be scheduled continuously if splitting is not required, or may be split on one or more floors to satisfy a constraint condition.

The number of floors of each activity segment is based on a parameter defined by the user. The parameter defines the number of floors considered for scheduling at any one time. If the parameter is set equal to one, the activity is scheduled one floor at a time. This produces an optimum activity schedule but is time expensive and may result in a maximum number of splits. If the parameter is set equal to the total number of typical floors, the activity is scheduled continuously along all the floors, with no splitting.

Progress Rate or Duration

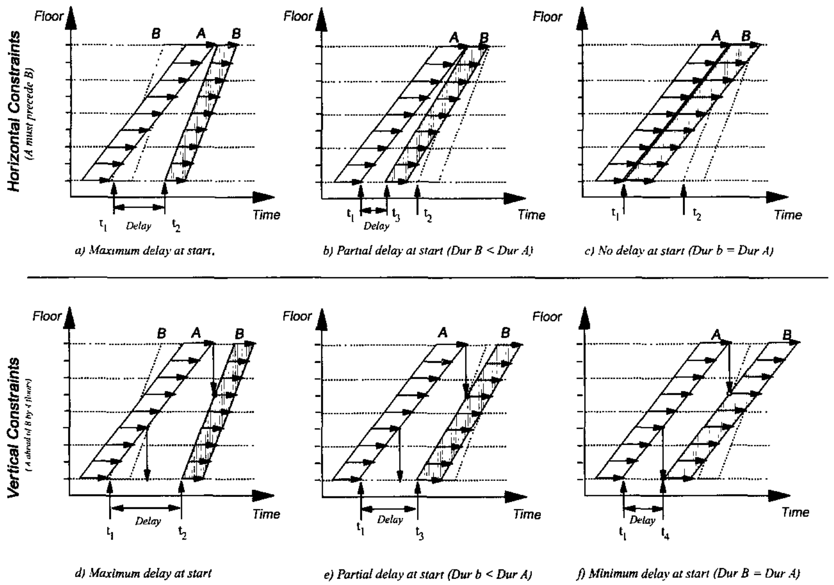
An activity progress rate or duration is dependent on the number of crews or resources assigned to the activity. Almost any activity can be performed with a wide range of assigned resources. These assigned resources determine the activity's duration along the floors. By manipulating the duration of the activity during scheduling, it is possible to reduce or eliminate time delays resulting from horizontal or vertical constraints.

The HVLS procedure utilizes two duration values for each activity—a normal duration value and a maximum duration value. These values are defined by the user during data input. During scheduling, the procedure selects the activity duration necessary to satisfy the horizontal and/or vertical constraints.

To account for horizontal constraints, the HVLS procedure selects the activity duration required to conform with the pace of the dominant activity. Also termed the controlling trade, a dominant activity is the activity that requires the longest execution period. It dominates the progress rates of all succeeding activities until an activity with a longer duration rate is encountered.

As shown in Fig. 2(*a*), activity B is eligible to start at time t_1 immediately after A is completed on the 1st floor. However, because B has a faster rate of progress than A (i.e., smaller duration), it would quickly overtake A on higher floors. This violates the horizontal constraint between the two activities defined in the floor network. To schedule B continuously, without modifying its normal duration, its starting date is shifted from time t_1 to t_2. This alternative is established by projecting the progress curve of B backwards, from the end of the dominant activity A on the last floor.

To start activity B earlier, the progress rate of B is balanced partially or completely with that of A, as illustrated in Fig. 2(*b* and *c*). If no maximum duration is specified, the activity duration cannot be balanced with the



a) Maximum delay at start.

b) Partial delay at start (Dur B < Dur A)

c) No delay at start (Dur b = Dur A)

d) Maximum delay at start

e) Partial delay at start (Dur b < Dur A)

f) Minimum delay at start (Dur B = Dur A)

FIG. 2. Varying Activity Duration to Satisfy Horizontal/Vertical Constraints

dominant activity. In this case the activity must be delayed, as shown in Fig. 2(*a*), or split if it is intermittent.

To account for vertical constraints, the same concept is applied to manipulate the duration of the activity. The scheduling procedure attempts to select a duration value necessary to minimize the delay resulting from the vertical lag imposed on the activity. This is illustrated in Fig. 2(*d*)–(*f*).

Vertical Workflow Direction

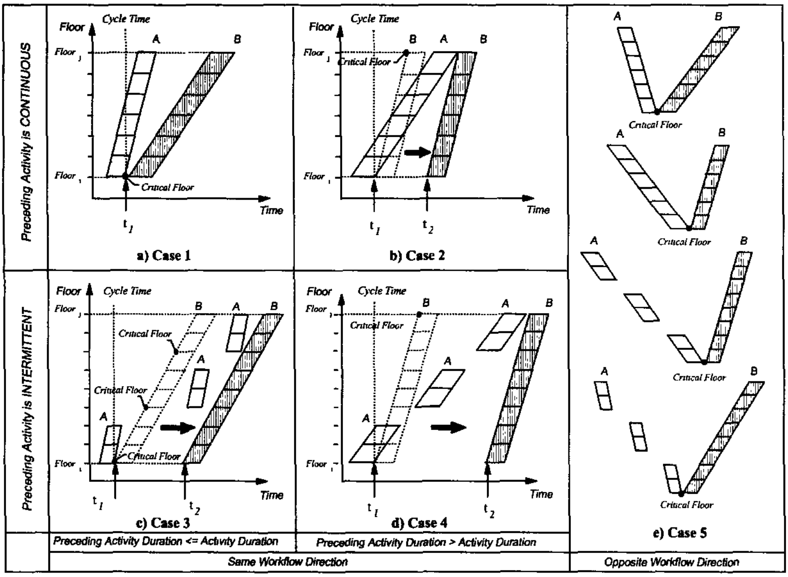
The repetitive characteristic of work in multistory buildings forms chains of activities. Each activity chain posses a spatial orientation or workflow in the vertical direction. The direction of flow is either upward or downward.

Most construction work is scheduled in an upward direction to follow the construction of the skeleton. Some other activities, however, are scheduled in a downward direction for reasons such as safety or to prevent damage of the installed work. The final finishing of external cladding, clearing, and cleaning may be typical examples of such types of activities.

SCHEDULING CASES FOR HORIZONTAL CONSTRAINTS

To satisfy a horizontal constraint for an activity, its starting dates must be equal to or greater than the completion dates of all preceding activities on every typical floor. However, activity starting dates need not be validated on every floor. These dates are checked at specific floors, termed critical floors. Validating the constraints on these critical floors insures that they are satisfactory for all typical floors.

Critical floors are identified for any activity based on its continuity, continuity of its preceding activities, duration of activity compared to its preceding activities, and workflow direction of activity compared to its preceding activities.


Preceding Activity is CONTINUOUS
Preceding Activity is INTERMITTENT
Floor
Cycle Time
Floor j
Floor i
A
B
Critical Floor
Time
t1
t2
Preceding Activity Duration <= Activity Duration
Preceding Activity Duration > Activity Duration
Same Workflow Direction
Opposite Workflow Direction

a) Case 1

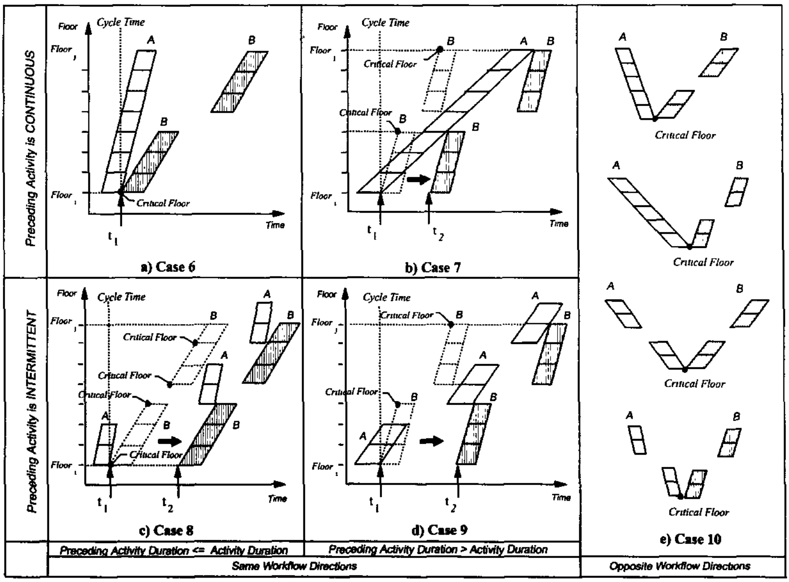
b) Case 2

c) Case 3

d) Case 4

e) Case 5

FIG. 3. Scheduling Cases for Continuous Activities (Cases 1–5)


Preceding Activity is CONTINUOUS
Preceding Activity is INTERMITTENT
Floor
Cycle Time
Floor j
Floor i
A
B
Critical Floor
Time
t1
t2
Preceding Activity Duration <= Activity Duration
Preceding Activity Duration > Activity Duration
Same Workflow Directions
Opposite Workflow Directions

a) Case 6

b) Case 7

c) Case 8

d) Case 9

e) Case 10

FIG. 4. Scheduling Cases for Intermittent Activities (Cases 6–10)

Critical floors are grouped in a set of scheduling cases. Cases 1–5 depicted in Fig. 3 are considered to validate horizontal constraints for continuous activities. Cases 6–10 depicted in Fig. 4 are considered to validate horizontal constraints for intermittent activities. The following is a description of each case:

Case 1

- Preceding activity is continuous
- Preceding activity duration is less than or equal to current activity duration
- Preceding activity vertical workflow direction is the same as current activity

This case has only one critical floor. With reference to Fig. 3(*a*), the starting date of the current activity B is only checked on its first typical floor (i).

Case 2

- Preceding activity is continuous
- Preceding activity duration is greater than current activity duration
- Preceding activity vertical workflow direction is the same as current activity

Similar to case 1, this case has only one critical floor. The starting date of B is only checked on its last typical floor (j). This is depicted in Fig. 3(*b*).

Case 3

- Preceding activity is intermittent
- Preceding activity duration is less than or equal to current activity duration
- Preceding activity vertical workflow direction is the same as current activity

The total number of critical floors for this case is equal to the number of scheduled segments of the preceding activity. As illustrated in Fig. 3(*c*), the starting date of B is checked at the beginning of each segment of activity A.

Case 4

- Preceding activity is intermittent
- Preceding activity duration is greater than current activity duration
- Preceding activity vertical workflow direction is the same as current activity

Similar to case 2, this scheduling case has one critical floor. As illustrated in Fig. 3(*d*), the starting date of B is only checked on its last typical floor (j).

Case 5

- Preceding activity vertical workflow direction is opposite to current activity

This scheduling case has only one critical floor defined. The preceding activity continuity class and duration has no influence in this case. As illustrated in Fig. 3(*e*), the starting date of B is only checked on its first floor.

Case 6

- Preceding activity is continuous
- Preceding activity duration is less than or equal to current activity duration
- Preceding activity vertical workflow direction is the same as current activity

This case has only one critical floor for all activity segments. With reference to Fig. 4(*a*), the starting date of activity B is only checked on the first floor of its first segment (i).

Case 7

- Preceding activity is continuous
- Preceding activity duration is greater than current activity duration
- Preceding activity vertical workflow direction is the same as current activity

This scheduling case has one critical floor for each activity segment. The starting date of B is checked on the last typical floor of each segment. This case is depicted in Fig. 3(*b*).

Case 8

- Preceding activity is intermittent
- Preceding activity duration is less than or equal to current activity duration
- Preceding activity vertical workflow direction is the same as current activity

This case has two critical floors for each activity segment. With reference to Fig. 4(*c*), the starting dates for each segment are checked on the first floor of the activity segment, and on an intermediate floor corresponding to the first floor of the preceding activity segment.

Case 9

- Preceding activity is intermittent
- Preceding activity duration is greater than current activity duration
- Preceding activity vertical workflow direction is the same as current activity

Similar to case 7, this scheduling case has one critical floor for each activity segment. With reference to Fig. 4(*d*), the starting date of each segment is checked on its last floor.

Case 10

- Preceding activity vertical workflow direction is opposite to current activity

Similar to scheduling case 5 for continuous activities, this scheduling case has one critical floor for all activity segments. The preceding activity continuity class and duration has no influence on this case. As illustrated in Fig. 4(*e*), activity B is only checked on the first floor of its first segment.

Validating Horizontal Constraints

To validate the horizontal constraints imposed on an activity, the HVLS procedure considers its preceding activities one at a time. For each preceding activity, the procedure selects the applicable scheduling case from cases 1–10. The case is selected based on the parameters of the activity compared to that of the preceding activity (i.e., continuity, rate of progress, and vertical workflow direction). On each critical floor defined in the selected case, the HVLS procedure compares the activity's preliminary starting date with the scheduled completion date of the preceding activity, and computes a time lag value. The lag value defines the time required to delay the activity segment beyond its preliminary starting date to satisfy the horizontal constraint imposed.

Two types of lag values are computed for each selected case. Fig. 5(*a*) depicts an example of lag values computed for case 8. The first type of lag values, lag-1 is computed at the critical floors, coinciding with the first floor of the activity segment being scheduled. The second type of lag values, lag-2 is computed at all other critical floors of the segment. The distinction between the two types of lags is made because of the different scheduling decisions considered to account for each type.

With reference to Fig. 5(*a*), lag-1 is computed from the preceding activity's scheduled completion date (t_2) at the critical floors, coinciding with the first floor of activity, as defined in case 8 (i.e., floor i), and the preliminary start date (t_1) of the activity segment on the critical floor

$$\text{lag-1} = t_2 - t_1 \tag{1}$$

Similarly, lag-2 is computed from the preceding activity's scheduled completion date (t_4) at other critical floors, defined in case 8 (floor j), and the preliminary starting date (t_3) of the activity segment on these floors

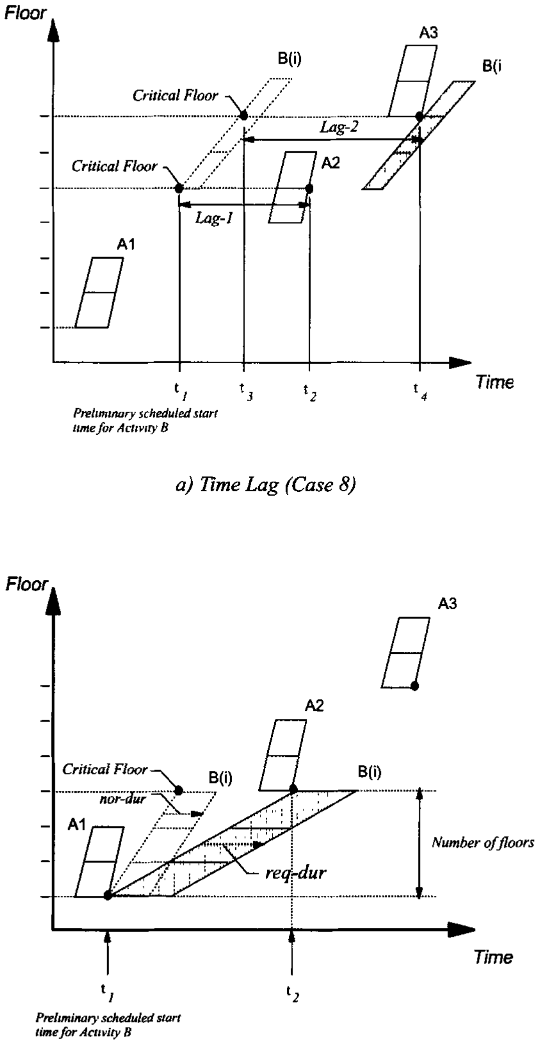
$$\text{lag-2} = t_4 - t_3 \tag{2}$$

Activity time lag values are computed for each preceding activity and considers the specific scheduling case that applies. From all time lag values computed, the maximum values are selected

$$\text{lag-1}_{\text{max}} = \max[(\text{lag-1})_1, (\text{lag-1})_2, \ldots, (\text{lag-1})_n] \tag{3}$$

$$\text{lag-2}_{\text{max}} = \max[(\text{lag-2})_1, (\text{lag-2})_2, \ldots, (\text{lag-2})_n] \tag{4}$$

Simultaneously, the HVLS procedure computes a new activity duration(s) required to satisfy the horizontal constraint imposed without delaying its preliminary starting time. Required duration values are computed based on all critical floors defined in the selected scheduling case, except on the first floor of the activity segment. An example is depicted in Fig. 5(*b*) for scheduling case 8. The required duration (req-dur) is computed by considering



a) Time Lag (Case 8)

b) Required Duration (Case 8)

FIG. 5. Time Lag and Required Duration Values for Case 8

the activity's preliminary starting date (t_1) on its first floor (floor_i), the preceding activity's scheduled completion date on critical floors identified (floor_j), and the total number of floors between floors i and j

$$\text{req-dur} = (t_2 - t_1)/\text{floor}(j) - \text{floor}(i) \qquad (5)$$

From all the duration values computed considering all preceding activities and corresponding critical cases, the maximum value is selected,

$$\text{req-dur}_{\max} = \max[(\text{req-dur})_1, (\text{req-dur})_2, \ldots, (\text{req-dur})_n] \qquad (6)$$

SCHEDULING CASE FOR VERTICAL CONSTRAINTS

Critical floors defined to validate vertical constraints are grouped in one scheduling case, depicted in Fig. 6. The critical floors are defined based on the end floors of the activity segment (B_i) considered for scheduling, the end floors of each activity segment (A_1 and A_2) imposing the constraint, and the imposed vertical constraint value expressed in floors (ver-lag). As depicted in Fig. 6, there are four types of critical floors defined

1. Critical floor 1: First floor (floor-i) of activity segment (B_i)
2. Critical floor 2: Last floor (floor-j) of activity segment (B_i)
3. Critical floor 3: Last floor (floor-y) of activity segment (A_1) imposing the constraint less the vertical lag (ver-lag).
4. Critical floor 4: First floor (floor-x) of activity segment (A_2) imposing the constraint less the vertical lag (ver-lag).

Validating Vertical Constraints

Similar to validating the horizontal constraint, the HVLS procedure considers the activities imposing vertical constraints on the activity one at a time. For each activity imposing a constraint, the procedure determines the critical floors to be checked and computes time lag values between the preliminary starting date of the activity and the scheduled completion dates of activity segments, forcing the vertical constraint. Again, two types of lag values are determined: lag-1 and lag-2.

Examples of lag values are depicted in Fig. 6. The values are computed as follows:

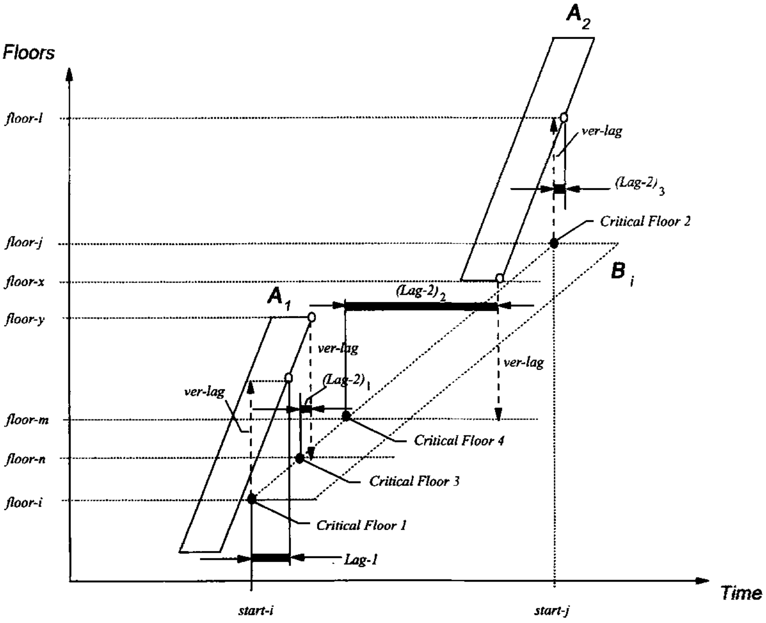


FIG. 6. Scheduling Case for Vertical Constraints

$$\begin{aligned}(\text{Lag-1}) &= \text{Scheduled finish date of } A_1 \text{ at (floor-i + ver-lag)} - \\ &\quad \text{Preliminary start date of } B_i \text{ at (floor-i)} \end{aligned} \qquad (7)$$

$$\begin{aligned}(\text{Lag-2})_1 &= \text{Scheduled finish date of } A_1 \text{ at (floor-y)} - \\ &\quad \text{Preliminary start date of } B_i \text{ at (floor-y} - \text{ver-lag)} \end{aligned} \qquad (8)$$

$$\begin{aligned}(\text{Lag-2})_2 &= \text{Scheduled finish date of } A_2 \text{ at (floor-x)} - \\ &\quad \text{Preliminary start date of } B_i \text{ at (floor-x} - \text{ver-lag)} \end{aligned} \qquad (9)$$

$$\begin{aligned}(\text{Lag-2})_3 &= \text{Scheduled finish date of } A_2 \text{ at (floor-j + ver-lag)} - \\ &\quad \text{Preliminary start date of } B_i \text{ at (floor-j)} \end{aligned} \qquad (10)$$

From all lag values computed considering all activities imposing vertical constraints, the maximum values (lag-1_{max}, and lag-2_{max}) are selected.

The procedure also computes a new activity duration required to satisfy the vertical constraint imposed without delaying the preliminary starting time of the activity segment. The required duration is computed for all critical floors except the starting floor of the activity segment.

For critical floor-2, the required duration is computed as follows:

$$\text{req-dur} = \frac{\begin{array}{c}[\text{Scheduled finish date of } A_2 \text{ at (floor-l)} - \\ \text{Preliminary start date of B(i) at its first floor (floor-i)}]\end{array}}{\text{Total number of floors between floor-i and floor-j}} \qquad (11)$$

For critical floor 3, the required duration is computed as follows:

$$\text{req-dur} = \frac{\begin{array}{c}[\text{Scheduled finish date of } A_1 \text{ at (floor-y)} - \\ \text{Preliminary start date of B(i) at its first floor (floor-i)}]\end{array}}{\text{Total number of floors between floor-i and floor-n}} \qquad (12)$$

For critical floor 4, the required duration is computed as follows:

$$\text{req-dur} = \frac{\begin{array}{c}[\text{Scheduled finish date of } A_2 \text{ at (floor-x)} - \\ \text{Preliminary start date of B(i) at its first floor (floor-i)}]\end{array}}{\text{Total number of floors between floor-i and floor-m}} \qquad (13)$$

From all required duration values computed considering all activities imposing the vertical constraint, the maximum value req-dur_{max} is selected.

SCHEDULING DECISIONS

If both lag values (lag-1_{max} and lag-2_{max}) computed for any type of constraint are less than or equal to zero, the constraint is satisfactory. However, if one or both of these values are found to be greater than zero, it indicates that the constraint is not satisfactory.

To eliminate the time lag and satisfy the constraint, one or both of the following actions are considered: (1) Delay starting the activity by a value less than or equal to lag-1_{max} or lag-2_{max}; and (2) decrease the activity pace by increasing its duration up to req-dur_{max} and such that

$$\text{req-dur}_{max} \Leftarrow \text{max-dur(limit defined by the user)} \qquad (14)$$

Scheduling decisions are explained in the following section. An example of each decision is depicted in Figs. 7 and 8 for scheduling case 8.

Decision 1

Condition

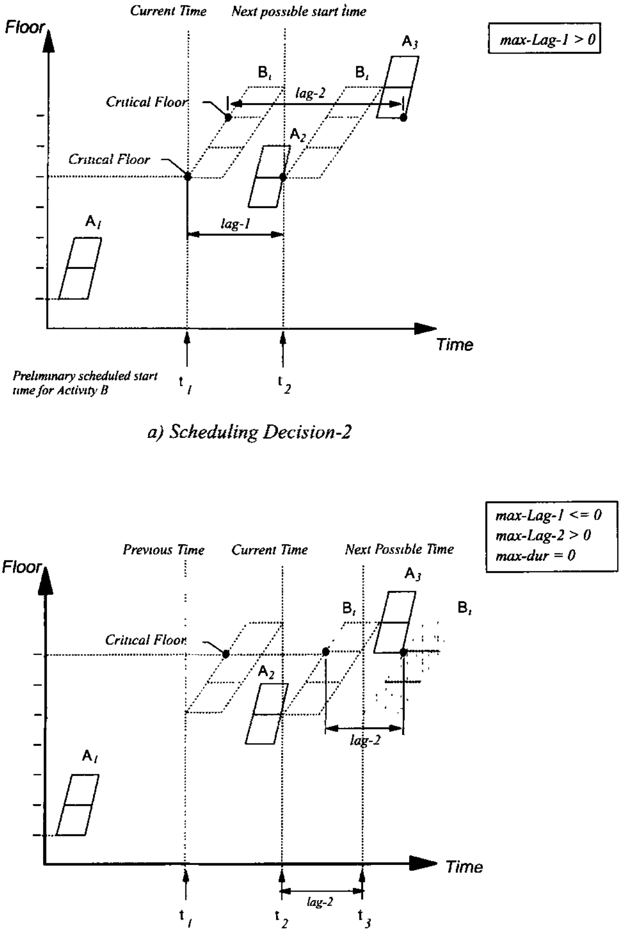
$$\text{lag-1}_{\max} \Leftarrow 0; \quad \text{and} \quad \text{lag-2}_{\max} \Leftarrow 0 \qquad (15a,b)$$

Decision: The constraint is satisfactory.

Decision 2

Condition

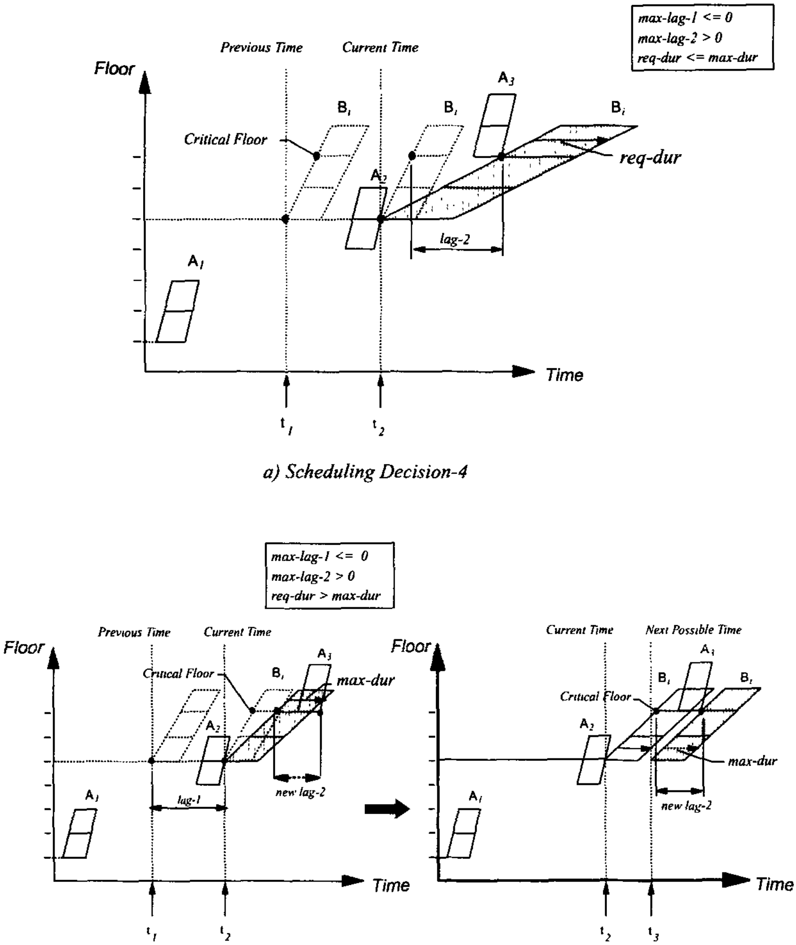
$$\text{lag-1}_{\max} > 0. \qquad (16a)$$



a) Scheduling Decision-2

b) Scheduling Decision-3

FIG. 7. Decisions to Satisfy Horizontal Constraints for Intermittent Activities



a) Scheduling Decision-4

b) Scheduling Decision-5

FIG. 8. Decisions to Satisfy Horizontal Constraints for Intermittent Activities

Decision: As depicted in Fig. 7(a), the starting date of the activity is delayed by lag-$1_{\max}$. This decision is made, regardless of the value of lag-$2_{\max}$, to satisfy the constraint at the activity segment's first floor. The new time to reconsider the activity for scheduling becomes

$$\text{new-time}(t_2) = \text{current-time}(t_1) + \text{lag-}1_{\max} \qquad (16b)$$

Decision 3

Condition

$$\text{lag-}1_{\max} \Leftarrow 0; \quad \text{and} \quad \text{lag-}2_{\max} > 0; \quad \text{and} \quad \text{max-dur} = 0 \qquad (17a\text{–}c)$$

Decision: The activity duration cannot be modified beyond its normal defined value. To satisfy the constraint, the starting date of the activity segment is delayed by lag-2_{max}. This is depicted in Fig. 7(b). The new time to consider the activity for scheduling becomes

$$\text{new-time}(t_2) = \text{current-time}(t_1) + \text{lag-}2_{max} \tag{18}$$

Decision 4
Condition

$$\text{lag-}1_{max} \Leftarrow 0; \quad \text{and} \quad \text{lag-}2_{max} > 0; \quad \text{and} \quad \text{req-dur}_{max} \Leftarrow \text{max-dur} \tag{19a–c}$$

Decision: As shown in Fig. 8(a), the activity segment duration is modified to the computed req-dur_{max} value to avoid delaying its starting time beyond t_2. This decision is considered because the maximum required duration (req-dur_{max}) computed by the procedure is less than or equal to the maximum duration limit defined by the user (max-dur). The modified duration is only applied to the activity segment and does not affect other segments.

Decision 5
Condition

$$\text{lag-}1_{max} \Leftarrow 0; \quad \text{and} \quad \text{lag-}2_{max} > 0; \quad \text{and} \quad \text{req-dur}_{max} > \text{max-dur} \tag{20a–c}$$

Decision: Increase the activity segment duration to the max-dur value defined by the user and compute a new maximum lag-2 value based on the new selected duration. To satisfy the constraint, the start of the activity segment is delayed by the new lag value $(\text{lag-}2_{max})_{new}$ computed. This is depicted in Fig. 8(b).

This decision is made because the maximum required duration computed by the procedure (req-dur_{max}) is greater than the max-dur limit defined by the user. Increasing the activity segment duration to the user-defined limit, however, will not satisfy the constraint. Using the new selected duration, a new maximum lag value is computed. To satisfy the constraint, the activity starting date is delayed by the new lag value. The activity is considered for scheduling at a new time equal to

$$\text{new time}(t_3) = \text{current time}(t_2) + \text{new lag-}2_{max} \tag{21}$$

INPUT DATA REQUIREMENTS

To incorporate horizontal constraints the HVLS procedure requires a logic network, for a typical floor, as input. The network must include activity numbers and their precedence relationships. To incorporate vertical constraints, the HVLS procedure requires the activity number imposing a vertical constraint, and the vertical constraint, expressed in floors.

The procedure also requires, as input, the normal and maximum activity durations, the activity workflow direction, and the minimum number of floors to be assigned to each activity segment. The final schedule of each activity may be based on a varying number of floors in each activity segment depending on the constraints imposed. However, the minimum number of floors in each segment is preserved. If a minimum value equal to one is defined, the activity schedule will be based on the earliest possible dates.

CONCLUSION

The scheduling of repetitive activities in multistory projects is controlled by the logical sequence defined between the activities of the same floor, termed horizontal logic, and between activities on different floors, termed vertical logic.

This paper described a formalized scheduling procedure to computerize knowledge about horizontal and vertical constraints for scheduling repetitive work in multistory projects. The proposed procedure is entitled horizontal and vertical constraints scheduling (HVLS). The HVLS procedure defined a set of scheduling cases to validate these constraints for any activity. There are 10 scheduling cases defined for horizontal constraints and one for vertical constraints. Each scheduling case identified one or more critical floors on which these constraints are checked. Satisfying the constraints on these floors insured that all other floors are satisfactory. The proposed HVLS procedure also included a set of scheduling decisions to sequence each activity. The scheduling decisions considered activity continuity, duration, and vertical workflow direction.

The HVLS procedure only applies to scheduling repetitive activities on typical floors, or linear portion, of multistory projects. The nonlinear portion cannot be scheduled using the proposed procedure and should be integrated if an overall scheduling model is intended for this type of construction. Another limitation of the proposed procedure is the consistency of workflow direction. The procedure schedules activities based on a constant workflow direction; either upward or downward, and does not provide for a varying of the workflow direction of crews. This may not be the case for some construction trades that may adopt inconsistent, but continuous, workflow direction to move between the different floors depending on the available work areas.

Although the HVLS procedure is developed for scheduling repetitive work multistory projects (i.e., vertical construction), it can be easily adopted to schedule horizontal linear projects (e.g., tunnels, pipelines, highways, and so on). However, since vertical constraints are not existent in this type of construction, it needs to be eliminated from the scheduling procedure of HVLS.

The scheduling procedure has been integrated with other resource-based and space-based scheduling procedures to develop a prototype computer program, SCaRC (Thabet 1992). SCaRC is a knowledge-based scheduling system for scheduling repetitive floors in multistory buildings. The system was developed at Virginia Tech as part of a broader research program and is currently operational.

APPENDIX. REFERENCES

Arditi, D., and Zeki, A. (1986). "Line-of-balance scheduling in pavement construction." *J. Constr. Engrg.*, ASCE, 112(3), 411–424.

Carr, R. I., and Meyer, W. L. (1974). "Planning construction of repetitive building units." *J. Constr. Engrg.*, ASCE, 100(3), 403–412.

Chrazanwski, E. N., and Johnston, D. (1986). "Application of linear scheduling." *J. Constr. Engrg.*, ASCE, 112(4), 476–491.

Johnston, D. W. (1981). "Linear scheduling method for highway construction." *J. Constr. Engrg.*, ASCE, 107(2), 247–261.

Mawdesley, M. J., Askew, W. H., Lees, J., Taylor, J., and Stevens, C. (1990). "Time change charts for scheduling linear projects." *Computing in Civ. Engrg.*, ASCE, 613–620.

O'Brien, J. J. (1975). "VPM scheduling for high rise buildings." *J. Constr. Engrg.*, ASCE, 101(4), 105–116.

Reda, R. M. (1990). "RPM: repetitive project modeling." *J. Constr. Engrg. and Mgmt.*, ASCE, 116(2), 316–330.

Stradal, O., and Cacha, J. (1982). "Time space scheduling method." *J. Constr. Engrg.*, ASCE, 108(3), 445–457.

Thabet, W. (1992). "A space-constrained resource-constrained scheduling system for multi-story buildings," PhD thesis, Dept. of Civ. Engrg., Virginia Tech, Blacksburg, Va.

Thabet, W., and Beliveau, Y. (1993). "Modeling work space to schedule repetitive floors in multi-story buildings." *J. Constr. Engrg. and Mgmt.*, ASCE.