Float Types in Construction Spatial Scheduling

Hisham Said\textsuperscript{1} and Gunnar Lucko\textsuperscript{2}

\textsuperscript{1} Assistant Professor, Department of Civil Engineering, 500 El Camino Real, Santa Clara University, Santa Clara, CA 95053, Email: hsaid@scu.edu; also adjunct lecturer, Faculty of Engineering, Cairo University, Egypt. \textbf{Corresponding Author}

\textsuperscript{2} Associate Professor and Director of Construction Engineering and Management Program, Department of Civil Engineering, Catholic University of America, 620 Michigan Avenue NE, Washington, DC 20064, Email: lucko@cua.edu

Abstract: Spatial scheduling has linked the one-dimensional temporal aspect with two geometric dimensions of area. To gain insights for decision-makers, it must be defined to what extent the activities in such spatio-temporal schedules should be deemed critical, i.e. in what constellations they may have little or no float; expanding the analogous – but lower-dimensional – concepts in network and linear scheduling. It must also be explored what possible float types can exist, under what circumstances they occur, and how they can be quantified. Geometric case distinctions are established for shift, rate, and combine float types, followed by expressing activities and their buffers in a spatial schedule as singularity functions, calculate temporal distances, and determine float values. Finally, a validation example demonstrates the new concepts of activity floats and space float contours.

Keywords: Space scheduling, floats, singularity functions, mathematical modeling

1. Introduction

Construction space is an important resource (Hegazy and Elbeltagi 1999) that is usually overlooked in planning construction operations, which result in congested sites, wasteful material relocation, and productivity decline. It is a challenging task to consider space availability in developing construction schedules due to the complexity of modeling the spatiotemporal relations between activities. On the other hand, modeling space availability and demand would support construction planners in producing shorter project schedules by facilitating concurrent execution of overlapping activities without resulting in spatial conflicts. Accordingly, construction spatial scheduling was introduced in previous research studies to utilize computational methodologies and decision support tools that facilitate flexible modeling of jobsite space in generating effective construction schedules. Such spatial scheduling techniques are very essential in congested construction sites with limited space availability and tight deadlines.

Executing, monitoring and controlling a construction schedule require the quantification of activities delay criticality in the form of their float times. The widely-applied Critical Path Method (CPM) provides the foundation to calculate different float types (total free, free float, independent float) that quantify activity criticality considering the impact of delays project finish time and succeeding activities. However, CPM and its float metrics don’t reflect the spatial dependencies between activities (Lucko 2007-a). Traditional float calculation methodologies cannot be applied to concurrent activities with overlapping workspaces due to the dynamic time difference between activities during their execution. Furthermore, additional float types should
be considered in spatial scheduling to account for the tighter coordination between activities and with other jobsite space demand items, like material staging and temporary facilities.

2. Literature Review

Construction time-space relation has attracted researchers to formulate typical types of spatiotemporal relations between activities and develop spatial conflicts identification processes. Akinci et al. (2002-a) developed mechanisms to automatically generate workspace needs of construction activities. Furthermore, Akinci et al. (2002-b) formulated a time-space conflict analysis to detect, categorize, and prioritize conflicts between construction workspaces. Guo (2002) developed a CAD-based tool that models space demands of construction activities, identify their conflicts considering project schedule, and utilize decision rules to resolve these conflicts. Dawood and Mallasi (2006) a space-time analysis approach models and quantifies space congestions by assigning workspace to activities and simulating their overlaps and conflicts. Chua et al. (2010) developed a four-dimensional computer aided methodology of modeling and analyzing site space congestions using a dynamic space interface index. Bansal (2011) developed a GIS-based (Geographic Information System) methodology that utilizes topology modeling and geospatial analysis to identify and resolve space-time conflicts. Finally, Su and Cai (2014) developed a lifecycle object-oriented approach to model construction workspaces, their requirements, and dynamic evolution.

Limited research studies were performed to incorporate activities spatial demand in the scheduling process instead of identifying spatial conflicts after the fact that schedule is created. Thabet and Berliveau (1997) developed a knowledge-based system for resources-constrained space-constrained (SCaRC) repetitive construction scheduling in multi-story buildings. The system recognizes the site space as a resource in addition to production crews during the generation of construction schedules. Winch and North (2006) developed a space scheduling decision support system that enables construction planners to identify and assign limited site space to activities. The tool provides automated capabilities of identifying available space from 2D drawings and interactive use tools to identify space overloading to be resolved manually by the user. Lucko et al. (2014) developed a spatial scheduling algorithm that utilizes singularity functions to model different spatial needs of activities and generates spatial schedules considering various types of activities relations.

Among the various studies that have broadened the dimensionality of schedules, very few appear to have focused on the phenomenon of spatially-constrained float. In linear (one-dimensional space) schedules, Harmelink (2001) distinguished ‘controlling’ and ‘non-controlling’ segments of activities. The latter of which were free ends of activity lines without an explicit link from or to a predecessor or successor, i.e. had some float. Yet only earlier starts or later finishes were described, i.e. ‘rate float’ as a lower productivity that was calculated geometrically. However, the study was based on an inherently incorrect definition of critical paths, as corrected in another study (Lucko 2007b). Lucko and Peña Orozco (2009) distinguished time and work buffers for completeness and demonstrated how rate float can be correctly calculated with singularity functions. Ammar (2003) took a narrower approach by allowing only constant productivities (not multiple segments of different slopes), distinguishing diverging and converging activities, and calculating rate float as a simple difference of the two activities’ productivities. But this study, and also Awwad and Ioannou (2007) were based on a rather limited model of linear schedules.
that considered links only between integer work units, which had been found to ignore potential changes that occur at fractional values (Lucko 2008). Kallantzis and Lambropoulos (2004) newly discussed minimum and maximum buffers between activities in linear schedules and proceeded by interrupting activities between work units to align their productivities, but did not extent their approach toward any possible impact on float by such interruptions.

Two main research gaps were identified when studying the relevant previous research. First, previous limited research on spatial scheduling produces incomplete activities scheduling information as the major focus was on activities start and finish times considering space availability. Complete scheduling information should also include activity floats to quantify its criticality and flexibility in accommodating potential site delays. Accordingly, there is a need to expand the applicability of spatial scheduling methodology by developing new activity float metrics that consider spatiotemporal relation between activities in two-dimensional space. Second, previous spatial scheduling models depend on a one-way communication of space availability and demand into the process of scheduling construction activities. However, spatial schedule information should also be communicated to site personnel to indicate the dynamic space availability and criticality for material staging and temporary facilities. Accordingly, there is a need to develop new metrics to quantify the criticality and availability of the construction space as a critical resource utilized in developing spatial schedules.

3. Objectives and Organization
The goal of this research is to derive float quantification metrics for spatial scheduling to provide a comprehensive analysis of the criticality of construction activities while considering space. Two objectives are established to fulfill this goal. **Research Objective 1** is to formulate float metrics of construction activities with a spatial scheduling model for different approaches of utilizing the slack time that is available between activities. **Research Objective 2** is to establish an approach to assess the criticality of jobsite space as a resource that is utilized by construction activities.

This paper is structured along the research methodology that was designed to achieve the goal and objectives, which included four main phases. The first phase incorporates a previously developed spatial schedule model (Lucko et al. 2014) with which so-called singularity functions are established that mathematically describe activity progress. The second phase derives three float metrics (shift, rate, and combined) that represent different options of quantifying available time slack with succeeding activities. The third phase creates a space float calculation algorithm to dynamically quantify the space criticality via space float contours. The fourth and last phase validates the performance of the float metrics via re-analysis of an example from the literature. The calculation of the activity floats and the space float algorithm are implemented using C++ programming to facilitate easy validation and illustration of their application.

4. Spatial Scheduling using Singularity Functions
4.1 Two-Dimensional Singularity Functions
Singularity functions are mathematically defined by applying their characteristic operator, which is indicated by pointed brackets ⟨ ⟩. Per Equation 1, to determine the dependent variable \( t(x) \), it performs a case distinction of whether the current value of the independent variable \( x \) is smaller than an activation cutoff \( a \) or not. In the former case, the operator yields zero, in the latter case,
the value is evaluated like a term within a standard polynomial. Its strength $s$ and exponent $n$ are the intensity and nature, respectively, of a quantitative behavior or part thereof that is modeled.

$$t(x) = s \cdot (x-a)^n = \begin{cases} 0 & \text{if } x < a \\ s \cdot (x-a)^n & \text{if } x \geq a \end{cases}$$  \hspace{1cm} \text{Eq. 1}$$

Once activated, the right-continuous term remains active until positive infinity. Deactivating it is achieved by simply subtracting said behavior at a later cutoff. A complete singularity function can contain one or multiple terms per Equation 1. Previous research has defined three principles, superposition, which means that active ranges can overlap; sorting, which means that terms are sorted by $a$ from low to high and $n$ from high to low for clarity; and simplification, which means that terms with identical $a$ and $n$ should be added for brevity (Lucko 2014). Singularity functions have provided a versatile 2D mathematical model, which has been applied to linear schedules (Lucko 2009), resource use (Lucko 2011a), and cash flows (Lucko 2011b) for entire projects. They have expressed one performance parameter — work quantity, resource need, or cost — over time. Their behavior can be constant over a range ($n=0$ to model resources), linear ($n=1$ to model cost), or take on any higher order, which allows customization to real-world phenomena.

An activity in a linear or repetitive schedule has two elements of quantitative information, the work that it performs and the duration that it takes to complete said work. Both are connected via the productivity, i.e. the ratio of work per time. Modeling it as a singularity function requires that the independent and dependent variables are assigned; since work quantity is typically given and time typically ought to be minimized in a project, work will be designated as $x$ and time as $t(x)$.

### 4.2 Three-Dimensional Singularity Functions

Project activities realistically occur within the geometric boundaries of a particular construction site, which can be discretized onto a grid system. To consider their spatial extent, the limitations of the two-dimensional model of work and time therefore only allows a limited depth of analysis. Recently, singularity functions thus have been broadened to the third dimension in a significant departure from their original definition (Macaulay 1919). Such 3D model is rooted in the idea to multiplicatively combine two projections of a 3D shape to express it mathematically (Lucko et al. 2014). Equations 2 and 3 show how the product of a function $t(x)$, which is a projection onto the $t-x$-plane, with a function $t(y)$, which projects onto the $t-y$-plane, can model 3D shapes. For example, two $n=0$ functions form a ‘block’, an $n=0$ function multiplied with an $n=1$ function forms a ‘ramp’, where the indices $S$ and $F$ denote start and finish cutoffs on the respective axis. Further shapes, e.g. a pyramid of two $n=1$ functions, are possible, but omitted here for brevity.

$$t_{block}(x,y) = s_0 \cdot \left( (x-a_{xS})^0 - (x-a_{xF})^0 \right) \cdot \left( (y-a_{yS})^0 - (y-a_{yF})^0 \right)$$  \hspace{1cm} \text{Eq. 2}$$

$$t_{ramp}(x,y) = s_1 \cdot \left( (x-a_{xS})^1 - (x-a_{xF})^1 \right) \cdot \left( (y-a_{yS})^0 - (y-a_{yF})^0 \right)$$  \hspace{1cm} \text{Eq. 3}$$

As determined by the respective $n$, $s_0$ is the height of the block, but $s_1$ is the slope of the ramp. Equation 3 models a ramp that grows parallel to the $x$-axis; switching axes could rotate this shape if needed. Note that it contains an extra term that steps downward at the far edge of the ramp.

Activities in a schedule that considers the spatial requirement and constraints can be modeled using the definitions of Equations 2 and 3 as follows: If the activity is stationary, i.e. occupies a particular area for a period of time, then it is expressed by the block shape of Equation 2. If the activity is directional, i.e. moves into one or several directions on the geometric axes, then each directional progress is expressed by the ramp shape of Equation 3. The geometric intervals $\{a_{xS}, a_{xF}\}$ and $\{a_{yS}, a_{yF}\}$ are determined based on factors including safety, the actual space needed at
the workface to complete the activity, and avoiding congestion that could lower productivity. Activities that exhibit variable progress rate can be discretized by modeling their segments as if these were activities in spatial scheduling, but preserving their strictly sequential relationship.

Buffers are inserted between activities to maintain a desired spacing e.g. for safety, technical, or resource-related reasons. As spatial buffers are assumed to be included in the aforementioned ranges already, time buffers remain to be modeled. They can be expressed as a time interval that is inserted between predecessor-successor pairs of activities. While they are typically assumed to be constant, in analogy with the 2D case (Lucko 2009), and can be expressed by adding a buffer per Equation 2 to the underlying activity, it is imaginable that they could be non-constant. In such case, they could be expressed by Equation 3, or even a higher-dimensional set of terms, which is beyond the scope of the present study. In general, activities and their buffers alternate in a schedule. For brevity, buffers will be considered in conjunction with the predecessor activity.

4.3 Space Scheduling Algorithm
Prior research by the first two authors (Lucko et al. 2014) has developed a scheduling algorithm that incorporates spatial constraints while minimizing the total project duration. Its steps include:
1) **Input:** List activities with productivity, buffer, work space, dependency, and direction option;
2) **Sort:** Determine list order by given priority heuristics (e.g. duration or cost) for tiebreaking;
3) **Start:** Recursively (in order of sorted list) and conservatively (project duration still is longer than possible) set start as “maximum of all predecessor finishes” (Lucko et al. 2014, p. 137);
4) **Options:** Write singularity functions for possible directions into which activity may progress;
5) **Potential Gain:** At spatial overlap corners, calculate time distance between pairs of activities, whose minimum is the potential gain that could be achieved when starting successor earlier;
6) **Conflict Checking:** Identify concurrent (i.e. not predecessor or successor) activities in work space, calculate corner distances, adjust gain if sign change indicates conflict from crossing;
7) **Update:** Final direction of best time gain, “subtract time gain to yield earliest start” (p. 137);
8) **Repeat:** Schedule next activity (if any) until project with minimum duration is configured.
Details of an example calculation and optimization have been presented by Lucko et al. (2014).

5. Activity Floats
Each activity can have one or all of three float types: Shift float, rate float, and combined spatio-temporal float. All types described in this paper refer to slack that is available for an activity to absorb delays in its scheduled dates (start or finish) or relax its scheduled production rate without delaying any of its successors. It thus bears resemblance to the free float of network schedules, however, differs in that it is defined within a 3D environment for spatially progressing activities, not merely as a time difference on a single axis (1D) of a network representation of the schedule. The three types indicate different approaches of utilizing this slack between scheduled activities.

To illustrate these float types, Figure 1 shows Example 1 that contains four activities, whose spatial schedule was generated by considering their dependencies and workspaces following the approach of Lucko et al. (2014), which has been described in the previous section. Dependency relations are that B and C depend on A, and D depends on B and C. Between these activities, four overlap zones exist: A-B, A-C, B-D, and C-D. **Overlap zones** in general are defined as sets of ranges on the x-axis and y-axis within which both activities fall. In other words, two activities share an overlap zone if they both occupy an area within the x-y-projection of a spatial schedule. Inspiration for overlaps zones is gleaned from traditional *interval relations* by Allen (1973), who examined how many different constellations can exist between two intervals of equal or unequal
duration. Yet the present research both extends and generalizes this classic study of concurrency; it extends in that overlap is considered within a 3D schedule analysis, and it generalizes in that intervals are not merely understood in a strictly temporal sense, but examined on the spatial axes.

Figure 1: Shift, Rate, and Combined Float in Spatial Scheduling (Example 1)

All activities in Example 1 can be written in the manner of Equation 3. Had any of them grown into the direction of the other axis (here $y$), all $x$- and $y$-labels would simply be switched. Inversely growing into the negative direction on an axis would be defined as a decreasing value into the positive direction on said axis, i.e. as an intercept after which a slope term is subtracted.

*Shift float* asks how long an activity can be delayed (i.e. shifted) within a spatial schedule without delaying any successor, while *maintaining* its planned productivity. As Figure 1 shows, B features shift float toward its successor D. Shift float $SF_{BD}$ equals the remaining available time difference between activities B and D at the $x$-coordinate $a_{xS_B}$ after having already subtracted the minimum required buffer between them. Other activities in Example 1 have zero shift float.

*Rate float* asks how much activity productivity (progress rate) within a spatial schedule can be reduced without delaying any of its successors. It is calculated by assuming that activities are initiated at their planned start, and is equivalent to an upward rotation of the activity slope. Note that due to charting time over distance, which is the inverse of the definition of productivity as distance divided by time, the slope in the diagram will *increase* if the productivity is *decreased*. A shown in Figure 1, Activities B and D have rate floats $RF_{BD}$ and $RF_{CD}$, respectively, as the differences between their minimum possible progress rates (without delaying successors) and
their planned rates. On the other hand, A has zero rate float due to already touching its successors B and C, which prevents reducing its productivity (i.e. increasing its slope) over its entire workspace. Activity D has also zero rate float, because as the last activity it has no successor and therefore defines the project finish time.

**Combined spatio-temporal float** is the time-space volume (prism) between the activities that can be used as an indicator of schedule criticality. It has less direct practical implications than the shift float or rate float, because it is expressed in time-space units. Yet it provides an integrated approach to quantify the criticality of activities when optimizing construction spatial scheduling. Depending on the shape of its prism, combined float can be utilized as either shift or rate float. For example, the combined float $CF_{BD}$ in Figure 1 can be converted into either shift float $SF_{BD}$ or rate float $RF_{BD}$. However, the shape of a combined float prism can limit or even prevent its use as shift or rate float. For example, the shape of the $CF_{CD}$ prism allows only its use as rate float, because the minimum time difference between C and D already is zero in their overlapping zone. No prism exists for the last activity D, as combined float is calculated between pairs of activities. If an activity overlaps or precedes multiple other activities, its combined float is calculated as the minimum of the pairwise floats with its successors or concurrent activities. For example, $CF_A$ equals the minimum of float values with its successors $CF_{AB}$ and $CF_{AC}$. A single index indicates that it is the feasible one among the possible float values with double index. Table 1 summarizes the float cases for all activities. Note that the proposed modeling approach of the combined float is limited by the assumption of continuous progress of the activities without interruptability. If interruptability were allowed, combined float could be utilized as a mixture of shift or rate float in any shape of its time-space volume. The following sections present more detail on each type.

### Table 1: Activity Input and Float of Example 1

<table>
<thead>
<tr>
<th>Activity</th>
<th>x-Range</th>
<th>y-Range</th>
<th>Shift Float</th>
<th>Rate Float</th>
<th>Combined Float</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$a_x S_A$</td>
<td>$a_x F_A$</td>
<td>$a_y S_A$</td>
<td>$a_y F_A$</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>$a_x S_B$</td>
<td>$a_x F_B$</td>
<td>$a_y S_B$</td>
<td>$a_y F_B$</td>
<td>$SF_{BD}$</td>
</tr>
<tr>
<td>C</td>
<td>$a_x S_C$</td>
<td>$a_x F_C$</td>
<td>$a_y S_C$</td>
<td>$a_y F_C$</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>$a_x S_D$</td>
<td>$a_x F_D$</td>
<td>$a_y S_D$</td>
<td>$a_y F_D$</td>
<td>0</td>
</tr>
</tbody>
</table>

#### 5.1 Shift Float

The calculation of the shift float depends on the type of the relationship between the activity and its successor. Figure 2 shows Example 2, where A progresses into the positive $x$-direction and B into the negative $y$-direction. They overlap between the coordinates $\{a_{x S_B}, a_{x F_A}\}$ and between $\{a_{y S_A}, a_{y F_A}\}$ on their respective axes, echoing the constellations between intervals (Allen 1983).
Technical-Driven Successor Activity

- Resource-Driven, Finish-Start Relation
  \[ SF_{AB} = S_B - F_A - \text{buff}_{AB} \]
- Following-Overlapping, Concurrent Relation
  \[ SF_{AB} = \min\{td1, td2, td3, td4\}_{AB} - \text{buff}_{AB} \]

Figure 2a: Example 2

Figure 2b: Shift Float between Succeeding Activities (Example 2)
Shift float $SF_{AB}$ between activities A and B is calculated by considering their singularity functions $t_A(x,y)$ and $t_B(x,y)$ per Equations 4 and 5, where the buffer $buff_{AB}$ between them is directly added to the intercept $S_A$; $S$ and $F$ are the respective start and finish times of the activity as provided by its index; and $R$ is the progress rate (the inverse of productivity) into the axis direction as provided by its index. Note that B grows into the negative $y$-direction, which is modeled by switching the variables within the pointed bracket operator of Equation 5. Another modeling possibility, albeit less intuitive, would have been to use $F_B$ as an intercept from which a negative progress rate descends to $S_B$.

$$z_A(x,y) = S_A \cdot \left[ x-a_{s,A} \right]^0 \left[ y-a_{f,A} \right]^0 + R_{y,A} \cdot \left[ y-a_{s,A} \right]^0 \left[ y-a_{f,A} \right]^0$$

$$z_B(x,y) = S_B \cdot \left[ x-a_{s,B} \right]^0 \left[ y-a_{f,B} \right]^0 + R_{y,B} \cdot \left[ x-a_{s,B} \right]^0 \left[ x-a_{f,B} \right]^0 - \left[ a_{f,B} - y \right]^0 \left[ a_{s,B} - y \right]^0$$

$$S_{AB} = S_B - F_A - buff_{AB} \quad \text{Eq. 6}$$

$$SF_{AB} = \min[\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4] - buff_{AB}$$

$$\Delta t_1 = t_B \left( a_{s,B}, a_{f,A} \right) - t_A \left( a_{s,B}, a_{f,A} \right)$$

$$\Delta t_2 = t_B \left( a_{s,B}, a_{s,A} \right) - t_A \left( a_{s,B}, a_{s,A} \right)$$

$$\Delta t_3 = t_B \left( a_{s,B}, a_{f,A} \right) - t_A \left( a_{f,B}, a_{f,A} \right)$$

$$\Delta t_4 = t_B \left( a_{s,B}, a_{s,A} \right) - t_A \left( a_{f,B}, a_{s,A} \right)$$

$$\Delta t_5 = t_B \left( a_{s,B}, a_{s,A} \right) - t_A \left( a_{f,B}, a_{s,A} \right)$$

**5.2 Rate Float**

Rate float is expressed not in terms of time units like shift float, but as a change in the progress rate due to the geometric constellation between the activity and its successor. Figure 3 shows it...
for Example 2, which was used in the previous section to explain the concept of shift float. Again the two cases that have been defined for the aforementioned shift float are distinguished:

- **Case I: Resource- or Technical-Driven Finish-Start Relation:** In analogy to Case I of shift float, rate float is the ratio of the duration from $F_A$ to $S_B$ after subtracting $buff_{AB}$, divided by the overlap distance ($L_{overlap}$) from a rotation axis at $S_A$ to the opposite edge of the overlap zone, which is the minimum of $F_A$ or $F_B$, i.e., here $\min\{a_{xF_A}, a_{xF_B}\} = a_{xF_A}$ per Equation 12.

$$RF_{AB} = \frac{S_B - F_A - buff_{AB}}{\min\{a_{xF_A}, a_{xF_B}\} - a_{xS_A}}$$  \hspace{1cm} \text{Eq. 12}

- **Case II: Technical-Driven Concurrent or Overlapping Relation:** In analogy to Case II of shift float, more rate float may be uncovered by allowing progress to be delayed beyond the start of a successor. This means that A can rotate upward into the time interval $\{S_B, F_B\}$ so that A can finish after B has started. As Figure 3 shows, rate float per Equation 13 is the ratio of rise, the minimum time difference (less $buff_{AB}$) between both singularity functions (index B - A) at the edge of the overlap zone, divided by run, the $x$-length of the overlap zone, here $a_{xF_A} - a_{xS_A}$. Equation 13 applies to any general configuration of possible progress directions of the activities. But it needs to be revised for the configuration of both activities progressing into the same direction, as shown in Figure 3. In this configuration the rate float depends on the maximum time difference between the respective singularity functions of the activities.

$$RF_{AB} = \frac{\min\left\{\min(a_{xF_A}, a_{xF_B}), a_{yS_A}\right\}_{B-A} \cdot r\left\{\min(a_{xF_A}, a_{xF_B}), a_{yF_A}\right\}_{B-A} - buff_{AB}}{\min(a_{xF_A}, a_{xF_B}) - a_{xS_A}}$$  \hspace{1cm} \text{Eq. 13}
Technical-Driven Successor Activity

- Resource-Driven, Finish-Start Relation
  \[ RF_{AB} = \frac{(S_B - F_A) - buff_{AB}}{L_{overlap}} \]

- Following-Overlapping, Concurrent Relation
  - Perpendicular Direction
    \[ RF_{AB} = \frac{(td_{min} - buff_{AB})}{L_{overlap}} \]
  - Parallel Direction
    \[ RF_{AB} = \frac{(td_{max} - buff_{AB})}{L_{overlap}} \]

Figure 3: Rate Float Calculation between Two Activities (Example 2)
5.3 Combined Spatiotemporal Float

Combined float (CF) is a spatiotemporal metric of slack to quantify the average acceptable delay of the predecessor over the overlapping space without delaying the successor. It can be used as an *ad hoc* quantity that integrates existing shift and rate floats between activities, which can be beneficial for the following future research: (1) Assessing the criticality of spatial schedules that can be minimized by increasing the space-time volumes between activity progress planes, which contradicts the planning goals of minimizing total duration and maintaining resource continuity; and (2) allowing progress interruptions of activities, which requires a more holistic quantification of space-time volumes between activities as compared to shift and rate float. Figure 4 illustrates calculating of the combined float ($CF_{AB}$) between activities A and B in the following two cases:

- **Case I: Resource- or Technical-Driven Finish-Start Relation:** This case of activity relations limits the delay of the predecessor to the start of its successor. Per Equation 14, the combined float depends on the average of the time differences ($\Delta s_1, \Delta s_2, \Delta s_3,$ and $\Delta s_4$) at the corners of their overlapping areas between the predecessor A and the start of its successor B. The time difference $\Delta s_1$ at corner point ($a_{S\_B}, a_{F\_A}$), for example, is the difference between the values of the singularity functions of A and the start time $S_B$ of B per Equation 15. Other time differences of the example in Figure 4 are calculated per Equations 16 to 18. Combined float is calculated by multiplying the area of the overlapping zone ($Area_{overlap}$) by the subtraction value of the average time difference and the buffer time ($buff_{AB}$) between A and B.

\[
CF_{AB} = \left[ \frac{\Delta s_1 + \Delta s_2 + \Delta s_3 + \Delta s_4}{4} - buff_{AB} \right] \times Area_{overlap} \tag{Eq. 14}
\]

\[
\Delta s_1 = S_B - t_A(a_{S\_B}, a_{F\_A}) \tag{Eq. 15}
\]

\[
\Delta s_2 = S_B - t_A(a_{F\_B}, a_{F\_A}) \tag{Eq. 16}
\]

\[
\Delta s_3 = S_B - t_A(a_{F\_B}, a_{S\_A}) \tag{Eq. 17}
\]

\[
\Delta s_4 = S_B - t_A(a_{S\_B}, a_{S\_A}) \tag{Eq. 18}
\]

- **Case II: Technical-Driven Concurrent or Overlapping Relation:** The combined float for concurrent or overlapping relations is calculated similar to the previous case of finish-start relations, but per Equation 19 considering the full time differences between progress planes of activities over their overlapping zone. Thus the time differences ($\Delta t_1, \Delta t_2, \Delta t_3,$ and $\Delta t_4$) are calculated as the difference between the values of the singularity functions of A and B per the previous Equations 8 to 11. This means that allowing concurrent or overlapping relations between activities results in larger combined float between activities ($\Delta t$ always exceeds $\Delta s$).

\[
CF_{AB} = \left[ \frac{\Delta t_1 + \Delta t_2 + \Delta t_3 + \Delta t_4}{4} - buff_{AB} \right] \times Area_{overlap} \tag{Eq. 19}
\]
Figure 4a: Combined Spatio-Temporal Float Calculation for Resource-Driven, Finish-Start Relation (Example 2)

Figure 4b: Combined Spatio-Temporal Float Calculation for Following-Overlapping, Concurrent Relation (Example 2)
6. Space Float

The proposed metric of space float measures the time period that is available at a location on the worksite to be utilized without delaying future construction activities from their scheduled early times. The space on the worksite can be utilized for different purposes, e.g. staging materials, routing equipment, or accommodating the execution of delayed activities. As such, space float \( PF \) is a multi-purpose metric that can be used for schedule planning, control and site layout.

The value of \( PF \) is a dynamic quantity at point \( \{x,y\} \) at a given time \( t \). It changes as the work progresses. Figure 5 shows the algorithm to calculate \( PF(x,y,t) \). It is initiated by the preparation Step ‘a’ that receives the variable inputs of \( x \), \( y \), and \( t \), sets the activity counter to zero, initializes the \( PF(x,y,t) \) variable to equal a very large value, and sets the Boolean variable \( ACT \) to ‘false’ to indicate that no activity was found that is scheduled at point \( \{x,y\} \). The algorithm loops over all activities by incrementing their counter variable \( i \) in Step ‘b’ and checking its value against the total number of activities \( N \) in Step ‘h’. For each activity, four substeps are performed. First, the singularity function of activity \( i \) is used in Step ‘c’ to determine its progress time \( t_i(x,y) \) at point \( \{x,y\} \). Second, the time to conflict \( TC_i(x,y,t) \) with activity \( i \) is calculated in Step ‘d’ between the progress time \( t_i(x,y) \) of activity \( i \) and the current analysis time \( t \). Third, Step ‘e’ evaluates the calculated time to conflict and excludes the activity from further consideration if its \( TC_i(x,y,t) \) is negative, which means that activity progress time is before the analysis time and does not control the space float at \( \{x,y\} \). Fourth, Steps ‘f’ and ‘g’ perform the following calculations if \( TC_i(x,y,t) \) of activity \( i \) is less than the current value of \( PF(x,y,t) \): Updating \( PF(x,y,t) \) with the newly found minimal time to conflict and changing the value of the Boolean variable \( ACT \) to ‘true’ to indicate that an activity is found to be executed at \( \{x,y\} \) after time \( t \). Steps ‘k’ and ‘m’ handle the situation of finding no activity for execution at \( \{x,y\} \); it is detected if \( ACT = ‘false’ \). In this case, \( PF(x,y,t) \) is calculated as the difference between the project duration \( PD \) and the current analysis time \( t \).
Figure 5: Space Float Calculation Algorithm for a Given Point (x,y)

Example 3 demonstrates how to use the space float algorithm to determine and query space float contours at any spatial point. Contours are dynamically evaluated by changing the analysis time $t$. This process is shown in Figure 6, whose three activities occur on a site that is discretized as a grid of 20 by 22 units spaced at 0.5 m. Table 2 lists their durations and coordinates. Activity C depends on both A and B with a technical-driven concurrent relation. Spatial scheduling was performed (Lucko et al. 2014) to generate the singularity functions of Equations 20 to 22. The scheduled directions, start time, and finish time are listed in Table 2 and illustrated in Figure 6.

1. **Start**
2. **Point Coordinates** \{(x,y)\}
3. **Current Time** = $t$
4. **Set activities counter** $i = 0$
5. **Space float** $PF(x,y,t) = +\infty$
6. **Following activity found** $ACT = False$
7. **i = i + 1**
8. **Obtain execution time of activity i at point** $(x,y)$, $t_i(x,y)$, using its singularity function
9. **Calculate Time to Conflict with activity i**
   $$TC_i(x,y,t) = t_i(x,y) - t$$
10. **$TC_i(x,y,t) > 0?$**
11. **$TC_i(x,y,t) < PF(x,y,t)?$**
12. **Update space float**
    $$PF(x,y,t) = TC_i(x,y,t)$$
    $$ACT = True$$
13. **$i > N?$**
14. **$ACT = False?$**
15. **Space float is based on project finish time**
    $$PF(x,y,t) = PD - t$$
16. **End**
\[
t_A(x, y) = \left[ (y-10)^0 - (y-22)^0 \right] \left( \frac{15}{16} \right) \left[ (x-0)^1 - (x-16)^1 - (16-0) \cdot (x-16)^0 \right]
\]
Eq. 20

\[
t_B(x, y) = \left[ (x-8)^0 - (x-20)^0 \right] \left( \frac{20}{10} \right) \left[ (y-0)^1 - (y-10)^1 - (10-0) \cdot (y-10)^0 \right]
\]
Eq. 21

\[
t_C(x, y) = 9 \cdot \left[ (x-0)^0 - (x-20)^0 \right] \left[ (y-0)^0 - (y-16)^0 \right] + \left[ (y-0)^0 - (y-16)^0 \right] \left( \frac{40}{20} \right) \left[ (x-0)^1 - (x-20)^1 - (20-0) \cdot (x-20)^0 \right]
\]
Eq. 22

<table>
<thead>
<tr>
<th>Activity</th>
<th>Duration [h]</th>
<th>axS [m]</th>
<th>axF [m]</th>
<th>ayS [m]</th>
<th>aFy [m]</th>
<th>Direction [-]</th>
<th>S [h]</th>
<th>F [h]</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>15</td>
<td>0</td>
<td>16</td>
<td>10</td>
<td>22</td>
<td>Pos. x</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>B</td>
<td>20</td>
<td>8</td>
<td>20</td>
<td>0</td>
<td>10</td>
<td>Pos. y</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>C</td>
<td>40</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>16</td>
<td>Pos. x</td>
<td>9</td>
<td>49</td>
</tr>
</tbody>
</table>

Table 2: Activity Input and Spatial Scheduling of Example 3

For illustration the algorithm is applied to each grid point to generate contours at two example times as Figure 7 \((t = 0)\) and Figure 8 \((t = 35)\) show. For example, space float at point P \((12,13)\) is calculated at time 0 as the difference between the minimum of the progress times of A and C \((11.25 \text{ and } 33.00)\), and the analysis time \(t = 0\) hours as 11.25 hours. The progress times of activities A and C at point \((12,13)\) are obtained by retrieving their singularity function values \(t_A(12,13)\) and \(t_B(12,13)\) using Equations 20 and 22, respectively. Note that B is not considered in the calculation, because point P does not reside within its workspace. For time \(t = 35\), the space float at point P is 14 hours, which is the difference of the project duration \(PD = 49\) and the analysis time \((t = 35)\). The project finish is considered here, because the progress times of the overlapping activities \((11.25 \text{ for A, } 33.00 \text{ for C})\) both occur before the analysis time \(t = 35\).
Figure 6: Spatial Schedule of Example 3 to illustrate the calculation of the Space Float

Figure 7: Space Float Contours of Example 3 at $t = 0$
6. Validation Example

The proposed spatial float metrics and algorithms are validated with a numerical example, which was introduced in a previous study (Lucko et al. 2014) with its spatial schedule. Table 3 lists the workspace coordinates, durations, predecessors, and crews of the 11 activities of the validation example. As Figure 9 shows, they comprise the curtain wall, electrical, and heating, ventilation, air conditioning (HVAC) work in a typical floor of an office building. Resource-related relations exist between activities of the same type, plus technical-driven relations. Figure 10 depicts the overall spatial schedule with a total duration of 85.17 hours (Lucko et al. 2014) that has been generated previously. Its detailed directions and start and finish times are listed in Table 3.

Table 3: Activities and Float of Validation Example (partially from Lucko et al. 2014 p. 139)

<table>
<thead>
<tr>
<th>No.</th>
<th>Act.</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>Dur.</th>
<th>Successor</th>
<th>Crew</th>
<th>Dir.</th>
<th>S</th>
<th>F</th>
<th>SF</th>
<th>RF</th>
<th>CF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>0</td>
<td>50</td>
<td>0.0</td>
<td>30.0</td>
<td>16</td>
<td>G, H, J, K Cr-1</td>
<td>Pos. y</td>
<td>33.33</td>
<td>49.33</td>
<td>0</td>
<td>0</td>
<td>346.7</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>0</td>
<td>50</td>
<td>0.0</td>
<td>30.0</td>
<td>24</td>
<td>-</td>
<td>Cr-2</td>
<td>Pos. y</td>
<td>61.17</td>
<td>85.17</td>
<td>0</td>
<td>0</td>
<td>18000.0</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>0</td>
<td>50</td>
<td>25.0</td>
<td>30.0</td>
<td>14</td>
<td>A</td>
<td>Cr-3</td>
<td>Neg. x</td>
<td>0.00</td>
<td>14.00</td>
<td>0</td>
<td>0</td>
<td>1750.0</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>45</td>
<td>50</td>
<td>5.0</td>
<td>25.0</td>
<td>6</td>
<td>A</td>
<td>Cr-3</td>
<td>Pos. y</td>
<td>28.00</td>
<td>34.00</td>
<td>0</td>
<td>0</td>
<td>300.0</td>
</tr>
<tr>
<td>5</td>
<td>E</td>
<td>0</td>
<td>50</td>
<td>0.0</td>
<td>5.0</td>
<td>14</td>
<td>A</td>
<td>Cr-3</td>
<td>Neg. x</td>
<td>14.00</td>
<td>28.00</td>
<td>0</td>
<td>0</td>
<td>1750.0</td>
</tr>
<tr>
<td>6</td>
<td>F</td>
<td>0</td>
<td>5</td>
<td>5.0</td>
<td>25.0</td>
<td>6</td>
<td>A</td>
<td>Cr-3</td>
<td>Pos. y</td>
<td>34.00</td>
<td>40.00</td>
<td>0</td>
<td>0.233</td>
<td>233.3</td>
</tr>
<tr>
<td>7</td>
<td>G</td>
<td>30</td>
<td>45</td>
<td>20.0</td>
<td>30.0</td>
<td>16</td>
<td>B</td>
<td>Cr-4</td>
<td>Pos. y</td>
<td>67.17</td>
<td>83.17</td>
<td>0</td>
<td>0</td>
<td>600.0</td>
</tr>
<tr>
<td>8</td>
<td>H</td>
<td>5</td>
<td>20</td>
<td>5.0</td>
<td>25.0</td>
<td>20</td>
<td>B</td>
<td>Cr-4</td>
<td>Pos. y</td>
<td>47.17</td>
<td>67.17</td>
<td>0</td>
<td>0</td>
<td>3000.0</td>
</tr>
<tr>
<td>9</td>
<td>I</td>
<td>20</td>
<td>30</td>
<td>3.0</td>
<td>8.0</td>
<td>10</td>
<td>A, J*, K* Cr-5</td>
<td>Stationary</td>
<td>0.00</td>
<td>10.00</td>
<td>9.6</td>
<td>0</td>
<td>220.0</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>J</td>
<td>7</td>
<td>20</td>
<td>12.5</td>
<td>17.5</td>
<td>8</td>
<td>G, H, B</td>
<td>Cr-5</td>
<td>Neg. x</td>
<td>44.67</td>
<td>52.67</td>
<td>0</td>
<td>0</td>
<td>260.0</td>
</tr>
<tr>
<td>11</td>
<td>K</td>
<td>30</td>
<td>43</td>
<td>12.5</td>
<td>17.5</td>
<td>8</td>
<td>K</td>
<td>Cr-5</td>
<td>Neg. x</td>
<td>52.67</td>
<td>60.67</td>
<td>8.5</td>
<td>0.654</td>
<td>942.5</td>
</tr>
</tbody>
</table>

* F-S, all others allow overlap with 2 work-hours buffer.
The activity floats of the generated spatial schedule are quantified via the new shift, rate, and combined float metrics. As this schedule produces the shortest possible duration for the given activities, only three activities have shift or rate float as listed in Table 3. First, activity F has no shift float, because of the buffer time between its start time and its successor A (see Figure 10). However, F has rate float of 0.233 h/m, which is the available rate difference 16/30 - 6/20 to A. This means that F can absorb delay events that slow its progress by 0.233 h/m – the difference between its planned rate (6 work-hours for its y-length 20 m = 0.3 h/m) and the slower rate of A (16 work-hours for its y-length 30 m = 0.533 h/m). Second, I has shift float of 9.6 hours, which is the time period until it conflicts with its earliest successor A. However, I has no rate float, because it is a stationary activity that occupies all of its workspace during its duration (i.e. there is no progress of its work over the workspace). Third, K is the only activity that has both shift float (8.5 hours) and rate float (0.654 h/m). This is due to scheduling its overlapping activity B based on the times of other activities (H and G), which creates time float for activity K to be utilized as either shift or rate float. Unlike shift and rate floats, all activities are found to have combined floats greater than zero. Here is not feasible to transform the combined float of most activities into either shift or rate floats due to the modeling assumption of continuous activities progress (no interruptions allowed) with a fixed progress rate. For example, C has 1,750 h·m² combined float with no shift or rate float due to its crew-driven F-S relation with I that starts immediately after finishing C ($t = 14$). Accordingly, C cannot have shift and rate float unless it is allowed to interrupt its progress with a higher rate that compensates for a delay. Future modeling research is planned to also allow variable progress rates of activities to enable using combined float as shift or rate float.

Figure 9: Workspace of Validation Example (Lucko et al. 2014 p. 139)
The space float algorithm is used to generate the contours of the worksite at different times in the project duration. Figures 11 and 12 for example depict the space float contours at the project start $t = 0$ and at $t = 68$. At the beginning, the largest available space float resides in the middle of the worksite, where A (flooring) for the whole floor is scheduled after all curtain wall activities (C, E, D, and F), as Figure 11 shows. On the other hand, no float is available over the workspace of stationary activity I that is scheduled between from 0 and 10 hours. Also, the top right corner is critical with minimal space float, where C is scheduled to start at $t = 0$. Similar useful insights about the jobsite space float at $t = 68$ can be obtained from its space float contour in Figure 12.
7. Contributions to the Body of Knowledge
This paper has presented the development of novel float metrics for spatial scheduling to expand its practical application and facilitate its seamless integration with other tasks within construction management, including time-based scheduling, resource utilization, and cash flow management. This research has made two fundamental contributions to the body of knowledge:

- The proposed activity shift and rate floats expand the capabilities of the previously developed model of spatial scheduling (Lucko et al. 2014) to assess the criticality of activities;
- The proposed space float is envisioned to enable easy integration between project scheduling and other management tasks, such as site layout, material logistics, and trades coordination.
8. Conclusions and Recommendations
New float metrics were presented in this research to expand the capabilities of construction spatial scheduling by assessing the criticality of both activities and space. Activity shift and rate floats were developed to assess the ability to adjust activity’s scheduled times and progress rates in case of delay events of reduced productivity. A combined float metric was developed and calculated between the activities as a general indicator of the schedule flexibility in absorbing delays and interruptions. In addition, a space float was developed to quantify the available time period to occupy a location in the jobsite without delaying any of the construction activities. Space float contours were proposed to dynamically visualize the critical locations of the jobsite over the progress of the project time. These spatial schedule floats were formulated using singularity functions and implemented using C++ programming language.

Future research work includes spatial schedule optimization, contour-based dynamic site layout planning, and complex float modeling. First, optimization models will be developed to analyze the tradeoff between minimizing the duration of spatial schedules and their criticality in the form of the combined floats between the activities. Second, new dynamic site layout planning models will be developed that utilize space float contours generated from spatial schedule to represent the jobsite dynamic space availability and identify optimal positions of the temporary facilities. Third, a generalized modeling of the total float from network schedules will be performed by examining the path of all successors of a given activity and determining their mutual overlap zones and space float contours.

References


