Linear Scheduling with Multiple Crews Based on Line-of-Balance and Productivity Scheduling Method with Singularity Functions

Highlights:

- Scheduling repetitive work requires models that consider multiple different crews;
- Line-of-Balance can express repetitive work, but lacks a mathematical formulation;
- Differences are explained to linear scheduling, which does not focus on crew use;
- Singularity functions are expanded multiplicatively to assign and utilize crews;
- Reanalysis of a schedule from literature validates proper functioning of approach.
Linear Scheduling with Multiple Crews Based on Line-of-Balance and Productivity Scheduling Method with Singularity Functions

1. Abstract

Scheduling repetitive work by multiple crews is a typical mission for schedulers, who must arrange the productivity, number, and size of crews and their lead or lag. Currently, the Linear Scheduling Method (LSM) and Line-of-Balance (LOB) can analyze repetitive work. These two related methods possess virtues, but lack a unified theory so that they could complement each other. Therefore this paper explores a new approach. Its contribution to the body of knowledge is threefold: First, dissimilarities in activity representation, start point, and velocity measurement are reviewed and their causes are explained graphically and mathematically. Second, multiple crews are expressed with singularity functions in a three-part model – crew assignment, crew utilization, and crew linear schedule. The model is automatically updated when parameters are modified for ‘what-if’ analyses. Third, LSM and LOB are united so that crews can be optimized seamlessly. This research advances theory on deploying multiple crews on construction projects.

Keywords: Linear schedule, line-of-balance, singularity functions, multiple crews, resources

2. Introduction

Repetitive activities are ubiquitous in construction projects like multi-story buildings, pipelines, and highways, which require continuous resource use (e.g. crews) across repeating units (Harris and Ioannou 1998). To calculate regular schedules, the traditional Critical Path Method (CPM) models the activity sequence as a network, equips it with activity durations, and
sequentially adds them to give the start and finish of each activity. Computerization performs this efficiently. Yet for repetitive projects with multiple tasks within each activity, the network inevitably grows, which makes visualization cumbersome (Lucko 2008). Real-world schedules with hundreds of activities and thousands of tasks within them will therefore typically span many sheets, which are difficult to parse for information. These networks feature one dimension (1D) of information, namely a multitude of activity starts and finishes over time that are listed as numerical values on each activity or task box. Crews who perform individual repetitive tasks are not shown in such networks. Thus “the CPM network technique for … repetitive projects has … major disadvantages. First, it requires a large number of activities [or better tasks] to represent the project… This large number … makes it extremely difficult to visualize the project… Second, the CPM technique does not guarantee maintaining the continuity of work” Reda (1990, p. 317). This network size issue when tasks and not merely activities are shown also appeared in Kallantzis et al. (2007) for a pipeline example. How it rather unclearly depicts the critical path compared to linear and repetitive scheduling approaches was demonstrated by Kim et al. (2014).

Operations research (OR) defines scheduling not just as calculating starts and finishes, but as “the allocation of resources over a period of time to perform a collection of jobs subject to known constraints” (Chan and Hu 2002, p. 165). It has provided approaches for assigning tasks (jobs) to productive resources (machines). They include, from least to most constrained, open shop, job shop, and flow shop (Pinedo 2008): Jobs that can be executed by different machines in any order (e.g. customers in a shopping mall); jobs that each have a unique order of different machines to visit (or skip), which flexibly creates many possible workflows (e.g. diners in a restaurant); and jobs that must be executed on different machines in a strict order or workflow (e.g. an assembly line), respectively. Often multiple machines of the same type exist in parallel.
Sequencing – determining the proper order of jobs for an object to be produced – is assumed as having been done previously and is a challenge in its own right (Chang and Angkasith 2001).

Similar to the machines of manufacturing, crews in construction are typically assumed as single-skilled. But unlike in manufacturing, construction crews move between tasks; i.e. jobs are considered to be stationary, not machines, as OR assumes. Moreover, while OR allocates given crews, construction scheduling also seeks to determine the optimum crew number (e.g. start multiple crews in a staggered manner) and size (to balance the productivity across activities). Furthermore, sequence constraints between construction tasks are typically strong, because they are dictated by the laws of physics (e.g. building from the ground up), technical requirements, or contractual terms, which somewhat resembles a flow shop. Besides such conceptual distinctions, construction strongly lags in how it schedules its resources in a manufacturing manner, with rare exceptions, e.g. a study of stadium columns (Lu 2009). Chan and Hu (2002, p. 166) thus deplore:

“[P]roject managers are very familiar with network-based scheduling techniques such as the critical path method (CPM)... On the other hand, production process scheduling models – for example, single machine sequencing, multimachine sequencing, flow shop, and job shop models – have been widely applied to manufacturing systems and logistics, but are relatively unknown in the construction industry. Yet, precast production is carried out under factory-like conditions. Formal production scheduling models of the kind used in manufacturing are not widely available in the precast industry.”

Graphical methods that rely on two-dimensional (2D) time-work coordinate systems emerged almost concurrently, but independently from networks: LOB even before network schedules, and later LSM. Even earlier publications by Adamiecki (1866-1933) in Polish were acknowledged by Marsh (1975). In 1896 he created his harmonogram to increase factory productivity. It comprised
vertical paper strips on a planning board. Each strip listed predecessor, activity, and successor in its header (Marsh 1975). It was divided into time units and carried a slider that was sized to the activity duration. Scheduling thus involved arranging the strips sequentially from left to right (as in a network) and moving their sliders down until all dependencies were fulfilled. The critical path is shown as the longest sequence, which is broken up across several strips. In use, tabbed parts of strips are marked with diagonal pencil lines to track their actual work progress. This feature makes harmonograms an intellectual forerunner of linear schedules, while Marsh (1976, p. 21) also viewed them as “work-flow network diagrams” as opposed to “Gantt [bar] charts [that] cannot be regarded as utilizing the work flow concept”. Adamiecki (1932) obtained U.S. Patent 1,860,763 for a drawing board for his charts that remained unmentioned in the literature.

In general, they can be distinguished by their origin and orientation: The original environment of LOB is the manufacturing industry (Office of Naval Material 1962). It morphed into a simpler form when it became introduced to construction (Lucko and Gattei 2016), where it was intended for use on multi-unit housing projects (Lumsden 1968) as a later section explains. Yet LSM was created within construction and was originally intended for geometrically linear projects like highways (Johnston 1981). It represents activities and tasks as single lines and can thus be deemed progress-oriented. Crew tasks within activities are enclosed by double lines in LOB, which makes it more crew-oriented, because each repetitive activity may be conducted by multiple crews. In this aspect LOB is more detailed than LSM, because it considers different crews who may perform their tasks in parallel, while LSM favors progress with one continuous productive resource. Both LOB and LSM have a different, yet complementary set of capabilities:

- The earlier largely graphical LSM has been underpinned with a model that uses singularity functions (Lucko 2008). It can handle time, cost, and single resources (Lucko 2011a, 2011b);
• This facilitates flexible modeling, analysis, and optimization, because each activity is one equation, a schedule is a group of activities that are additive, and their variables for starts and productivities allow expressing not just one, but all possible versions of a particular schedule;
• A mathematical model enables computerization toward automated analysis and optimization;
• LSM does not explicitly address crew counts and sizes as they occur in a repetitive process;
• On the other hand, LOB fosters a crew focus that is of particular importance for construction schedules, whose repetitive tasks are typically performed by various skilled crafts and trades;
• Some point-wise formularization (Al-Sarraj 1990) notwithstanding, LOB remains a graphical method that requires manual steps instead of applying mathematical equations as part of an algorithm, which hampers analysis and optimization that computerization could automate.
Crew modeling of LOB has merit, but cannot yet be handled by LSM mathematics. Put together, an even stronger analytical method could be gained, which would finally unify LOB and LSM.

3. Literature Review
Repetitive projects can be categorized as typical and atypical; for the former, the productivity of tasks in an activity is constant, whereas for the latter, segments can possess different production rates “due to variability of conditions and volumes” (Selinger 1980, p. 196). Units of repetitive work may or may not be identical, because the work quantity, construction methods, or types of equipment often vary for different units (Huang and Sun 2006). For planning purposes, identical repetitive units may still be modeled to derive an average estimate for atypical repetitive units in a project. Thus this paper will focus on typical repetitive projects, whereas the atypical case can be realized by modeling parameters as a function of time or of the unit (not assuming constant as in this paper), which will be explored in future research.
3.1 Original and Current Definitions of LSM and LOB

In LSM, “the repetitive activities are plotted as lines of constant or varying slopes on two axes, distance versus time” (Chrzanowski and Johnston 1986, p. 476), which allows managers to “monitor progress, suggest modifications, implement them, observe the resultant effects, and readily understand them without the aid of a computer or any special training” (Chrzanowski and Johnston 1986, p. 478). Historically, related approaches and variations featured names such as flowline (Slipchenko 1966), time versus distance diagram (Gorman 1972), and progress schedule (Peer 1974), which “typically model a linear growth of activities or segments thereof that progress within a two dimensional coordinate system of time and work” (Lucko and Su 2014, p. 3366). Features that LSM charts explicitly visualize are starts and finishes, duration, productivity, float (Lucko and Peña Orozco 2009), and criticality (Lucko 2008; Harmelink and Rowings 1998; Harris and Ioannou 1998), which aids in tracking and improving project performance. Being deterministic, it attempted to reflect probabilistic effects in “the time buffer that exists between activities … [which] is an acknowledgment that the timing of the occurrence of activities cannot be scheduled with pinpoint accuracy” (Hinze 2012, p. 203). Adding information on cost and resources allowed expansion toward cash flows (Su and Lucko 2015b; Lucko 2011a), bidding strategies (Su and Lucko 2015a), and resource use (Lucko 2011b, Mattila and Abraham 1998). Khisty (1970) defined LOB as “a management tool which assembles, selects, interprets, and presents in graphic form the essential factors involved in a production process, against a background of time”. According to Yang and Ioannou (2004), it was first developed by the Goodyear Company in 1940s, and intensively used by the U.S. Navy (Office of Naval Material 1962). Its original purpose was line balancing in manufacturing, which “is the process through which you evenly distribute the work elements within a value stream in order to
meet takt time… it balances workloads so that no one is doing too little or too much” (Tapping et al. 2002, p. 57). ‘balance’ means that the work quantity of a predecessor “exactly fulfills the successor’s demand in the assembly tree without accumulating any excess inventory to sustain a balanced production, no more, no less” (Su and Lucko 2015c, p. 1321). However, the current understanding of LOB in the construction management domain is “a variation of linear scheduling methods that allows the balancing of operations such that each activity is continuously performed” (Arditi et al. 2002, p. 545). In other words, it “determine[s] a balanced mix of resources and synchronize their work such that they fully employed” (Ammar 2013, p. 44), which “considers the information of how many units must be completed on any day to achieve the programmed delivery of units” (Damci et al. 2013, p. 681). According to previous studies (e.g. Dolabi et al. 2014; Ammar 2013; Hegazy 2001), the current construction LOB can be summarized as three steps: (1) calculate the total number of repetitive units for each activity; (2) estimate crew size, number of crews, and productivity per crew to gain “a natural rhythm for each activity (e.g., number of units / day)” (Arditi and Albulak 1986, p. 412); (3) draw double lines of an LOB quantity diagram that mark the start and finish event of individual activities. Between the double lines, crews may work sequentially or concurrently on repetitive units. Figure 1b shows this simpler approach with detailed substeps side-by-side to the original LOB of Figure 1a. It appears that its characteristic sloped double lines may have been derived from a combination of a production plan (which contains unit durations) and an objective chart (which contains a progress slope). This explanation is clear in the literature: Networks “with repeating units of work have a ladder-like appearance, where each rung is a subnetwork that consists of the activities and precedence links for one unit” (Harris and Ioannou 1998, p. 270). It is further supported by the original report (Lumsden 1968, p. 7) that explicitly lists a “ladder construction”.

Figure 1b shows this simpler approach with detailed substeps side-by-side to the original LOB of Figure 1a. It appears that its characteristic sloped double lines may have been derived from a combination of a production plan (which contains unit durations) and an objective chart (which contains a progress slope). This explanation is clear in the literature: Networks “with repeating units of work have a ladder-like appearance, where each rung is a subnetwork that consists of the activities and precedence links for one unit” (Harris and Ioannou 1998, p. 270). It is further supported by the original report (Lumsden 1968, p. 7) that explicitly lists a “ladder construction”.

Figure 1b shows this simpler approach with detailed substeps side-by-side to the original LOB of Figure 1a. It appears that its characteristic sloped double lines may have been derived from a combination of a production plan (which contains unit durations) and an objective chart (which contains a progress slope). This explanation is clear in the literature: Networks “with repeating units of work have a ladder-like appearance, where each rung is a subnetwork that consists of the activities and precedence links for one unit” (Harris and Ioannou 1998, p. 270). It is further supported by the original report (Lumsden 1968, p. 7) that explicitly lists a “ladder construction”.

Figure 1b shows this simpler approach with detailed substeps side-by-side to the original LOB of Figure 1a. It appears that its characteristic sloped double lines may have been derived from a combination of a production plan (which contains unit durations) and an objective chart (which contains a progress slope). This explanation is clear in the literature: Networks “with repeating units of work have a ladder-like appearance, where each rung is a subnetwork that consists of the activities and precedence links for one unit” (Harris and Ioannou 1998, p. 270). It is further supported by the original report (Lumsden 1968, p. 7) that explicitly lists a “ladder construction”.

Figure 1b shows this simpler approach with detailed substeps side-by-side to the original LOB of Figure 1a. It appears that its characteristic sloped double lines may have been derived from a combination of a production plan (which contains unit durations) and an objective chart (which contains a progress slope). This explanation is clear in the literature: Networks “with repeating units of work have a ladder-like appearance, where each rung is a subnetwork that consists of the activities and precedence links for one unit” (Harris and Ioannou 1998, p. 270). It is further supported by the original report (Lumsden 1968, p. 7) that explicitly lists a “ladder construction”.
In other words, construction LOB contains tasks nested within activities. Previous work has advocated that original LOB has broader capabilities than its current modified and simplified use in construction (Su and Lucko 2015c). To fulfill its analytical potential, this research therefore explores Multiple Crew Linear Scheduling (MCLS).

**Figure 1: Differences between Line of Balance Approaches**

(a) Original in Manufacturing

(b) Current in Construction

**Figure 1: Differences between Line of Balance Approaches**

In other words, construction LOB contains tasks nested within activities. Previous work has advocated that original LOB has broader capabilities than its current modified and simplified use in construction (Su and Lucko 2015c). To fulfill its analytical potential, this research therefore explores Multiple Crew Linear Scheduling (MCLS).
3.2 Advantages and Disadvantages of LSM and LOB

Per Table 1, LSM and LOB have shared and individual advantages and disadvantages. Together, the virtues of these methods demonstrate the potential for a conceptual integration. First, both “are essentially alike in that they schedule the work in the project by plotting the progress of repeating activities against time” (Harris and Ioannou 1998, p. 269). A common advantage is that both of them have graphical representations that exceed network schedules in information content, because they visualize work as well as time. Second, in terms of formularization, “LOB is primarily a graphical technique that lacks the analytical qualities of CPM scheduling” (Ammar 2013, p. 44). But LSM has been transformed into a completely mathematical model called the productivity scheduling method (PSM), which employed singularity functions (Lucko 2008). Inter-activity relations, such as criticality, float, and buffers, can all be modeled mathematically, but remain essentially unrealized for LOB. Third, a major challenge continues to exists in that “multiple crew usage cannot be modeled using line representation” with LSM (Ammar 2013, p. 45), which is at least feasible with LOB. Yet again the lack of inter-activity relations of LOB impedes its application to complex and dynamic schedules. Fourth, resource utilization has been modeled for LSM with modified singularity functions that measure resources (Lucko 2011), whereas LOB still needs such extension of its scheduling model (Damci et al. 2013). While LOB offers a geometric representation of crews that can be shaded or colored to distinguish them (El-Rayes and Moselhi 1998), an explicit crew assignment remains unexplored for LSM and LOB.
Table 1: Comparison of LSM and LOB

<table>
<thead>
<tr>
<th>Virtue</th>
<th>LSM</th>
<th>LOB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphical</td>
<td>Yes.</td>
<td>Yes.</td>
</tr>
<tr>
<td>Formularization</td>
<td>Yes, PSM by Lucko (2008) provided equations with singularity functions and a scheduling algorithm to analyze criticality. Float types were explored (Lucko and Peña Orozco 2009).</td>
<td>No, scheduling approach requires manual steps for given values of inputs. Must be extended to realize criticality analysis (Ammar 2013).</td>
</tr>
<tr>
<td>Available for multiple crews</td>
<td>No, focused on continuity of sloping activity line for single crew, must be extended to realize multiple crews.</td>
<td>Yes.</td>
</tr>
<tr>
<td>Resource Utilization</td>
<td>Yes, changing values in model using singularity functions (Lucko 2011) will modify entire resource profile.</td>
<td>No, is not integrated with scheduling, currently requires separate analysis.</td>
</tr>
<tr>
<td>Crew assignment</td>
<td>No.</td>
<td>No, only graphical representation by label or color of overlapped lines.</td>
</tr>
</tbody>
</table>

3.3 Research Objectives

Therefore, three Research Objectives are set to realize the goal of enhancing LOB and LSM:

1. Align the geometric rules of current LOB and LSM and find the reason for their differences;
2. Express the work of multiple crews within and between activities with singularity functions;
3. Unify linear scheduling with line of balance concepts and validate such LOB-PSM approach.

3.4 Geometrical Difference between LSM and LOB

According a comparison of LSM versus LOB, numerous differences exist (Su and Lucko 2015c).

First, LSM resembles activity-on-node network scheduling (AON), but LOB shows some similarities with activity-on-arrow (AOA). Linear and repetitive schedules can always be reduced to a network form, but at a loss of informational content from work and time (two dimensions of information) to just time (one dimension of information). Support for a connection of LSM and LOB with AON and AOA is found in the literature: Harris and Ioannou (1998, p. 270) in their paper on repetitive scheduling, which is a variation of linear scheduling as
introduced by Johnston (1981), made a direct comparison with AON; its “CPM diagrams show all of the linkages between similar activities in successive units, the number of links and nodes will likely be large and the network will appear unnecessarily complicated.” Similarly explicit is the LOB-AOA connection. The original LOB report began with a detailed description of AOA, whose circular start and finish marker “does not consume time or resources, it merely represents a point in time” (Lumsden 1968, p. 5). Second, start points differ in the time-work coordinate systems of LSM versus LOB: The profile starts from 0 in LSM but 1 in LOB. This is explained by the unit of the work axis; whereas LSM features a continuous work quantity, which measures from the origin (0), LOB only lists discrete completed work units, which counts in integers from the first completion (1), as Lucko and Gattei (2016) have pointed out, which is caused by using different counting conventions – start-of-period versus end-of-period – as Lucko (2007) had extracted for CPM. This leads to the third difference, the meaning of the slope to express the progress speed. In LSM it is the productivity of an activity as the ratio of work divided by time. This is typically assumed to be performed by a single continuously working crew. But in LOB it visualizes the delivery rate. The delivery rate, also called “natural rhythm” (Damci et al. 2013, p. 683), denotes “the speed by which work is to be finished in the repetitive units” (Hegazy 2001, p. 125). It does not assume just one crew and in fact often overlaps tasks within an activity, which will require multiple crews. In other words, the LSM slope means how fast work (by one crew) progresses, but the LOB slope means how fast units are completed (by several crews). This also leads to the fourth difference. In LOB a pair of lines envelopes the characteristic parallelogram shape. Within it, at each work unit exist a schedule sub-network. Its duration is the horizontal distance between the double lines. Sub-networks were even shown in the aforementioned and
otherwise LSM-oriented repetitive scheduling study (Harris and Ioannou 1998). A summary of these fundamental differences is illustrated by Figure 2, which fulfills Research Objective 1.

<table>
<thead>
<tr>
<th>Item</th>
<th>LSM</th>
<th>LOB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related CPM representation</td>
<td>AON diagram: Activity defined by node with one duration or two dates</td>
<td>AOA diagram: Activity defined by arrow with two labeled event nodes</td>
</tr>
<tr>
<td>CPM input of dependencies</td>
<td>AON requires FTS, STS, FTF, STF link with any lead or lag duration</td>
<td>AOA may require “dummy links” to correctly express exact link structure</td>
</tr>
<tr>
<td>Plot in 2D</td>
<td><img src="image" alt="Plot in 2D LSM" /></td>
<td><img src="image" alt="Plot in 2D LOB" /></td>
</tr>
<tr>
<td>Activity and task representation</td>
<td>Single line (may be staggered for multiple concurrent crews)</td>
<td>Start and finish event parallel lines (dashed line, not solid arrow)</td>
</tr>
<tr>
<td>Activity starts on vertical axis</td>
<td>Zero</td>
<td>One</td>
</tr>
<tr>
<td>Date convention</td>
<td>Start-of-unit counting</td>
<td>End-of-unit counting</td>
</tr>
<tr>
<td>Slope represents</td>
<td>Production rate</td>
<td>Delivery rate</td>
</tr>
<tr>
<td>Output generated</td>
<td>Link types and point of closest proximity between activity pairs</td>
<td>Not addressed explicitly, but represents relative productivities</td>
</tr>
</tbody>
</table>

Figure 2: Differences between LSM and LOB  
(adapted and expanded from Su and Lucko 2015c)

4. Singularity Functions Definition

Equation 1 establishes the basic expression within all singularity functions. The bracketed term distinguishes two cases for the relationship between the independent and dependent variables $y$ and $z$, depending on the value of the cutoff parameter $a$: If $y$ is smaller than $a$, then the pointed bracket equals 0. But if $y$ equals or is larger than $a$, then the pointed brackets are treated as round
ones and calculated as normal algebra prescribes. The definition of Equation 1 is left-continuous.

Modifying the scale parameter $s$ and exponent $n$ equips the function with flexible forms (Terry and Lucko 2012): If $n$ is negative, it acts like the Dirac $\delta$ function per Figure 3a; if $n$ is 0, it acts similar to the Heaviside function, where $s$ amplifies the step height per Figure 3b; if $n$ is 1, it becomes a linear growth function where $s$ determines the slope per Figure 3c; and if $n$ is 2 or larger, it describes a parabolic behavior or any higher-order non-linear function per Figure 3d.

$$z(y) = s \cdot (y-a)^n = \begin{cases} 
0 & \text{for } y < a \\
 s \cdot (y-a)^n & \text{for } y \geq a
\end{cases} \quad (1)$$

![Figure 3: Singularity Function Forms for Different Exponent $n$ ($a = 1$)](image)

(a) $n = -1$  (b) $n = 0$  (c) $n = 1$  (d) $n = 2$

5. Methodology for Multiple Crew Linear Schedule with Singularity Functions

Previous research on scheduling multiple crews handled several input parameters and constraints: Selinger (1980, p. 196) listed the “labour [sic] requirement per activity, and feasible quantities of resources, as basic input data”, plus constraints to fulfill the required resource continuity. Russell and Caselton (1988) allowed interruptions between tasks within activities to align activities with different slope. El-Rayes and Moselhi (1998, p. 433) stated that “to maximize the efficiency of crew utilization, the schedule of repetitive activities should be resource driven, and should satisfy the crew work continuity constraint in addition to precedence
relationships and crew availability constraints”. Hyari et al. (2009) used activity identifiers, sequencing relations, work quantity, and crew numbers, productivity, and costs in their algorithm. Based on such studies, Table 2 lists the inputs that this paper will employ. Values that are assumed as known are the available crew count ($C \,[-]$), number of work units that are assigned to each crew per time ($v_c \, [\text{unit/crew}]$), crew productivity ($v_p \, [\text{day/unit}]$), overlap period (i.e. a lead or lag) between adjacent crews ($t_o \, [\text{day}]$), lead time from the first to the second crew ($v_d \, [\text{day/unit}]$) (which equals to the delivery rate that can be calculated as $v_c \cdot v_p - t_o$), total work quantity ($U \, [\text{unit}]$), and start time and start work of the repetitive activity ($a_{ys}, \, a_{xs}$). An ‘as soon as possible’ policy applies: All crews shall start as soon as they can to decrease idle time and increase a learning effect (Hyari and El-Rayes 2006).

Table 2: Activity Parameters

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Domain</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C$</td>
<td>Available crew count</td>
<td>Positive integer: 1, 2, 3, …</td>
<td>crews</td>
</tr>
<tr>
<td>$v_c$</td>
<td>Work units per crew and time</td>
<td>Real number $\geq 0$</td>
<td>unit/crew</td>
</tr>
<tr>
<td>$v_p$</td>
<td>Crew productivity</td>
<td>Real number $\geq 0$</td>
<td>day/unit</td>
</tr>
<tr>
<td>$t_o$</td>
<td>Overlap period between adjacent crews</td>
<td>Real number $\geq 0$</td>
<td>day</td>
</tr>
<tr>
<td>$v_d = v_c \cdot v_p - t_o$</td>
<td>Lead time between 1$^{\text{st}}$ and 2$^{\text{nd}}$ crew, also delivery rate (calculated from $t_o$)</td>
<td>Real number $\geq 0$</td>
<td>day/unit</td>
</tr>
<tr>
<td>$U$</td>
<td>Total work</td>
<td>Real number $\geq 0$</td>
<td>units</td>
</tr>
<tr>
<td>$a_{ys}$</td>
<td>Start time</td>
<td>Real number $\geq 0$</td>
<td>day</td>
</tr>
<tr>
<td>$a_{xs}$</td>
<td>Start work</td>
<td>Real number $\geq 0$</td>
<td>unit</td>
</tr>
</tbody>
</table>

*Note: ASAP rule means that each crew starts work as soon as possible and works continuously*

Yang and Ioannou (2001) classified types of repetitive projects, discrete and continuous. A project of the former type consists of many units, where the actual work quantity in each unit may or may not be identical. Examples for this are floors of high rise buildings, foundation piles, or rooms of a house. The total work ($U$) is an integer and the start of work ($a_{ys}$) is 0. In contrast, a continuous repetitive project “involves a number of activities following each other rather than
the uniform repetition of a unit network” (Yang and Ioannou 2001, p. 5). Projects of this type are measured by their length, e.g. highways. For them, \( U \) and \( a_{cs} \) can be any positive real number.

In the following sections, pairwise relations between the dimensions of time, work, and crews are expressed mathematically as a crew assignment function \( r(x) \), crew utilization function \( r(y) \), and multiple crew linear schedule \( y(x) \), respectively. The \( r(x) \) reflects which crew will work on what unit as a monitoring tool for crew use. The \( r(y) \) will incorporate resource performance. And the \( y(x) \) function models how work grows over time for multiple crews in a linear schedule.

### 5.1 Crew Assignment Function \( r(x) \)

Equation 2 provides the crew assignment function \( r(x) \). It is inspired by research by the second author on calendarization with singularity functions: “[T]wo or more basic terms that contain rounded divisions can create repeating patterns... [V]arying their parameters \( s \), \( a \), and divisor within their basic terms customizes their pattern of recurring peaks... or valleys” (Lucko 2014, p. 235). The \( r(x) \) should be established ideally before the project starts. A basic assignment pattern is that all crews finish their assigned work units per time (\( v_c \) [unit/crew]) in the first cycle, then let the first crew move to the second cycle of work, and so forth until the total work is completed. Note that crews may receive dissimilar work quantities per task, e.g. some crews may have more experienced workers and be able to finish work faster. But workers can be distributed evenly among crews to some degree so that they have an approximately equal productivity. Therefore it can still initially be assumed that each crew is assigned the same work per task. The total work quantity \( U \) and the assigned work to each crew \( v_c \) can be any positive real number for discrete and continuous repetitive projects. Productivity is often fractional, e.g. half a floor per workweek.
Equation 2 is divided into two parts, the first half returns a stair function that starts at \( a_{sS} - v_c \) and finishes at \( a_{sS} - v_c + U \) with a step height and width of 1 and \( v_c \), respectively. Inserting the parameters of an integer scenario \( (a_{sS} = 0 \text{ unit}, C = 3 \text{ crews}, v_c = 1 \text{ unit/crew}, U = 9 \text{ units}) \) creates the profile per Figure 4a. The second half returns a similar stair function that starts at \( a_{sS} \) and finishes at \( a_{sS} + U \) with the height and width of \( C \) and \( C \cdot v_c \), respectively. Figure 4b is its profile. Their difference gives the crew assignment function per Figure 4c to show which crew performs what unit. Note that the vertical axis is crew number (a nominal identifier), not cumulative crew count. Equation 2 allows fractional inputs, which means that it can be used for both discrete and continuous repetitive project schedules. The outermost round-up operator envelops Equation 2 to guarantee that a fractional work unit will still be assigned to a crew number. For example, for \( a_{sS} = 0.5 \text{ unit}, C = 3 \text{ crews}, v_c = 1.6 \text{ unit/crew}, U = 8.5 \text{ units} \), the profile of Figure 5 will be correct.
Sometimes it is necessary to focus on specific crews when scheduling. The crew assignment model should therefore allow turning particular crews on or off. It will exploit the feature that the profile of the crew assignment function $r(x)_{\text{crews}}$ has a stair shape whose steps denote matches of units with crews. Equation 3 applies a controller term that returns 1 if $r(x)_{\text{crews}}$ is equal to a crew number, else 0. Summing from crews 1 to $C$ covers all crews. To turn off a specific crew number $n$, one replaces it with a value that does not equal any other, e.g. a very large number like 1,000.

$$r(x)_{\text{signal}} = \sum_{n=1}^{C} \left( r(x)_{\text{crews}} - n \right)^{0} \cdot \left( n - r(x)_{\text{crews}} \right)^{0}$$

(3)

5.2 Crew Utilization Function $r(y)$

According to Lumsden (1968, p. 21), LOB “provides a simple means of constructing a histogram of the labour [sic] loading which has been superimposed on the diagram”. Having such a complete histogram facilitates subsequently applying an optimization algorithm for resource allocation or resource leveling, e.g. the minimum moment algorithm (Christodoulou et al. 2010).

This paper will follow Harris’ (1978) approach to model the crew resource utilization: First plot individual crew bars along a time axis, then add them to generate the total crew resource
Overlapping bars create a peak when their counts are added on the vertical axis as Figure 6 shows conceptually. Note that this principle of adding shapes can also be applied to growth patterns, i.e. triangles, as will be used in the following, in addition to the rectangles of Figure 6.

![Figure 6: Adding Individual Resources for Resource Histogram]

The crew utilization model must fulfill several requirements: It must be able to handle single or multiple crew cases. Discrete and continuous repetitive projects should be possible. In other words, the parameters $U$ and $v_c$ should be fractional number, if necessary. And $U$ may or may not allow all crews to finish their units in many cycles. Or a crew may just work on one unit and be done. For clarity, these desirable features are implemented separately in the following equations, which are then aggregated into the general crew utilization function that Equation 7 provides.

Equation 4 acts as a controller to identify single crew cases. If the crew count $C = 1$, then its double terms are evaluated as $(1 - 1)^0 \cdot (1 - 1)^0 \cdot (\text{start and finish cutoffs}) = 1 \cdot 1 \cdot (\ldots)$. In other
words, they only give a multiplicative factor of 1 if there is a single crew and otherwise are zero.

Then this non-cumulative (because the exponents are 0) resource profile is just a single bar chart.

The following equations cover two cases: Equation 5 handles if the total unit of work is enough so that all crews finish at least one full cycle. Equation 6 handles if it is too small to allow this.

\[ r(y) = \langle C - 1 \rangle^0 \cdot \langle 1 - C \rangle^0 \cdot \left( y - a_{ss}^* \right)^0 \cdot \left( y - \left( a_{ss}^* + U \cdot v_p \right) \right)^0 \]  

(4)

Equation 5 calculates the duration for the case where all crews perform multiple cycles for their portion of the total work quantity \( U \), which is modeled with the variables as defined in Table 2 in the numbered Terms I-VII of this equation: The product \( \langle C - n \rangle^0 \cdot \langle U - C \cdot v_c \rangle^0 \) of Term I enforces that the crew number \( n \) remains smaller than its available crew count \( C \), e.g. if \( C = 3 \) crews, \( n \) cannot be 4. The second pointed bracket in Term I controls that the total work quantity \( U \) is larger than \( C \cdot v_c \), so that all crews will be working at least on one unit of \( U \) itself.

\[ r(y)_2 = \sum_{n=1}^{C} \left( \langle C - n \rangle^0 \cdot \langle U - C \cdot v_c \rangle^0 \right) \cdot \left( y - \left( a_{ss}^* + (n-1) \cdot (v_c \cdot v_p - t_o) \right) \right)^0 \]

(5)

Term II models the start time of the \( n^{th} \) crew. Here the activity start time \( a_{ss}^* \) is also the first crew start time. Its following product \( v_c \cdot v_p - t_o \) is the lead time from the first to the second crew, which is also the delivery rate per Table 2. Adding them will give the start time of the \( n^{th} \) crew.

The remaining terms model several cases for activity starts and finishes: Since units of work product and their crew cycles may be fractional, these terms express all possible cases: Terms III
and IV calculate durations of crews with full or leftover fractional cycles, respectively; for the latter Term V checks if the leftover is less than the assigned work; and Term VI checks if it is more than zero. Terms V and VI are multiplied so that they become boundaries for the leftover. If their product is one, i.e. true, then Term VII gives the duration of crews whose leftover is less than their assigned units of work product. The following sections explain these terms in detail.

In Term III, only the start time is needed, because the finish equals the start plus one period. The round-down operator \( \lfloor U/(C \cdot v_c) \rfloor \) calculates the number of full cycles per crew, e.g. for \( U = 5 \) units, \( C = 2 \) crews, \( v_c = 1 \) unit/crew it is \( \lfloor 5/2 \cdot 1 \rfloor = \lfloor 2.5 \rfloor = 2 \) cycles. Multiplying it with the period \( v_c \cdot v_p \) (1 unit/crew, crew productivity is 1 day/unit) is the duration of a crew for its cycles. Crew 1 e.g. performs two cycles in \( 2 \cdot v_c \cdot v_p = 2 \) units \( \cdot 1 \) unit/crew \( \cdot 1 \) day/unit \( \cdot \) crew \( = 2 \) days.

The next two terms cover two cases that may exist for fractional work: Case 1 in Term IV models only the crews who perform full cycles. For example, for \( U = 5.7 \) units, \( C = 3 \) crews, \( v_c = 1 \) unit/crew, crews 1 and 2 complete full cycles, but crew 3 only does 0.7 units. Term V therefore separately deals with any crews who have fractional finish cycles, here crew 3. But first, \( U - \lfloor U/(C \cdot v_c) \rfloor \cdot C \cdot v_p \) in Term IV gives the amount of the unit in the last cycle and \( n \cdot v_c \) calculates how many units should have been finished by the \( n^{th} \) crew in a full cycle. For example, for crew 1, 1 crew \( \cdot 1 \) unit/crew \( = 1 \) unit should have been done; for crew 2, 2 units; for crew 3, 3 units; and so forth. They sum to a full cycle. Term IV assesses if the leftover units of work product are enough for each crew to perform at least a full cycle. For example, for crew 1, \( (5.7 - \lfloor 5.7 / 3 \cdot 1 \rfloor \cdot 3 \cdot 1 - 1 \cdot 1) - 0 \rfloor = (5.7 - 3 - 1) - 0 \rfloor = 1 \), i.e. crew 1 can finish its work in the last cycle. Term IV is multiplied with \( v_c \cdot v_p \) to give the duration of that crew. Increasing \( n \), crew 2 can also finish its assigned job. Since \( 0 \rfloor = 1 \), if the last cycle is still full, this term would give a positive answer.
But here for crew 3, $(5.7 - \lfloor 5.7 / 3 \rfloor - 3 \cdot 1 - 3 \cdot 1) - 0)^0 = \langle (5.7 - 3 - 3) - 0 \rangle^0 = 0$, which means that crew 3 has no full cycle, which is Case 2. Its fractional cycle is therefore handled in Term V.

The product of Terms V through VII assesses if work of a crew in the last fractional cycle is less than its assigned work. As mentioned before, 0.7 units are leftover for crew 3. The term $U - \lfloor U / (C \cdot v_c) \rfloor \cdot C \cdot v_c - n \cdot v_c$ calculates leftover work of a crew, which may be positive or negative with a value larger than $-v_c$. Term VI is left-continuous (as its subscript indicates) to cancel $0^0 = 1$, because the product must be smaller than zero, not smaller than or equal to zero. If leftover work is insufficient, Term VII calculates the duration of such fractional assignment, e.g. $(5.7 - \lfloor 5.7 / 3 \rfloor - 3 \cdot 1 - (3 \cdot - 1) \cdot 1) \cdot 1 = 0.7$ days, which is simply added to the finish time.

Equation 6 calculates the duration if $U$ is so small that some crews finish less than one cycle. In it, $(C \cdot v_c - U)^0$ assesses if this is the case. Its left-continuous singularity function can cancel out $C \cdot v_c = U$ from Equation 5 (which had modeled a full cycle). Just as before, two cases exist: Case 1 models crews who perform full cycles in Terms I-III. For example, for $U = 5.2$ units, $C = 3$ crews, $v_c = 2$ unit/crew, each crew is assigned 2 work units per time. Crews 1 and 2 complete full cycles, but crew 3 has only 1.2 units to do. This fractional cycle is handled under Case 2.

The term $(U - n \cdot v_c)^0$ in Term IV checks if the total work is only enough for less than one cycle. Then $v_c \cdot v_p$ calculates the duration of the full cycles of a crew. Terms V and VI again assess if a crew may perform less than a full cycle due to fractional work. Term V and VI have the same effect as an inequality that enforces $-v_c < U - n \cdot v_c < 0$. The $(U - (n - 1) \cdot v_c) \cdot v_p$ in Term VII calculates the period. For example, for $U = 5.2$ units, $C = 3$ crews, $v_c = 2$ units/crew, for crew 3 the leftover 1.2 units prevent it from finishing a full cycle ($v_c = 2$ units/crew). Evaluating the inequality gives $-v_c < U - n \cdot v_c < 0 \Rightarrow -2 < 5.2 - 3 \cdot 2 < 0 \Rightarrow -2 < -0.8 < 0$, which is true and its duration is only $(U - (n - 1) \cdot v_c) \cdot v_p = (5.2 - (3 - 1) \cdot 2 \text{ unit/crew}) \cdot 1 \text{ day/unit} = 1.2 \text{ days}. $
\[ r(y) = \frac{C}{\omega} \left( (C - n)^{b} \cdot \left( C \cdot v_{c} - U \right)^{o} \right) \cdot \left( y - (a_{SS} + (n-1) \cdot (v_{c} \cdot v_{p} - t_{o}))^{o} \right) \]

\[ - \left( y - (a_{SS} + (n-1) \cdot (v_{c} \cdot v_{p} - t_{o}))^{o} \right) \cdot \left( y - (a_{SS} + (n-1) \cdot (v_{c} \cdot v_{p} - t_{o}))^{o} \right) \cdot \left( U - n \cdot v_{c}^{o} \cdot v_{c} \right) \]

\[ + \left( U - n \cdot v_{c}^{o} \cdot v_{c} \cdot (0 - (U - n \cdot v_{c})^{0} \cdot v_{c} \cdot v_{p}) \right) \]

(6)

Combining case distinctions for full and fractional cycles of Equations 4 through 6, Equation 7 finally is the general crew utilization function for discrete and continuous repetitive projects. It considers of single and multiple crews with another term of \((C - 1)^{0} \cdot (1 - C)^{0}\) to cover all cases.

\[ r(y)_{\text{crew}} = r(y)_{\text{cycle}} + \left( 1 - (C - 1)^{0} \cdot (1 - C)^{0} \right) \cdot (r(y)_{\text{cycle}} + r(y)_{\text{cycle}}) \quad \text{subject to} \quad (C \cdot v_{c} - U) < v_{c} \]  

(7)

A constraint \((C \cdot v_{c} - U) < v_{c}\) must be added, so that the product of the crew and the units of each crew should not be equal or larger than the work of one crew per cycle, e.g. \(U = 4\) units, \(C = 3\) crews, \(v_{c} = 2\) units/crew would be infeasible, because crew 3 would idle, which would be a waste. The aforementioned combinations of example inputs for the crew utilization function are illustrated in Figure 7, which is applicable to both discrete and continuous repetitive projects.

![Image of Figure 7](image-url)

**Figure 7:** Crew Utilization Function Scenarios \((a) U = 5\) units, \(C = 2\) crews, \(v_{c} = 1\) unit/crew \n
**Figure 7:** Crew Utilization Function Scenarios \((b) U = 5.7\) units, \(C = 3\) crews, \(v_{c} = 1\) unit/crew \n
**Figure 7:** Crew Utilization Function Scenarios \((c) U = 5.2\) units, \(C = 3\) crews, \(v_{c} = 2\) units/crew
5.3 Multiple Crew Linear Schedule $y(x)_{\text{crews}}$

The multiple crew linear schedule $y(x)_{\text{crews}}$ is a unique linear schedule pattern, where a group of overlapped line segments represents different crews who conduct different units of work. Due to the overlap in time, the relation $x(y)$ of time and work is not a true function, as multiple $x$-values appear for a single $y$-value in Figure 8a. Yet by swapping independent and dependent variables, $y(x)$ can be expressed as a function. Its profile $f_{\text{MCLS}}$ is represented by subtracting a stair function $f_{\text{stair}}$ and an offset function $f_{\text{offset}}$ from a linear function $f_{\text{line}}$ per Figure 8b. The $f_{\text{stair}}$ will be affected by the overlapping $t_o$ between adjacent crews; the $f_{\text{offset}}$ is influenced by the lead time $v_d$ between crews. Their combination fulfills the ASAP rule, i.e. crews 1 and 2 are working continuously.

![Figure 8: Multiple Crew Linear Schedule](image)

$U=5$ units, $C=3$ crews, $v_c=1$ unit/crew, $a_yS=0$ day, $t_o=0.5$ day, $v_p=1$ day/unit

The detailed $y(x)_{\text{crews}}$ function is given by Equation 8. Terms I, II, and III follow the concept $f_{\text{MCLS}} = f_{\text{line}} - f_{\text{stair}} - f_{\text{offset}}$. Term I is a linear segment that starts at $a_xS$ and finishes at $a_xS + U$, whose slope is the crew productivity $v_p$. If only one crew exists, this term is the linear schedule...
The factor $1 - \langle C - 1 \rangle^0 \cdot \langle 1 - C \rangle^0$ in Term II term controls if multiple crews exist: It equals 1 if $C$ is larger than one, and equals 0 if $C$ is one. The whole term is a stair function that has the same start and finish as the linear segment, but with period $v_c$. In other words, the width of each step equals the number of units that are assigned to each crew per time. The overlap period $t_o$ between adjacent crews determines the height of the stair function. The factor $((C - 1) \cdot v_d - t_o)$ in Term III controls the offset that that the model fulfills the ASAP rule: The start time of a crew at the next cycle should be equal to its finish time in the previous cycle. The floor operators in Term III are similar to those in Term II, but with a different width of each step $C \cdot v_c$, which equals the total work quantity of one cycle of all crews.

$$y(x)_{crews} = v_p \cdot \left( \langle x - a_{ss} \rangle^i - \langle x - (a_{ss} + U) \rangle^i - U \cdot \langle x - (a_{ss} + U) \rangle^0 \right)_t - t_o \cdot \left( (C - 1)^0 \cdot \langle 1 - C \rangle^0 \right)$$

$$\cdot \left( \left( \frac{x - a_{ss}}{v_c} - 0 \right)^i - \left( \frac{x - a_{ss}}{v_c} - U \cdot \frac{x - (a_{ss} + U)}{v_c} \right)^0_{III} \right)$$

$$\cdot \left( (C - 1)^0 \cdot \langle 1 - C \rangle^0 \right) \cdot \left( \left( \frac{x - a_{ss}}{C \cdot v_c} - 0 \right)^i - \left( \frac{x - a_{ss}}{C \cdot v_c} - \frac{U}{C \cdot v_c} \right)^0_{III} \right)$$

5.4 Possible Use and Relations of Functions

The three functions – crew assignment model, which is a bar chart $r(x)$; crew utilization function, which is a bar chart $r(y)$, and the multiple crew linear schedule $y(x)_{crews}$ – share the inputs of Table 2. Changing any value in one model will automatically alter the other two. Their plots are shown in Figure 9. Combined with the signal function of Equation 3, these functions model how the selected crews perform. The plots for crews 2 and 3 are shown in Figure 10.
Patterns for multiple crews vary: They may partially overlap, as is common in LOB, be staggered (concurrent), sequential, or interruptible in the schedule. Modeling these diverse cases is realized by modifying the parameter $t_o$: Partial overlap if $0 < t_o < v_p \cdot v_c$; staggering if $t_o = v_p \cdot v_c$; sequential if $t_o = 0$, and interruptible if $t_o < 0$. These diverse scenarios are shown in Figure 11.
Figure 11: Staggering, Sequential, and Interruptible Cases

(a) Staggering $t_o = 1$ day  
(b) Sequential $t_o = 0$ day  
(c) Interruptible $t_o = -0.5$ day

$U = 5.7$ units, $C = 3$ crews, $v_c = 1$ unit/crew, $a_{ys} = 0$ day, $v_p = 1$ day/unit

Lucko and Su (2014) noted that projecting a 3D model onto a 2D plane yields the behavior in those dimensions. Applying this concept, an integrated 3D model for the multiple crew linear scheduling also exists as shown in Figure 12d. The multiple crew linear schedule $y(x)_{crew}$ in the $y$-$x$ plane of Figure 12a, the crew assignment function $r(x)$ in the $x$-$r$ plane in Figure 12b, and the crew utilization function $r(y)$ in the $r$-$y$ plane in Figure 12c. Such integrated 3D model illustrates the crucial nature of time-work-crew interactions. Exploring the model is equivalent to more advisable than three separate 2D models; they are merely projections from different perspectives. Having modeled work of multiple crews as singularity functions fulfills Research Objective 2.
6. Validation of Productivity Scheduling Method for Multiple Crew Linear Schedule

Linear scheduling of repetitive projects was an essentially graphical approach. It was modeled for PSM, whose standard process is “formulating initial equations, stacking and consolidating them, and deriving information about their criticality” (Lucko 2008, p. 711). But this method has never been applied to multiple crews. This paper fills this gap to expand theory with LOB-PSM. Its merit is threefold: First, it heeds the advice of Ockham’s razor on the level of detail of models – ‘as simple as possible, but as complex as necessary’ to fully capture the real-world interactions of crews as productive resources with time and work in linear and repetitive schedules. Second, its mathematical formulation flexibly allows different types of analyses, e.g. minimizing total project duration, resource leveling, or performing a multi-objective analysis. Neither of these would be possible at the aforementioned graphical level at which LOB still resides. Third, the
model and its algorithm that the following section will describe are suitable for use with any existing optimization approach, whether that would directly calculate a solution, employ heuristic rules-of-thumb, or apply an evolutionary method that is inspired by nature. Depending on the chosen optimization approach, a solution is exact (only possible for small projects of limited complexity with a smooth solution space), very good, but not guaranteed (depending on the sophistication of heuristics), or as-good-as-needed (by letting an evolutionary method run until a formal stopping criterion has been reached, e.g. having not found a better solution within a certain number of randomized runs). The reason for this distinction of model and optimization is their separate roles: The mathematical formulation contains all parameters as variables per Table 2; i.e. it embodies all possible permutations of a given example of a project schedule. Holding some of these constraints constant while varying others can solve various different optimization problems, e.g. for limited crews $C$ and total work $U$, what is the optimum crew assignment $v_c$ and $v_p$ for the minimum total project duration or to level resources? What if there were many different types of crews? What if the crews are multi-skilled? The target function then describes the pairwise relation between the time, work and crews aspects in form of a crew assignment function $r(x)$, crew utilization function $r(y)$, or multiple crew linear schedule $y(x)$, respectively. In other words, the model builds a bridge from the real world to the optimization. Independently, users can tailor to their needs and preferences how to search solutions until an optimum is deemed to have been reached. Of course, the latter step is typically computerized for speed. Experimentation with different optimization methods is beyond the scope of this paper.
6.1 Initializing Step

The first step of PSM is to set up the activity equations without yet defining links among them. Different from the old PSM, here the multiple crew linear schedule \( y(x)_{crews} \) per Equation 8 is used instead of a single line form. The buffer function of an activity is denoted as \( y(x)_{crews} \) plus a buffer term per Equation 9. For validation, the often-reanalyzed example by Harris and Ioannou (1998) is the input. Each activity can newly be conducted by two crews per Table 3. By inserting the values into Equations 8 and 9, the profile of this initializing step is visualized in Figure 13a.

\[
y(x)_{buffer} = D_{buffer} \cdot \left( x - a_{sS} \right)^{\alpha} - \left( x - (a_{sS} + U) \right)^{\alpha} + y(x)_{crews}
\] (9)

![Images of graphs and Table 3](image-url)

Figure 13: Plot of LOB-PSM Processes

Table 3: Inputs of a Validation Example with Six Activities
(modified based on Harris and Ioannou 1998)

<table>
<thead>
<tr>
<th>Name</th>
<th>Successor</th>
<th>Duration [day]</th>
<th>U [unit]</th>
<th>Buffer [day]</th>
<th>a_{sS} [unit]</th>
<th>C [crew]</th>
<th>v_c [unit/crew]</th>
<th>v_p [day/unit]</th>
<th>t_o [day]</th>
<th>v_d^* [day/unit]</th>
</tr>
</thead>
<tbody>
<tr>
<td>A_1</td>
<td>B, C</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0.5</td>
<td>1.5</td>
</tr>
<tr>
<td>A_2</td>
<td></td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>0.5</td>
<td>3.5</td>
<td></td>
</tr>
<tr>
<td>A_3</td>
<td></td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0.5</td>
<td>1.5</td>
</tr>
<tr>
<td>B_1</td>
<td>D</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>B_2</td>
<td></td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>
6.2 Stacking Step

Having established the initial activity and buffer equations, this step sequences predecessors and successors in the schedule with a ‘pessimistic’ view. Here the earliest start time of a successor equals the latest finish time of the predecessor buffer, which Figure 13b shows as dotted arrows. The total duration will be 44.5 days, and the last work task is the 6th unit of activity F by crew 2.

6.3 Consolidating Step

This step calculates the minimum distance between the successor and its predecessor buffer and deduct it “from the successor equation to shift it down into its earliest possible position where it touches its predecessor” (Lucko 2008, p. 716). Consolidating these stacked activities improves the schedule per Figure 13c. In this step, the last work task is still the 6th unit of activity F by crew 2, but the total duration is only 25.5 days. But modifying inputs, e.g. number of crews per activity, the number of units assigned to each crew per time $v_c$, and start working unit $a_s$ (for a continuous repetitive project) will generate different total durations in Figures 14a through 14c. These scenarios are 30 days for $C = 1, v_c = 1$; 17.5 days for $C = 2, v_c = 0.5$; and 22.5 days for $C = 2, v_c = 1$ with $A$ stopping at the 5th unit and $E$ starting from the 2nd unit. Crew resource utilization profiles $r(y)$ are plotted per Equation 7 in Figures 14d through 14f, which allows monitoring if a crew exceeds its availability limit. Having established LOB-PSM fulfills Research Objective 3.
This example is small enough that an optimum solution can be determined with certainty for any particular set of inputs. Results of Figures 14a through 14c have been found with a computer implementation for the case of minimum total project duration. They represent, respectively, the optimum schedule for a single crew with a productivity of one work unit per crew and time unit (Figure 14a) and for two crews with half that productivity, but overlapping (Figure 14b), both for a project with discrete work units, and two crews with the initial productivity and overlapping for a continuous project (Figure 14c). For brevity, resource leveling is excluded from this paper.
7. Conclusions and Contributions to the Body of Knowledge

Repetitive scheduling is a common trait of construction projects with numerous identical units, e.g. high-rise buildings, bridge segments, road miles, etc. Scheduling these projects properly will not only save time and cost, but also keep crews working smoothly and efficiently. Traditionally the activities that are performed across such repetitive units have been portrayed as lines in LSM. But to shorten the total project duration, multiple crews can be deployed to work on different units within the same activity. The LOB approach excels in such cases. Both LSM and LOB are repetitive scheduling methods, but possess unique advantages and disadvantages: While LSM has been mathematically formulated as PSM, which provides the theory for starts and finishes, time and work buffers, criticality and float, shifts and delays, LOB has remained a detailed, but limited graphical approach to visualize the timing of multiple crews. Motivated to complement and merge both LSM and LOB, the contributions of the body of the knowledge of this study are:

- Basic differences between LSM and LOB, including activity representation, starts points, and velocity measurement, have been extracted and explained mathematically and graphically;
- Multiple crew linear scheduling has been derived with singularity functions. Three functions compose the model: Crew assignment function, crew utilization function, and multiple crew linear schedule. They can be applied to both discrete and continuous repetitive cases. Their formulations are directly connected, so that altering one parameter automatically modifies all outputs. This enables more efficient scheduling in future research and practice. Finally, a 3D integrated perspective view of the time, work, and resource performance has been provided.
The combined approach has been validated by reanalyzing a schedule by Harris and Ioannou (1998). By modifying input values, the model applied its powerful ability to automatically generate single or multiple crew linear schedules. This has extended PSM from its previous limited merely activity-oriented capabilities to a truly crew-oriented scheduling method.

In summary, this paper has merged LSM and LOB by mathematically combining the time, work, and resources to provide a comprehensive theory for the repetitive scheduling domain. Future research should address issues of optimization and practical use: (1) Assuming fixed crews, what is the optimum allocation to different activities if crews are unique to an activity or universal? (2) If minimizing duration is the objective, what are the optimum values of the parameters in Table 3 (e.g. \( C, v_p, t_o, v_c \), etc.)? (3) What other beneficial uses may exist for the integrated 3D Figure 12?

8. Acknowledgements

The authors thank the two anonymous reviewers for the detailed constructive feedback that has improved the presentation of the approach for scheduling multiple crews as is described herein.

9. References


10. Notation

**Symbols**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>cutoff in singularity function;</td>
</tr>
<tr>
<td>$C$</td>
<td>available crew count;</td>
</tr>
<tr>
<td>$D$</td>
<td>duration;</td>
</tr>
<tr>
<td>$f$</td>
<td>function;</td>
</tr>
<tr>
<td>$n$</td>
<td>exponent in singularity function; OR integer sequential number of crew</td>
</tr>
<tr>
<td>$r$</td>
<td>crew resource, could be a function of time $r(y)$, or a function of work $r(x)$;</td>
</tr>
<tr>
<td>$s$</td>
<td>scaling factor in singularity function;</td>
</tr>
<tr>
<td>$t_o$</td>
<td>overlapped period between adjacent crews;</td>
</tr>
<tr>
<td>$U$</td>
<td>number of units of total work amount;</td>
</tr>
<tr>
<td>$v_c$</td>
<td>number of units assigned to each crew;</td>
</tr>
</tbody>
</table>
$v_d =$ lead time between the first and the second crew;

$v_p =$ productivity of crew;

$x =$ variable along work;

$y =$ variable along time;

$z =$ dependent variable along vertical axis;

$\langle \rangle =$ brackets of singularity functions;

$\lfloor \rfloor =$ floor operator rounding downward to integer.

**Subscripts**

$A_{buffer} =$ buffer function of activity A;

$Buffer_A =$ buffer value of activity A;

$crews =$ crew linear scheduling function;

$crews_A =$ crew linear scheduling function of activity A;

$left =$ left continuous definition of the singularity function;

$line =$ linear function;

$MCLS =$ multiple crew linear scheduling;

$offset =$ offset function;

$stair =$ stair function;

$xS =$ start work unit of an activity;

$yS =$ start time of an activity;

$S =$ start of an activity;

$1 =$ index for $r(x)$ function;

$2 =$ index for $r(x)$ function;

$3 =$ index for $r(x)$ function.
Figure 1: Differences between Line of Balance Approaches

I. Production Plan for One Unit

<table>
<thead>
<tr>
<th>User</th>
<th>Work Time</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Production Duration</td>
<td>12</td>
<td>10</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Time axis direction from right to left, but number control points from left to right

II. Objective Chart with Target Output

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Date</th>
<th>Calendar Time [weeks]</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>1. Control Point Durations until Due Date</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

III. Progress Chart with Line of Balance

<table>
<thead>
<tr>
<th>Product [count]</th>
<th>Planned</th>
<th>Actual</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>Control Points</th>
</tr>
</thead>
</table>

(a) Original in Manufacturing

(b) Current in Construction

1. Determine Number of Repetitive Units
   8 work units

2. Determine Activity Sequence
   {A – B – C}

3. Determine Single Task Durations
   {2, 3, 1 weeks}

4. Determine Number of Crews
   {2, 2, 1 crews}

5. Total Units Used as 100%

6. Draw from 1 work unit

7. Add Double Lines Around Each Task

8. Slope is Delivery Rate
   {1, 2/3, 1 per week}

9. Criticality and Float
   {Late float, fully critical, early float}

10. Total Project Duration
    16.5 weeks
<table>
<thead>
<tr>
<th>Item</th>
<th>LSM</th>
<th>LOB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Concept applied</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity defined by</td>
<td></td>
<td></td>
</tr>
<tr>
<td>node with one duration or two dates</td>
<td></td>
<td>AOA diagram: Activity defined by arrow with two labeled event nodes</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AON offers FTS, STS, FTF, STF</td>
<td></td>
<td>AOA layout is inherited by LOB</td>
</tr>
<tr>
<td><strong>Plot in 2D</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 crews, staggered by 2 time units</td>
<td></td>
<td>2 crews, staggered by 2 time units</td>
</tr>
<tr>
<td><strong>Activity representation</strong></td>
<td>Single line (staggered)</td>
<td>Start and finish event parallel lines (dashed line, not solid arrow)</td>
</tr>
<tr>
<td><strong>Activity starts on vertical axis</strong></td>
<td>Zero</td>
<td>One</td>
</tr>
<tr>
<td><strong>Slope represents</strong></td>
<td>Production rate</td>
<td>Delivery rate</td>
</tr>
</tbody>
</table>

*Figure 2: Differences between LSM and LOB*  
(adapted and expanded from Su and Lucko 2015c)
Figure 3: Singularity Function Forms for Different Exponent $n$ ($a = 1$)
Figure 4: Integer Inputs to Crew Assignment Function

\( a_{xS} = 0 \text{ unit, } C = 3 \text{ crews, } v_c = 1 \text{ unit/crew, } U = 9 \text{ units} \)
Figure 5: Fractional Inputs to Crew Assignment Function
($a_x = 0.5 \text{ unit, } C = 3 \text{ crews, } v_c = 1.6 \text{ unit/crew, } U = 8.5 \text{ units}$)
Figure 6: Adding Individual Resources for Resource Histogram
(a) $U = 5$ units, $C = 2$ crews, $v_c = 1$ unit/crew  
(b) $U = 5.7$ units, $C = 3$ crews, $v_c = 1$ unit/crew  
(c) $U = 5.2$ units, $C = 3$ crews, $v_c = 2$ units/crew

Figure 7: Crew Utilization Function Scenarios ($a_{YS} = 0$ day, $t_o = 0.5$ day, $v_p = 1$ day/unit)
(a) $x(y)$ is not a function; one $y$ matches multiple $x$ values

(b) $y(x)$ is a function;
$f_{\text{MCLS}} = f_{\text{line}} - f_{\text{stair}} - f_{\text{offset}}$

Figure 8: Multiple Crew Linear Schedule
($U = 5$ units, $C = 3$ crews, $v_c = 1$ unit/crew, $a_{js} = 0$ day, $t_o = 0.5$ day, $v_p = 1$ day/unit)
Figure 9: Components of Multiple Crew Linear Schedule

\( U = 5.7 \text{ units}, \ C = 3 \text{ crews}, \ \nu_c = 1 \text{ unit/crew}, \ \alpha_y = 0 \text{ day}, \ t_o = 0.5 \text{ day}, \ \nu_p = 1 \text{ day/unit} \)
Figure 10: Components of Crews 2 and 3

(U = 5.7 units, C = 3 crews, \( v_c = 1 \) unit/crew, \( a_{yS} = 0 \) day, \( t_o = 0.5 \) day, \( v_p = 1 \) day/unit)
Figure 11: Staggering, Sequential, and Interruptible Cases

(a) Staggering $t_o = 1$ day  (b) Sequential $t_o = 0$ day  (c) Interruptible $t_o = -0.5$ day

$U = 5.7$ units, $C = 3$ crews, $v_c = 1$ unit/crew, $a_y = 0$ day, $v_p = 1$ day/unit
Figure 12: 3D Integrated Multiple Crew Linear Schedule and 2D Projections

(a) Multiple Crew Linear Schedule \( y(x)_{\text{crews}} \)

(b) Crew Assignment Function \( r(x) \)

(c) Crew Utilization Function \( r(y) \)

(d) 3D View
Figure 13: Plot of LOB-PSM Processes

(a) Initiating

(b) Stacking

(c) Consolidating
Figure 14: MCLS $y(x)$ and Crew Resource Profile $r(y)$
for Different C Scenarios of LOB-PSM