A linear-discrete scheduling model for the resource-constrained project scheduling problem
Pierre Bonnal, Didier Gourc, Ari-Pekka Hameri, Germain Lacoste

To cite this version:

HAL Id: hal-01663967
https://hal.archives-ouvertes.fr/hal-01663967
Submitted on 13 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
A linear-discrete scheduling model for the resource-constrained project scheduling problem

PIERRE BONNAL1*, DIDIER GOURC2, ARI-PEKKA HAMERI3 and GERMAIN LACOSTE4
1CERN, CH1211 Geneva 23, Switzerland
2ENSTIMAC, F81013 Albi, France
3UNIL-HEC, CH1015 Lausanne, Switzerland
4ENT, F65016 Tarbes, France

Received 16 July 2003; accepted 26 November 2004

For some specific types of construction projects, the classical CPM or PDM scheduling techniques are not the most suitable. Few specific scheduling approaches have been developed to cope with construction projects that are made of either repetitive activities or activities with linear developments. But real-world construction projects do not consist only of such activities. They are generally made of a mixture of linear and/or repetitive activities and of more conventional activities. To allow this, the linear scheduling problem is reformulated, so classical schedule calculation approaches can be used. The implementation of some Allen’s algebra features to avoid adverse discontinuities and to allow crew/work continuity, together with a resource-driven and space-constrained scheduling are among the key features of the proposed approach. It is also a spin-off of off-the-field practices used for scheduling real projects in the particle accelerator construction domain; an excerpt from such a construction project is provided for illustrating the methodology.

Keywords: Construction scheduling, linear scheduling, optimization

Introduction

Over the last two decades, several articles (Selinger, 1980; Johnston, 1981; Chrzanowski and Johnston, 1986; Handa and Barcia, 1986; Russell and Caselton, 1988; Eldin and Senouci, 1994, 2000; Harmelink and Rowings, 1998; Harmelink, 2001; Moselhi and Hassanein, 2003) have been published to address the scheduling problem of construction or installation projects which typically have a linear development. Most of these papers refer to the so-called Linear Scheduling Model (LSM) that is an alternate methodology to the 40-year-old Critical Path Method (CPM). The linear scheduling methods that are presented in these papers do not allow considering systematically resources except the spatial one. In the present article, we have reformulated the problem so it can be treated as a resource-constrained project scheduling problem. Repetitive construction scheduling problems are a class of problems similar to those that can benefit from the LSM approach; the approach presented in the present article can also be seen as a generalization of the repetitive construction scheduling to the linear scheduling.

The projects concerned by the LSM methodology are projects that consist of a majority of linear activities. Harmelink and Rowings (1998) give the following definition to the latter: linear activities are those activities that are completed as they progress along a path. For instance, the digging of a railway tunnel, the repaving of a motorway or the installation and interconnection works of a particle accelerator, are typically projects that belong to the family of linear development projects.

The implementations of LSM-like approaches on real life projects have demonstrated their efficiency. From a practitioner point of view, the benefits of using the approach stem from:

- a smaller breakdown of activities is sufficient to perform the activity network analysis; and
- the schedules issued are more concise, which improve communication.

Among the scheduling approaches suited to a particular class of construction project, one should also mention

*Author for correspondence. E-mail: pierre.bonnal@cern.ch
all the approaches proposed for dealing with repetitive projects (Ashley, 1980; Kavanagh, 1985; O’Brien et al., 1985; Reda, 1990; Russell and Wong, 1993; El-Rayes and Moselhi, 2001; El-Rayes, 2001; El-Rayes et al., 2002) and the Line-of-Balance approach, proposed by Fouch in the 1940s to respond to industrial scheduling problems at Goodyear Tire & Rubber Co. (see Suhail and Neale, 1994, for instance). The LSM approaches differ from the latter by the fact that they are applicable to activities with a pure linear development that cannot be systematically discretized. For instance, the notion of start and finish stations is something that is specific to LSM.

In terms of development, repetitive scheduling approaches have certainly reached a higher level of maturity. Several successful attempts have been made to make schedule analyses systematic: CPM-like calculation with forward and backward passes, resource-driven scheduling (Moselhi and El-Rayes, 1993; Russell and Wong, 1993; Suhail and Neale, 1994; El-Rayes and Moselhi, 1998; El-Rayes, 2001; Leu and Hwang, 2001; El-Rayes et al., 2002). LSM approaches have not yet benefited of such functionalities; LSM schedule calculations are often limited to graphical analyses.

As mentioned by El-Rayes (2001), scheduling of repetitive construction projects can be significantly improved by considering three main practical requirements. These requirements are (1) the application of resource-driven scheduling that enable crew/work continuity, (2) the minimization of the project duration by an optimized utilization of the resources available, and (3) the integration in a single activity network of activities on different types: repetitive and non-repetitive activities for instance.

The aim of the present article is to describe a scheduling methodology that is suitable to projects that have activities with linear developments, and that integrates the three practical requirements mentioned here before. To do so, the problem has been formulated in such a way resource-constrained scheduling methodologies can be implemented. Several types of activities are supported: activities that are space-constrained can be mixed with activities that are not; activities with a linear development can be mixed with discrete activities. Some Allen’s algebra features (Allen, 1983) are implemented for addressing the crew/work continuity issue that is also mentioned here before.

The scheduling methodology presented in this article is henceforth called LDSM (Linear-Discrete Scheduling Model) for distinguishing it among the many scheduling techniques that are available for addressing this type of construction problem.

The rest of the article is organized in the following way: definitions are given in section 2. Then an approach for solving a linear-discrete scheduling (LDS) problem is proposed in section 3. Section 4 details the relevant algorithms. Before conclusion, this approach is applied to a real case problem (section 5).

Definitions and introductory remarks

Space-constrained vs. non-space-constrained activities

Space-constrained activities can be defined as activities that rely on the availability of a scarce spatial resource for being schedules, and then executed. Non-space-constrained activities are activities for which spatial resources do not need to be considered to have them scheduled, and then executed. In the framework of such construction project, space-constrained activities are in general the construction/installation activities carried out on the construction site (excavation works, digging works, pipe works, cable pulling, etc.) while non-space-constrained activities are those performed offsite (engineering works, etc.) or at supplier or contractor premises (prefabrication, etc.).

Linear vs. discrete-space-constrained activities

Linear space-constrained activities have been defined here before: these are activities that progress along a path; they generally use this ‘path’ a renewable resource, e.g. the digging of a railway tunnel, the paving of a motorway, etc. Discrete-space-constrained activities are activities that also use a spatial renewable resource, but that do not progress along a path, e.g. the digging of a small alcove in a railway tunnel, the earthwork for a junction at the crossing of two motorways, etc.

Linear-discrete schedule diagramming

Above all, schedules are communication media even if the underlying optimization mechanisms are of prime importance. Gantt charts are known to be the best means for communicating the temporal information of a project made of non-space-constrained activities. On these charts, the time goes along the horizontal axis from left to right, while the Work Breakdown Structure (WBS) depicting in a hierarchical way all the activities to perform to complete the project is viewed vertically. Authors of LSM have preferred the one-dimensional space resource viewed over the horizontal axis, while the time runs along the vertical one.

For convenience, and especially for facilitating the communication of LDSM schedules among people certainly more familiar with Gantt charts, the Gantt’s scheme is preferred: the time running along the
horizontal axis and the one-dimensional space stations spread along the vertical axis.

For compactness and for complying with LSM principles, also shared by the critical chain approach (Newbold, 1998), several non-overlapping activities can be merged on a same row. Both space-constrained activities and non-space-constrained activities are segregated into distinct horizontal strips.

Figure 1 The three types of activities LDSM incl. naming conventions

Figure 2 Optimally calculated vs. CPM calculated LDS diagram – example no. 1

Figure 3 Fragmented LDS diagram; calculated fragmented LDS diagram
The typology of LDSM activities

Basically, LDSM supports three types of activities:

- linear space-constrained activities, resource-constrained or not;
- discrete space-constrained activities, resource-constrained or not; and
- and other activities, discrete, non-space-constrained, resource-constrained or not.

In the LDSM vocabulary, all space-constrained activities are called blocks, while others are called bars, in reference to bar charts, a name used for Gantt charts (cf. Figure 1). Blocks can be of two types: parallelogram-shaped blocks referring to linear activities and

<table>
<thead>
<tr>
<th>Symbol and inverse</th>
<th>Meaning</th>
<th>Relations between dates</th>
<th>Gantt chart representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_j &lt; a_k$</td>
<td>&quot;before&quot;</td>
<td>$SD_j &lt; FD_j &lt; SD_k &lt; FD_k$</td>
<td>$a_j$</td>
</tr>
<tr>
<td>$a_k &gt; a_j$</td>
<td>&quot;after&quot;</td>
<td></td>
<td>$a_k$</td>
</tr>
<tr>
<td>$a_j m a_k$</td>
<td>&quot;meets&quot;</td>
<td>$SD_j &lt; FD_j = SD_k &lt; FD_k$</td>
<td>$a_j$, $a_k$</td>
</tr>
<tr>
<td>$a_k m l a_j$</td>
<td>&quot;overlaps&quot;</td>
<td>$SD_j &lt; SD_k &lt; FD_j &lt; FD_k$</td>
<td>$a_k$, $a_j$</td>
</tr>
<tr>
<td>$a_j s a_k$</td>
<td>&quot;starts&quot;</td>
<td>$SD_j = SD_k &lt; FD_j &lt; FD_k$</td>
<td>$a_j$, $a_k$</td>
</tr>
<tr>
<td>$a_k s l a_j$</td>
<td>&quot;during&quot;</td>
<td>$SD_k &lt; SD_j &lt; FD_j &lt; FD_k$</td>
<td>$a_k$, $a_j$</td>
</tr>
<tr>
<td>$a_j f a_k$</td>
<td>&quot;finishes&quot;</td>
<td>$SD_k &lt; SD_j &lt; FD_j = FD_k$</td>
<td>$a_j$, $a_k$</td>
</tr>
<tr>
<td>$a_k f l a_j$</td>
<td>&quot;equals&quot;</td>
<td>$SD_k = SD_j &lt; FD_j = FD_k$</td>
<td>$a_k$, $a_j$</td>
</tr>
<tr>
<td>$a_j = a_k$</td>
<td>&quot;equals&quot;</td>
<td></td>
<td>$a_j$, $a_k$</td>
</tr>
</tbody>
</table>

Figure 4  CPM calculated LDS diagram – example no. 2

Figure 5  Allen’s temporal logic relations
rectangle-shaped blocks that correspond to discrete activities.

**Linear space-constrained activities (cf. Figure 1a)**

Let \( a_j \) be a linear space-constrained activity, in addition to precedence and resource constraints, such an activity is characterized by a 6-tuple \((x_{j1}, x_{j2}, \theta_j, TS_j, C_j, r_j)\) with:

- \( x_{j1} \): start station of activity \( a_j \)
- \( x_{j2} \): finish station of activity \( a_j \)
- \( \theta_j \): production rate of activity \( a_j \)
- \( TS_j \): temporal span associated with activity \( a_j \)
- \( C_j \): set of predecessors of activity \( a_j \) and \( C_j^{-1} = \emptyset \) if \( a_j \) has no predecessor
- \( r_j \): vector of the resources required for completing activity \( a_j \).

The production rate \( \theta_j \) is the spatial progress foreseen per unit of time: typically, production rates can be expressed in feet/hour, km/week, units/day... \( \theta_j \) is positive if \( a_j \) progresses along \( x \) ascending, i.e. if \( x_{j1} < x_{j2} \); otherwise \( \theta_j \) is negative. The true production rate capacity is based on \( |\theta_j| \) and not on \( \theta_j \).

If the start date \( SD_j \) of activity \( a_j \) is known, the finish date \( FD_j \) of this activity can be obtained as follows:

\[
FD_j = SD_j + \left( x_{j2} - x_{j1} \right) / \theta_j + TS_j.
\]

Figure 6  Forward pass DG of example no. 1

Figure 7  Backward pass DG of example no. 1

It is also important to remark that the definition to predecessor activity in the LDS context differs to the one of predecessors in a CPM context. To illustrate this, let \( a_j \) and \( a_k \) be two discrete activities, such as \( C_j^{-1} = \{a_j\} \). In a CPM context, this means that \( a_k \) cannot start until \( a_j \) has ended. In the LDS environment, precedence constraints generally means that an activity can start, as soon as its predecessor activities has ended in the vicinity of its start station.

Figure 8  Forward date graph with sub-graphs highlighted
Discrete space-constrained activities (cf. Figure 1b)

Such an activity is also characterized by a 6-tuple \((x_{j1}, x_{j2}, \text{DUR}_j, \text{C}_j, r_j)\) with:

- \(x_{j1}\): start station of activity \(a_j\)
- \(x_{j2}\): finish station of activity \(a_j\)
- \(\text{DUR}_j\): duration of activity \(a_j\)
- \(\text{C}_j\): set of predecessors of activity \(a_j\); and \(\text{C}_j = \emptyset\) if \(a_j\) has no predecessor
- \(r_j\): vector of the resources required for completing activity \(a_j\).

A discrete space-constrained activity is a linear space-constrained activity to which \(\theta_l = \infty\) and \(\text{TS}_j = \text{DUR}_j\). In the remaining of this article, we won’t make any distinction anymore between linear and discrete space-constrained activities. They will be simply called block activities.

Non-space-constrained activities (cf. Figure 1c)

In addition to precedence and resource constraints, the duration is sufficient to describe a non-space-constrained activity, henceforth called as bar activity.

Calculated dates

The aim of a critical path analysis is to determine for all the activities of a network the earliest and latest start and finish dates: \(\text{ESD}_j\), \(\text{EFD}_j\), \(\text{LSD}_j\) and \(\text{LFD}_j\). The critical activities are those for which earliest and latest dates are identical, i.e. \(\text{ESD}_j = \text{LSD}_j\) and \(\text{EFD}_j = \text{LFD}_j\). CPM, LSM and LDSM do share the same objective.

An approach for solving a LDS problem

Several approaches can be found to solve a LDS problem. We take two examples to find out where the solving difficulties are located.
<table>
<thead>
<tr>
<th>Activity number and label</th>
<th>Bar act.</th>
<th>Linear</th>
<th>Discrete</th>
<th>Start</th>
<th>Finish</th>
<th>Duration</th>
<th>Perpetual</th>
<th>Predecessors $\Gamma_{j}^{-1}$ (act., type, LAG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 Mark position of equipment and services</td>
<td>●</td>
<td>20 265 23 270</td>
<td>12</td>
<td>2001/10/15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02 Repair floor and drill holes between UA83 and RA83</td>
<td>●</td>
<td>20 265 23 270</td>
<td>10</td>
<td>01, FS, 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03 Re-assemble metallic structure of RR77</td>
<td>●</td>
<td>—</td>
<td>—</td>
<td>4</td>
<td>2002/04/08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04 Install metallic structure in RR77</td>
<td>●</td>
<td>20 265 20 275</td>
<td>2</td>
<td>02, FS, 0 (03, FS, 0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05 Procure electrical equipment for alcoves</td>
<td>●</td>
<td>—</td>
<td>—</td>
<td>8</td>
<td>2002/02/18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>06 Refurbish electrical alcove RE78</td>
<td>●</td>
<td>20 935 20 940</td>
<td>4</td>
<td>02, FS, 0 (05, FS, 0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07 Refurbish electrical alcove RE82</td>
<td>●</td>
<td>22 380 22 385</td>
<td>4</td>
<td>02, FS, 0 (06, FS, 0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>08 Install cable trays and pull AC cables in R78 and R79</td>
<td>●</td>
<td>20 265 21 660</td>
<td>200</td>
<td>04, FS, 0 (06, FS, 0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09 Install cable trays and pull AC cables in R81 and R82</td>
<td>●</td>
<td>21 660 23 270</td>
<td>200</td>
<td>07, FS, 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 Install and weld new water, N, and He pipes</td>
<td>●</td>
<td>20 265 23 270</td>
<td>250</td>
<td>02, FS, 0 (09, FS, 0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 Pull DC cables in R78 and R79</td>
<td>●</td>
<td>20 265 21 660</td>
<td>175</td>
<td>10, FS, 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 Pull DC cables in R81 and R82</td>
<td>●</td>
<td>21 660 23 045</td>
<td>175</td>
<td>10, FS, 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13 Install electrical sub-station in UA83</td>
<td>●</td>
<td>23 045 23 270</td>
<td>3</td>
<td>12, FS, 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14 Pull water cooled cables between UA83 and RA83</td>
<td>●</td>
<td>23 045 23 270</td>
<td>2</td>
<td>13, FS, 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15 Pull signal cables in R78 and R79</td>
<td>●</td>
<td>20 265 21 660</td>
<td>100</td>
<td>11, FS, 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 Pull signal cables in R81 and R82</td>
<td>●</td>
<td>21 660 23 270</td>
<td>100</td>
<td>14, FS, 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: "YYYY/MM/DD" means: activity cannot start before YYYY/MM/DD

Figure 10 Example-description of the activities

<table>
<thead>
<tr>
<th>Activity no.</th>
<th>Sub-activity</th>
<th>$x_{i1}$</th>
<th>$x_{i2}$</th>
<th>$x_{i1}'$</th>
<th>$x_{i2}'$</th>
<th>$\Theta_{i}$</th>
<th>$TS_{j}/DUR_{j}$</th>
<th>Predecessors $\Gamma_{j}^{-1}$ (act., type, LAG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>01.1 20 265 23 270</td>
<td>01.1</td>
<td>20 265 23 270</td>
<td>01.1</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>02</td>
<td>01.2 20 265 23 270</td>
<td>01.2</td>
<td>20 265 23 270</td>
<td>01.2</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>03</td>
<td>01.3 20 265 23 270</td>
<td>01.3</td>
<td>20 265 23 270</td>
<td>01.3</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>04</td>
<td>01.4 20 265 23 270</td>
<td>01.4</td>
<td>20 265 23 270</td>
<td>01.4</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>05</td>
<td>01.5 20 265 23 270</td>
<td>01.5</td>
<td>20 265 23 270</td>
<td>01.5</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>06</td>
<td>01.6 20 265 23 270</td>
<td>01.6</td>
<td>20 265 23 270</td>
<td>01.6</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>07</td>
<td>01.7 20 265 23 270</td>
<td>01.7</td>
<td>20 265 23 270</td>
<td>01.7</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>08</td>
<td>01.8 20 265 23 270</td>
<td>01.8</td>
<td>20 265 23 270</td>
<td>01.8</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>09</td>
<td>02.1 20 265 23 270</td>
<td>02.1</td>
<td>20 265 23 270</td>
<td>02.1</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>10</td>
<td>02.2 20 265 23 270</td>
<td>02.2</td>
<td>20 265 23 270</td>
<td>02.2</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>11</td>
<td>02.3 20 265 23 270</td>
<td>02.3</td>
<td>20 265 23 270</td>
<td>02.3</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>12</td>
<td>02.4 20 265 23 270</td>
<td>02.4</td>
<td>20 265 23 270</td>
<td>02.4</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>13</td>
<td>02.5 20 265 23 270</td>
<td>02.5</td>
<td>20 265 23 270</td>
<td>02.5</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>14</td>
<td>02.6 20 265 23 270</td>
<td>02.6</td>
<td>20 265 23 270</td>
<td>02.6</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>15</td>
<td>02.7 20 265 23 270</td>
<td>02.7</td>
<td>20 265 23 270</td>
<td>02.7</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>16</td>
<td>02.8 20 265 23 270</td>
<td>02.8</td>
<td>20 265 23 270</td>
<td>02.8</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>03</td>
<td>03.1 20 265 23 270</td>
<td>03.1</td>
<td>20 265 23 270</td>
<td>03.1</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>04</td>
<td>04.1 20 265 23 270</td>
<td>04.1</td>
<td>20 265 23 270</td>
<td>04.1</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
<tr>
<td>05</td>
<td>05.1 20 265 23 270</td>
<td>05.1</td>
<td>20 265 23 270</td>
<td>05.1</td>
<td>20 265 23 270</td>
<td>—</td>
<td>4</td>
<td>2001/10/15</td>
</tr>
</tbody>
</table>

Note: "YYYY/MM/DD" means: activity cannot start before YYYY/MM/DD

Figure 11 Example-breakdown of the activities into sub-activities
Example no. 1

Let \( a_j \) and \( a_k \) be two block activities such as

\[
 a_j = (x_1, x_2, \theta_j, TS_j, \Omega, \Omega) \quad \text{and} \quad a_k = (x_1, x_3, \theta_k, TS_k, \{a_j\}, \Omega),
\]

with \( x_1 < x_2 < x_3 \), i.e. \( \theta_j < 0 \) and \( \theta_k > 0 \). Figure 2a shows how these two activities should be scheduled optimally, i.e. as soon as possible. It can be observed that the early finish date \( EFD_j \) of activity \( a_j \) is scheduled later than the early start date \( ESD_k \) of activity \( a_k \).

Figure 2b shows how CPM would have scheduled these two activities. The approach we are proposing in this article for solving such a problem requires a systematic fragmentation of the space-constrained activities along the space axis. This means for instance that activities \( a_j \) and \( a_k \) would be broken down into four sub-activities \( a_j' \), \( a_j'' \), \( a_k' \), \( a_k'' \) such as:

\[
 a_j' = (x_2, x_3, \theta_j, TS_j, \Omega, \Omega), \quad a_j'' = (x_1, x_3, \theta_j, TS_j, \Omega, \Omega), \quad a_k' = (x_1, x_2, \theta_k, TS_k, \{a_j\}, \Omega), \quad a_k'' = (x_2, x_3, \theta_k, TS_k, \{a_j\}, \Omega) \quad \text{(see Figure 3a)}.
\]

The precedence constraints of this set of sub-activities are the following: \( a_j' < a_j'' \), \( a_j'' < a_k' \) and \( a_j'' < a_k'' \) (the ‘\( x < y \)’ stands for ‘\( x \) precedes \( y \)’). The results of a CPM calculation are given in Figure 3b.

In some situations, such schedule can be acceptable. But in most real-world situations, project management practitioners would prefer a continuity in the performing of activity \( a_k \), i.e. baselining \( LSD_k' \) and \( LFD_k' \) instead of \( ESD_k \) and \( EFD_k \). Implementation of Allen’s temporal logic relation leads to a calculation procedure that fulfills real-world requirements, especially the possibility of mixing earliest and latest dates when needed, and by the way responding to the crew/work continuity requirement.

Example no. 2

Let \( a_j \) and \( a_k \) be two activities such as

\[
 a_j = (x_2, x_3, \theta_j, TS_j, \Omega, \Omega) \quad \text{and} \quad a_k = (x_1, x_3, \theta_k, TS_k, \{a_j\}, \Omega),
\]

with \( x_1 < x_2 < x_3 \), i.e. \( \theta_j > 0 \) and \( \theta_k > 0 \).

Three cases shall be distinguished: \( \theta_j < \theta_k \) (cf. Figure 4a), \( \theta_j > \theta_k \) (cf. Figure 4b) or \( \theta_j = \theta_k \). In all cases, \( a_k \) must be scheduled so it just does not interfere with
a. After fragmentation, the sub-activities sharing the same space-strip are scheduled using the following formulae:

\[ SD'_k = SD_j + TS_j + (x_1 - x_2)/\theta_j - (x_3 - x_2)/\theta_k \]  \hspace{1cm} (2)

\[ FD'_k = FD_j + TS_k \]  \hspace{1cm} (3)

if \( \theta_j \leq \theta_k \):

\[ SD'_k = SD_j + TS_j \]  \hspace{1cm} (4)

\[ FD'_k = FD_j + TS_k + (x_1 - x_2)/\theta_k - (x_3 - x_2)/\theta_j \]  \hspace{1cm} (5)

Extended set of precedence constraints

In his temporal algebra, Allen (1983) has proposed eight relations (and 13 if their inverses are considered) for describing all the possible configurations of two temporal intervals. These 13 relations are presented in Figure 5.

The four types of precedence constraints of the precedence method can be defined using Allen’s formalism as follows (let \( a_j \) and \( a_k \) be two activities; symbol \( \sim \) stands for ‘or’):

\[ a_j \sim \text{finish-start} a_k \equiv a_j(m \lor m'') \lor a_k \]

\[ a_j \sim \text{start-start} a_k \equiv a_j(s \lor s'') \lor a_k \]

\[ a_j \sim \text{start-finish} a_k \equiv a_j(m \lor m'') \lor s \lor f \lor s \lor f \lor s \lor f \lor m \lor m' \lor m'') \lor a_k \]

\[ a_j \sim \text{finish-finish} a_k \equiv a_j(f' \lor f'') \lor s \lor f \lor s \lor f \lor s \lor f \lor m \lor m' \lor m'') \lor a_k \]

Let \( a_j \) and \( a_k \) be two space-constrained activities, the Allen’s temporal logic relations that best describe the precedence constraints featured in a fragmented LDS diagram are:

\[ a_j \text{ finish-start precedence constraint} \equiv a_j(m \lor m'') \lor a_k \]

\[ a_j \text{ precedence constraint in a fragmented activity} \equiv a_j(m \lor m'') \lor a_k \]

Equivalent date graph

The use of a date graph (DG) – or point graph (Zaidi, 2001) – is convenient to solve an activity network that involves Allen’s temporal logic relations. A date diagram is a valued digraph in which nodes (vertices) feature the dates to calculate – also called time stamps in some textbooks and articles – and arcs (edges) the time intervals that separate dates that are directly dependent.

In the framework of the LDS problem, two types of arcs are distinguished:

- strong intervals, that could also be called fixed intervals, i.e. the arcs that either correspond to activity duration, or \( m, m', f, f', s, \) and \( s' \) temporal logic relations; and
- weak intervals, that could also be called non-negative intervals, i.e. the arcs that correspond to basic constraints of the precedence method.

Two types of DGs are required to solve this problem: a forward pass DG that is used for calculating early start dates, and a backward pass DG for calculating latest dates. Figure 6 gives the forward pass DG that corresponds to example no. 1, and Figure 7 gives the backward pass DG of the same DS diagram. On both DGs, the dashed arc between \( FD'_k \) and \( SD_k \) depicts a weak interval, while all other arcs are representing strong intervals.

Date calculations

The calculations of the dates of such a date graph are straightforward. This can be performed in three steps.

Step 1

The graph is broken down into sub-graphs by eliminating temporarily all weak arcs (cf. Figure 8). All the resulting sub-graphs should be irreflexive (self-loop free) and acyclic. A sub-graph \( G \) is said to be antecedent to a sub-graph \( G' \), if a weak interval, mapping from \( G \) to \( G' \) has been removed.

Step 2

Early date calculation (forward pass) starts in the sub-graphs that have no antecedent sub-graph, and within these sub-graphs, on the node that has no incoming arc. The early start date of the project (ESD\text{}_{\text{project}}) is assigned to these very first nodes. Formulas such as (1) are then used for propagating calculations in a given sub-graph.

Step 3

When all the dates are calculated in a given sub-graph, propagation can continue over weak intervals. To do that, CPM principles are applied. This is repeated until all the dates are calculated.

In a given sub-graph, backward calculations may sometimes be required for time stamping some of its nodes. This is the case for instance of ESD\text{}_{k}^{s} of example no. 1.
The results of these forward pass calculations are earliest start or finish dates. A similar procedure running backward gives latest start or finish dates; the just calculated earliest finish date of the project (EFD\textsubscript{project}) is assigned to the very last nodes, i.e. those with outgoing arcs. Critical paths and floats can be derived from this information.

Algorithms

The block diagram of Figure 9 gives an overview of the LDSM computational procedure. A more detailed view of the algorithms involved in the schedule calculation is given in the present section.

Seven algorithms need to be run successively to solve a LDS problem. A set of activities including their precedence and resources constraints is sufficient to describe the problem. Algorithm no. 1 aims at finding all the stations associated with the space-constrained activities. Algorithm no. 2 fragments the activity network into sub-activities. Algorithms nos. 3 and 4 generate the forward and backward pass DGs. Algorithms nos. 5 and 6 calculate, respectively, earliest and latest dates of sub-activities. Finally, algorithm no. 7 identifies the critical sub-activities.

For the sake of simplicity, these algorithms are presented in a pseudo-code form. Symbols $\land$ and $\lor$ stand respectively for ‘and’ and ‘or’; statement ‘a $\rightarrow$ b’ means ‘assign b to a’.

Let A be the set of all the activities of the project; each activity $a_j$ is defined as in section 2. Let X be the set of all the stations $x_i$ associated with this problem.

Algorithm no. 1: identification of the stations

10 $X \leftarrow \emptyset$
20 for $j = 1$ to $|A|$ do:
30 if $a_j$ is a block activity then:
40 if $x_{j_1} \not\in X$ then: $X \leftarrow X \cup \{x_{j_1}\}$
50 end if
60 end if
70 $j \leftarrow j + 1$
80 end do
90 order $X$ such as $x_p < x_{p+1}$, $1 \leq p \leq |X|

Algorithm no. 2 provides a tool for fragmenting all the block activities into sub-activities. Bar activities are just duplicated into this new set. Let $A'$ be the set of all the sub-activities of the project.

Note: The set $\Gamma_{j-1}$ of the predecessors of an activity $a_j$ is made of 3-tuples made of:

- the predecessor activity featured by either a 6-tuple (block activity) or a 3-tuple (bar activity);
- the type of precedence constraint

$$\in \{\prec, \preceq, \llhd, \llhd, =, \lor, \land\}$$
- the lag that separates the activities constrained.

Algorithm no. 2: fragmentation into sub-activities

10 $A' \leftarrow \emptyset$
20 order A such as $a_m \prec a_k \prec a_j$; $a_0 \in A \land j \leq k$
30 for $j = 1$ to $|A|$ do:
40 if $a_j$ is a block activity $\& \lor \theta_j > 0$ then:
50 for $n = k$ to $j$ do:
60 $\Gamma_j \leftarrow \emptyset$
70 $\Gamma_j \leftarrow \emptyset$
80 $\Gamma_j \leftarrow \emptyset$
90 otherwise:
100 $\Gamma_j \leftarrow \emptyset$
110 end if
120 end for
130 $A' \leftarrow A' \cup \{(x_k, x_{k+1}, \theta_k, \Gamma_k, \Gamma_{k+1}, \Gamma_{k+1} = P \cup Q, \theta_j)\}$
140 $n \leftarrow n + 1$
150 end do
160 else if $a_j$ is a block activity $\& \lor \theta_j < 0$ then:
170 for $n = k$ to $j$ do:
180 $\Gamma_j \leftarrow \emptyset$
190 $\Gamma_j \leftarrow \emptyset$
200 $\Gamma_j \leftarrow \emptyset$
210 otherwise:
220 $\Gamma_j \leftarrow \emptyset$
230 end if
240 end if
250 $j \leftarrow j + 1$
260 end do

Explanations

Line 30 scans all the activities of A in an ordered way given by line 20. For block activities, two cases are considered: either $\theta > 0$ (line 40) or $\theta < 0$ (line 110). Lines 50 and 120 scan in an ordered way all the stations spread all along the block activity that is being processed. This means that such an activity is broken down into as many sub-activities as there are intermediate stations between the activity’s start and finish stations. P and Q are temporary sets that are used for identifying all the predecessors of the sub-activity that is scrutinized. The following rule is used for
transferring the set of predecessors of an activity $a_j$ to its sub-activities:

- The sub-set of $\Gamma^{-1}_j$ made of activities that are of bar type is transferred to the earliest of the sub-activities of $a_j$. This is the purpose of the last statement of lines 60 and 130.
- The sub-set of $\Gamma^{-1}_j$ made of activities that are of block type is looked at through their corresponding sub-activities. A precedence constraint is set up between possible predecessor sub-activities and the sub-activity that is being processed if the start and end stations are matching. This is the purpose of the last statement of (lines 70 and 140). The first two lines of these statements aim at addressing the problem presented with example no. 2.

Once the predecessors of a sub-activity are defined, they are appended to the set $A'$.

Algorithms nos. 3 and 4 are used for generating the forward and backward pass DGs. Let $G_F$ be the forward pass DG; $G_B = G_{D_5}$ $I_F$ where $D_F$ is the set of the earliest dates to be calculated and $I_F$ is the set of 4-tuples featuring intervals. These 4-tuples are made of four data items: the extremity nodes/dates, the delay that separates these two dates and the type of interval that can be either strong or weak. Because precedence constraints can be of several types, the predecessor set $\Gamma^{-1}_j$ of an activity $a_j$ is made of triplets (as already seen) $(a_k, \sigma_{kj}, LAG_{kj})$, where $a_k$ is an immediate predecessor of $a_j$, $\sigma_{kj}$ denotes the type of precedence relation between $a_k$ and $a_j$, and $LAG_{kj}$ which is used to separate the finish of $a_k$ and the start of $a_j$.

Algorithm no. 3: generation of the forward pass DG

10 $D_F = \emptyset$
15 $I_F = \emptyset$
30 for $j = 1$ to $|A'|$ do:
35 if $ESO_j \notin \emptyset$ then:
40 $D_F = D_F \cup \{ESO_j\}$ end if
45 if $EDF_j \notin \emptyset$ then:
50 $D_F = D_F \cup \{EDF_j\}$ end if
55 $I_F = I_F \cup \{\{ESO_j, EDF_j, \frac{f_j - s_j + LAG_{kj}}{\sigma_{kj}}, \sigma_{kj}, \text{strong}\}\}$
60 if $\Gamma^{-1}_j \neq \emptyset$ then:
65 for $k = 1$ to $|\Gamma^{-1}_j|$ do:
70 if $a_k \subset \{a_j \mid a_j \in \emptyset \}$ then:
75 if $EDF_k \notin \emptyset$ then:
80 $D_F = D_F \cup \{EDF_k\}$ end if
85 if $ESO_k \notin \emptyset$ then:
90 $D_F = D_F \cup \{ESO_k\}$ end if
95 $I_F = I_F \cup \{\{ESO_k, EDF_k, \frac{f_k - s_k - LAG_{kj}}{\sigma_{kj}}, \sigma_{kj}, \text{weak}\}\}$
100 else if $a_k \subset \{a_j \mid \sigma_{kj} = \text{strong}\}$ then:
105 $I_F = I_F \cup \{\{ESO_k, EDF_k, \frac{f_k - s_k - LAG_{kj}}{\sigma_{kj}}, \sigma_{kj}, \text{strong}\}\}$
110 end if
115 $k = k + 1$
120 end do
125 repeat;
130 $S_A \leftarrow \{a_p \mid a_p \in A \land \Gamma^{-1}_j \subseteq S_A \land a_p \notin S_A\}$
135 $S_W \leftarrow \emptyset$
140 if $\Gamma^{-1}_j = \emptyset$ then:
145 $S_D \leftarrow \{d_j \mid d_j \in S_A \land \Gamma^{-1}_j = \emptyset\}$
150 $ED_j = 0$ or the project start date
155 $S_W \leftarrow S_W \cup \{d_j\}$
160 $S_D \leftarrow S_D \cup \{d_j\}$
165 $d_j \in S_D$
170 end if
175 $S_D = \{d_j \mid d_j \in S_W \land ED_j = \min \{ED_k\}\}$
180 $\forall a_p \in S_A$
185 $\forall a_p \in S_A$
190 $S_A \leftarrow S_A \setminus S_A$
195 run SubNetPropagationFW$(a_p)$
200 $\forall a_p \in S_A$
205 while $|S_A| < |A|$
The set $\mathcal{P}_A$ of the activities, i.e. of the sub-graphs, that have already been scheduled, is initialized in line 10. A repeat...while statement (lines 20 and 170) is used for scanning all the activities of $A$.

The set $\mathcal{P}_E$ of the eligible activities is set up in line 30. It is made of the activities of $A$ that have not yet been scheduled, i.e., $a_p \notin \mathcal{P}_A$ but that do have their predecessors scheduled, i.e. $\Gamma^{-1}_p \subseteq \mathcal{P}_A$.

The set $\mathcal{P}_E$ is initialized to empty set with line 40, and reinitialized with every new activity being scrutinized. This set is used as a starting point for date calculations within a given activity.

Two cases may occur for date eligibility, i.e. for identification of the vertices of the forward pass DG that can be calculated. Either the activity has no predecessor (line 50) or it has one (line 90). For both cases, a set $\mathcal{P}_D$ of eligible dates is set up (lines 60 and 100). This set is made of the date vertices that either
have no in-coming edge (case $C_{p^2}^1 = \emptyset$) or that have weak type in-coming edges (case $C_{p^2}^1 \neq \emptyset$). The project start date is assigned to the earliest date $ED_j$ if the parent activity has no predecessor (line 70); otherwise, it is the latest date among those associated to the date vertices of $C_{j^2}^1$ that is assigned to $ED_j$ (line 110). The vertex $d_j$ is then appended to the set $\mathcal{P}_W$ (lines 80 and 120).

The date propagation within a given activity/sub-graph requires a unique starting point; this is the purpose of line 140. When $\mathcal{P}_W$ is made of more than one date, it is the earliest one that is preferred and appended to $\mathcal{P}_S$.

The procedure SubNetPropagationFW is called for propagating the dates of a given activity/sub-graph (line 150). The sets $\mathcal{E}_T$ and $\mathcal{E}_R$ are used for identifying eligible dates (lines 220 and 290), i.e. the dates that are not yet scheduled ($d_j \notin \mathcal{P}_S$) but have adjacent vertices $d_k$ already scheduled ($d_k \in \mathcal{P}_S$). It may happen that such a date has already been scheduled: $d_j \in \mathcal{P}_W$ but $d_j \notin \mathcal{P}_S$. If the newly calculated $ED_j$ is greater than the previously calculated (lines 230 and 300), this latter is superseded by the new one (lines 240 and 310). Otherwise, the procedure SubNetPropagationFW is run again with a new starting point.

Algorithm no. 6: calculation of the latest dates

```
10 $\mathcal{P}_A \leftarrow \emptyset$
20 repeat:
30 $\mathcal{E}_A \leftarrow \{ a_p \mid a_p \in A \land \Gamma_{p^2} \subseteq \mathcal{P}_A \land a_p \notin \mathcal{P}_A \}$
40 $\mathcal{P}_W \leftarrow \emptyset$
50 if $\Gamma_p = \emptyset$ then:
```
Finally, algorithm no. 7 calculates the critical activities of the project. The critical activities of an activity network are defined as activities for which earliest dates and latest dates coincide. This means that one cannot delay one of these activities without delaying proportionally the completion date of the project.

Explanations

Line 10 picks up a date \(d_p\), the earliest, among all the dates of an activity \(a_p\). The total float \(TF_p\) is calculated as the difference between the latest and the earliest dates that have been calculated for \(d_p\) (line 20). The set \(AC\) of the critical activities of the project is made of all the activities that have a total float equal to zero.

Example

A wide range of projects use linear scheduling approaches: construction of highways and railways, repavement of runways, excavation of tunnels, installation of pipelines, etc. The construction of large particle accelerators also involves such scheduling approaches. The LHC (Large Hadron Collider) that is being built near Geneva, Switzerland, uses at coordination level a scheduling approach that is very similar to the LDSM. The one used for that project is slightly more complex because it also integrates a line-of-balance mechanism.

The LHC will provide particle physics community with a tool to reach the energy frontier above 1 TeV. To deliver 14 TeV proton-proton collisions, it will operate with about 1700 cryo-magnets using NbTi superconductors cooled at 1.8 K. These cryo-magnets will be installed in the 27-km long, 100-m underground ring tunnel that was excavated 15 years ago for housing the LEP (Large Electron-Positron) accelerator. After a decade of research and development, the LHC main components are being manufactured in industry.

The installation works have started with the refurbishing of the existing infrastructures. The installation schedule of the LHC project consists of about 2000 work units. The construction of new civil works has started in 1998. The particle physics community expects to have this new accelerator installed and fully commissioned by mid-2007. For the sake of simplicity, we have limited our example to a small sub-set of this large-scale project: the installation of the general services of sectors 7–8 (one-eighth of the main ring).

The LHC co-ordination schedule can be seen from www.cern.ch. A summary of this schedule is provided in the Appendix of this paper.
space-constrained activities; all their $\theta_i$ have been set to positive values in order to provide a pre-optimization of the schedule.

The seven algorithms presented in the previous section have been run on this sub-set of activities; the results of the computations are given in Figures 11–17. The result of algorithm no. 2, i.e. the corresponding sub-activities is given in Fig. 11. The forward and backward pass DGs, as generated using algorithms 3 and 4, are given in Figures 12 and 13. The results of the computations obtained using algorithms 5, 6 and 7 are given in Figures 14 and 15. Two Gantt charts are proposed, one featuring the earliest dates for these 16 activities (Figure 16), another with the latest dates (Figure 17).

Based on this case experience, this approach improves both the analysis of the activity network and, because of a better visualization, ease the communication of the schedule. Rescheduling and the dynamic impact of changes can be better understood.

<table>
<thead>
<tr>
<th>act. no.</th>
<th>Sub-act.</th>
<th>Vertex</th>
<th>Earliest start date ESD$_j$</th>
<th>Latest start date LSD$_j$</th>
<th>Vertex</th>
<th>Earliest finish date EFD$_j$</th>
<th>Latest finish date LFD$_j$</th>
<th>Total float TF$_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.2</td>
<td>061</td>
<td>Thu 2002/06/13</td>
<td>Thu 2002/06/13</td>
<td>062</td>
<td>Fri 2002/07/19</td>
<td>Fri 2002/07/19</td>
<td>Ow.</td>
<td></td>
</tr>
<tr>
<td>10.3</td>
<td>063</td>
<td>Mon 2002/07/01</td>
<td>Mon 2002/07/01</td>
<td>064</td>
<td>Fri 2002/07/19</td>
<td>Fri 2002/07/19</td>
<td>Ow.</td>
<td></td>
</tr>
<tr>
<td>10.4</td>
<td>065</td>
<td>Mon 2002/07/01</td>
<td>Mon 2002/07/01</td>
<td>066</td>
<td>Fri 2002/08/09</td>
<td>Fri 2002/08/09</td>
<td>Ow.</td>
<td></td>
</tr>
<tr>
<td>10.5</td>
<td>067</td>
<td>Mon 2002/07/22</td>
<td>Mon 2002/07/22</td>
<td>068</td>
<td>Fri 2002/08/30</td>
<td>Fri 2002/08/30</td>
<td>Ow.</td>
<td></td>
</tr>
<tr>
<td>10.6</td>
<td>069</td>
<td>Mon 2002/08/12</td>
<td>Mon 2002/08/12</td>
<td>070</td>
<td>Fri 2002/08/30</td>
<td>Fri 2002/08/30</td>
<td>Ow.</td>
<td></td>
</tr>
<tr>
<td>11.2</td>
<td>077</td>
<td>Thu 2002/07/04</td>
<td>Tue 2002/08/27</td>
<td>078</td>
<td>Mon 2002/08/12</td>
<td>Fri 2002/10/04</td>
<td>8w.</td>
<td></td>
</tr>
<tr>
<td>11.3</td>
<td>079</td>
<td>Tue 2002/07/30</td>
<td>Mon 2002/09/23</td>
<td>080</td>
<td>Mon 2002/09/12</td>
<td>Fri 2002/10/04</td>
<td>8w.</td>
<td></td>
</tr>
<tr>
<td>11.4</td>
<td>081</td>
<td>Tue 2002/07/30</td>
<td>Mon 2002/09/23</td>
<td>082</td>
<td>Tue 2002/09/10</td>
<td>Fri 2002/11/01</td>
<td>8w.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>083</td>
<td>Mon 2002/08/12</td>
<td>Mon 2002/08/12</td>
<td>084</td>
<td>Fri 2002/09/20</td>
<td>Fri 2002/09/20</td>
<td>Ow.</td>
<td></td>
</tr>
<tr>
<td>12.3</td>
<td>087</td>
<td>Mon 2002/09/09</td>
<td>Mon 2002/09/09</td>
<td>088</td>
<td>Fri 2002/10/18</td>
<td>Fri 2002/10/18</td>
<td>Ow.</td>
<td></td>
</tr>
<tr>
<td>15.3</td>
<td>097</td>
<td>Mon 2002/09/02</td>
<td>Mon 2002/10/28</td>
<td>098</td>
<td>Mon 2002/09/30</td>
<td>Fri 2002/11/22</td>
<td>8w.</td>
<td></td>
</tr>
<tr>
<td>15.4</td>
<td>099</td>
<td>Tue 2002/09/03</td>
<td>Mon 2002/10/28</td>
<td>100</td>
<td>Tue 2002/11/19</td>
<td>Fri 2002/01/10</td>
<td>8w.</td>
<td></td>
</tr>
<tr>
<td>16.3</td>
<td>105</td>
<td>Tue 2002/10/15</td>
<td>Tue 2002/10/15</td>
<td>106</td>
<td>Fri 2002/12/17</td>
<td>Fri 2002/12/17</td>
<td>Ow.</td>
<td></td>
</tr>
<tr>
<td>16.4</td>
<td>107</td>
<td>Mon 2002/12/02</td>
<td>Mon 2002/12/02</td>
<td>108</td>
<td>Fri 2002/01/10</td>
<td>Fri 2002/01/10</td>
<td>Ow.</td>
<td></td>
</tr>
</tbody>
</table>

**Conclusion**

For some specific types of construction projects, specifically the ones that require activities with linear development and/or repetitive activities, the classical project scheduling approaches, based on the CPM or PDM, are not the most suitable. Therefore, few specific approaches were developed to cope with these types of projects; but none of them addresses specifically the mixture of linear type activities together with non-space-constrained activities, in a single resource-driven scheduling system.

With the linear-discrete scheduling model, the linear problem is transformed into a standard resource-constrained project scheduling problem, onto which a wide range of algorithms can be applied, especially those for optimizing the allocation of resources (see, for example, Hartmann, 2001), including the critical chain approach (Newbold, 1998).

One of the key features of the methodology is the implementation of a sub-set of Allen’s temporal...
logic relations, used to avoid adverse discontinuities. If we look at the past, activity-on-arrow diagrams were first proposed in the late 1950s, for finding solutions to project activity networks. These networks could model only finish–start precedence constraints. Activity-on-node diagrams offer more flexibility from that point of view: in addition to FS precedence constraints, the project scheduler has the possibility to model SS, SF or FF constraints, with or without lag. The Allen’s temporal logic relations offer 13 additional configurations that can be very useful to project schedulers, especially to those who are in charge of large-scale speculative project schedules, and who have not the possibility to describe thousand of activities. The scheduling of space-constrained projects are among the ones that can benefit from using precedence constraints mixed with Allen’s relations, but such a scheduling approach can be beneficial to many other project contexts.

Such an integrated method provides project management practitioners with better means to handle time variation in their tasks. In doing this, it enables project managers to better master overall time usage in the project while at the same time giving them better control to meet project milestones and, most importantly, to reduce project lead-times. Lead-time reduction correlates with a better quality, a better productivity, and by the way to projects with higher return on investment.
References


