KEYWORDS construction project scheduling · linear scheduling method (LSM) · repetitive scheduling method (RSM) · unified project scheduling system (UPSS)

ABSTRACT

There are projects for which the classical Critical Path method (CPM) or Precedence Diagramming Method (PDM) scheduling techniques are not the most suitable. Several alternative approaches have been developed over the last two decades to cope with the scheduling requirements of construction projects that are made of either repetitive activities or activities that have a linear development. Construction projects, and more specifically large-scale construction endeavors, are often composed of a mixture of repetitive activities, linear activities and more conventional project activities. The approach that is proposed in the present article enables construction practitioners to consider these three types of activities in a unique scheduling model — called Unified Project Scheduling System (UPSS) — that has a level of complexity similar to the one of CPM or PDM. UPSS has been designed so that most of the resource-constrained project scheduling algorithms can be used.

MERGING PDM, RSM AND LSM SCHEDULING APPROACHES

into a single construction project scheduling system

Pierre Bonnal
CERN, Geneva, Switzerland
pierre.bonnal@cern.ch

Mathieu Baudin
CERN, Geneva, Switzerland, Product design and Innovation Laboratory, ENSAM ParisTech, Paris, France
mathieu.baudin@cern.ch

Jurgen De Jonghe
CERN, Geneva, Switzerland
jurgen.de.jonghe@cern.ch

INTRODUCTION

A project can be defined as a unique endeavor, composed of activities characterized by their uniqueness. Such a definition is certainly true for small- or medium-size projects, but not so true for large-scale construction projects that are often made-up of a mixture of one-of-a-kind activities, of repetitive activities, and of activities that have a linear development.

Few examples: the construction of a residential area consisting of dozens of more or less identical houses; the construction of a hydroelectric power plant that requires several identical hydraulic turbines and power generators; the construction of an industrial facility in which several more or less similar production lines are to be installed; the digging of a railway tunnel that progresses linearly over quite long distances; the repaving of a motorway; the construction of a particle accelerator that is made of electromagnets and other components that are manufactured in small- or medium-size series production.

Implementing traditional project planning and scheduling techniques for these types of projects can quickly result in heavy systems to deal with. To illustrate this, let's consider the aforementioned residential area construction project; and assume that for each of the said 100 houses to be built, the masonry works are made up of four activities. This leads to a project activity network diagram made of 400 activities. After taking into account all the other activities, the resulting schedule can easily end up with tens of thousands of activities, and every professional project planner knows that such a schedule is extremely difficult to handle.

The following example is at the origin of the techniques discussed herein: the construction of a particle accelerator such as the Large Hadron Collider *(LHC)* that is being installed underground, at CERN, near Geneva, Switzerland *(see Appendix 1 for more details)*. More than 1600 cryo-magnets need to be manufactured before being installed in the 27-km LHC underground main ring tunnel. Because of the sophistication of these extremely complex elements, hundreds of operations are needed to outline the assembly of a single unit. As a consequence, over one hundred thousand activities would be needed to set up a project activity network diagram in the strictest application of the Critical Path Method *(CPM)* or of the Precedence Diagramming Method *(PDM)* principles. One cannot reasonably consider to proceed in such a way!

Even if this quantitative aspect is of prime importance in terms of scheduling a large-scale project, it is not the only one related to the use of the CPM or PDM methodologies: for the sake of an efficient use of the available resources, project planners seek optimal schedules that ensure workload continuity between repetitive activities. When the CPM or PDM methodologies are used, this can be achieved by considering the earliest dates calculated for some activities and the latest ones for other activities. Even if these scheduling methodologies allow doing so, these professional planners know this is not efficient. This assessment is not new, it has been made my many authors, and several of them have proposed alternative approaches to address these scheduling difficulties. The way the problem has been addressed can be divided into two types: the approaches that provide an answer to repetitive activities, and those that provide an answer to projects made-up of activities with a linear development. The following are some of the various approaches that address the repetitive activity problem:

- Line of Balance (LOB) technique (Lumbsden (1968), O'Brien (1969), Carr & Meyer (1974), Halpin & Woodhead (1976), Harris & Evans (1977), Arditi & Albulak (1986), Lutz & Halpen (1992));
- Construction Planning Technique (CPT) (Peer (1974), Selinger (1980));
- Vertical Production Method (VPM) (O'Brien (1975), Barrie & Paulson (1978));
- Time-Location Matrix Model (Birrell (1980));
- Time Space Scheduling Method (Stradal & Cacha (1982));
- SYRUS (Arditi & Psarros (1987), Psarros (1987));
- Disturbance Scheduling (Whitman & Irwig (1988));
- Horizontal and Vertical Logic Scheduling (HVLS) (Thabet & Beliveau (1994));
- Repetitive construction scheduling (El-Rayes & Moselhi (2001a), El-Rayes (2001b), El-Rayes, Ramanathan & Moselhi (2002));
- A repetitive scheduling model with sharable resource constraints (Leu & Hwang (2001)).

The following approaches address the linear activity problem:

- Time Versus Distance Diagrams (Gorman (1972));
- Linear Balance Charts (Barrie & Paulson (1978));
- Velocity Diagrams (Dressler (1980));
- Linear Scheduling Method (LSM) (Johnston (1981), Chrzanowski & Johnston (1986), Handa & Barcia (1996), Russell & Casselton (1988), Eldin & Senouci (1994), Eldin & Senouci (2000), El-Rayes & Moselhi (1998), Harmelink & Rowings (1998), Harmelink (2001));
- Resource BALanced Scheduling (BAL) (Hegazi, Moselhi & Fazio (1993));
- Linear-Discrete Scheduling Method (LDSM) (Bonnal et al. (2005)).

Although these approaches and methodologies have been developed to answer specific requirements *(Harris & Ioannou (1998))*, they all share similar characteristics: reducing the size of the portfolio of activities to manage and ensuring work load continuity between repetitive activities or linear activities.

In the present article we propose an enhancement to the construction scheduling problem by integrating in a single activity network: repetitive activities, linear activities and one-of-a-kind activities. Section 1 presents the outline of the Precedence Diagramming Method; Section 2 provides the basic concepts of the Repetitive Scheduling Method approach, together with key definitions; those of the Linear Scheduling Method in section 3. Section 4 presents the merging of the three approaches into a unique system, henceforth called Unified Project Scheduling System *(UPSS)*, prior to some conclusions.

1. PDM Basics and Definitions

Project activity networks can be analyzed by two means: one regards linear or integer programming and the other graph theory and propagation algorithms. The latter is known to be simpler for calculating the start and finish dates of the activities of a PDM activity-on-node network. In such a framework, the project is modeled as a valued digraph **G=**(A,U) where j is the set of the activities j of the project and U is its set of precedence constraints *(*j, k*)*; this set is defined as U = {(j, k | precedes k}.

The activities j are characterized by some additional information: their duration $\mathbf{D}_j$ *(a.k.a. processing time)*, and their resource vector $\mathbf{r}_j$ required for them to proceeded. Γ_j^{-1}, a set of predecessors of an activity j, can easily be derived from U for each activity j of the project. The 3-tuple $(\mathbf{D}_j; \Gamma^{-1}; \mathbf{r}_j)$ is sufficient to define the PDM activities.

The analysis of the activity network leads to the calculation of the following for each activity j of the project: its earliest start ES_j and finish EF_j dates, its latest start LS_j and finish LF_j dates, its so-called total TF_j and free FF_j floats or slacks.

Two dummy activities are generally added to the set A: they can be called 0 and *n*+1 with *n*=|A| or α and ω . These two dummy activities are zero-duration activities. α features the project start date; all activities with $\Gamma_j^{-1} = \emptyset$ have α as a predecessor. ω features the project finish date; all activities with $\Gamma_j = \emptyset$ have ω as a successor.

In the framework of the precedence diagramming method, there are four types of constraints between activities:

- Finish-start: $j_{FS}k$ means that activity k can start as soon as activity j is completed; this is the most common type of precedence constraints that can be found in a project activity network;
- Start-start: $j_{SS}k$ means that activity k can start as soon as activity j has started;
- Start-finish: $j_{SF}k$ means that activity k can finish as soon as activity j has started;
- Start-start: $j_{FF}k$ means that activity k can finish as soon as activity j is completed.

In this framework, the set of predecessors Γ_k^{-1} of an activity k is made of 3-tuples $(j, \sigma_{jk}, \mathbf{L}_{jk})$ where j is a predecessor activity of activity k, σ_{jk} is the precedence relationship with $\sigma_{jk} \in \{FS, SS, SF, FF\}$, $\mathbf{L}_{jk}$ is the lag between activities j and k *(positive or negative)*.

Temporal constraints on an activity j can be added by the means of precedence constraints between α and j or between j and ω, with an appropriate lag.

Schedule dates are calculated by running an algorithm that consists mainly of two parts. The earliest start dates are calculated during the so-called forward pass, by propagation from α to ω. The latest finish dates are calculated during the so-called backward pass, still by propagation, but from ω to α. The propagation principle is straightforward: as soon as the dates ES_j of the activities of the set Γ_k^{-1} are calculated, ES_k can be calculated, and as soon as the dates LF_k of the activities of the set Γ_j are calculated, LF_j can be calculated.

Earliest start dates ES_j are calculated as follows:

$$ES_k = \begin{cases} ES_\alpha \text{ if } \Gamma_k^{-1} = \emptyset \\ \max\limits_{j \in \Gamma_k^{-1}} \left\{ \begin{matrix} ES_j + D_j & & + L_{jk} & \text{if } \sigma_{jk} = FS \\ ES_j & & + L_{jk} & \text{if } \sigma_{jk} = SS \\ ES_j & - D_k & + L_{jk} & \text{if } \sigma_{jk} = SF \\ ES_j + D_j & - D_k & + L_{jk} & \text{if } \sigma_{jk} = FF \end{matrix} \right\} \text{otherwise} \end{cases}, \forall k \quad (1)$$

Earliest finish EF_k dates, are derived from ES_k as follows: $EF_k = ES_k + D_k$, ∀k.

Latest finish dates LF_j are calculated as folows:

$$F_j = \begin{cases} LF_\omega \text{ if } \Gamma_j = \emptyset \\ \min\limits_{k \in \Gamma_j} \left\{ \begin{matrix} LF_k - D_k & & - L_{jk} & \text{if } \sigma_{jk} = FS \\ LF_k - D_k & + D_j & - L_{jk} & \text{if } \sigma_{jk} = SS \\ LF_k & + D_j & - L_{jk} & \text{if } \sigma_{jk} = SF \\ LF_k & & - L_{jk} & \text{if } \sigma_{jk} = FF \end{matrix} \right\} \text{otherwise} \end{cases}, \forall j \quad (2)$$

Latest start dates LS_j are derived from LS_j as follows: $LS_j = LF_j - D_j$, ∀j.

The total and free floats are calculated as follows:
$TF_j = LF_j - D_j - ES_j$, ∀j and $FF_j = min\{ES_j\} - D_j - ES_j$, ∀j.
All the activities that have $TF_j = 0$ are critical activities.

2. Repetitive Scheduling Basics and Definitions

The implementation of a RSM system that calculates, for all the activities of a project made-up of repetitive activities, their earliest start and finish dates is quite simple. While date calculations of a PDM networks use either graph theory or linear programming backgrounds *(Elmaghraby (1977))*, the date calculations for the RSM problems are mainly graphical *(see Arditi et al. (2002)* or Harris & Ioannou *(1998)* for instance). The time is drawn on the X-axis while the number of units to build, manufacture, assemble, test and etc. are featured on the Y-axis. Repetitive activities are presented either as slanted lines *(***Figure 1a***)* or as parallelograms *(***Figure 1b***)*.

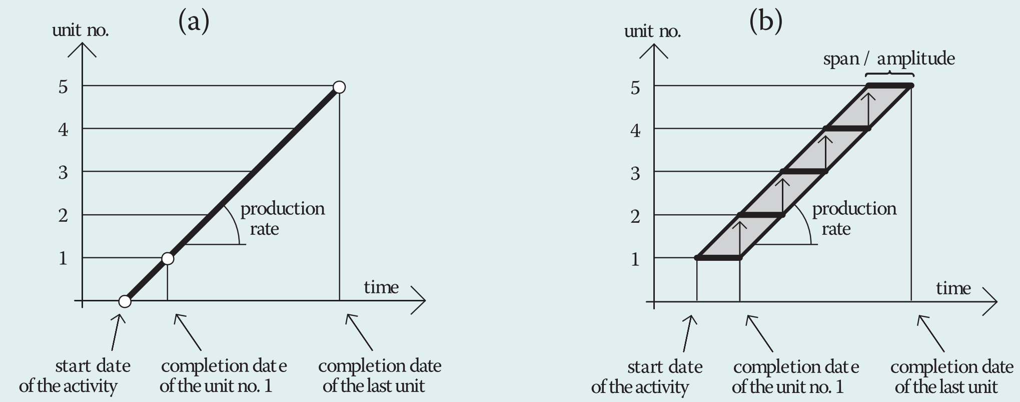


FIGURE 1. *Repetitive activity drawn as a slanted line (a) or as a parallelogram (b).*

While "standard" project activities are characterized by their duration and their immediate preceding activities, repetitive activities — also called multi-unit activities — are characterized by a production rate and by the number of units to produce. For progress control, it may be wise to express these parameters in term of the physical units to produce, but for integration reasons, it is necessary to have these parameters expressed in "equivalent end product." Let's take an example to illustrate this. Approximately 1240 dipole cryo-magnets are needed to construct the Large Hadron Collider *(being constructed near Geneva, Switzerland)*. Each of them has two poles; each pole has two coils; each coil has an inner layer and an outer layer; the inner layer contains a unit length of superconducting cable of a certain type and the outer layer of a different unit length of superconducting cable of another type. **Figure 2** illustrates a "unit network" for the manufacturing of some components and the assembly of a dipole cryo-magnet; **Table 1** summarizes for each of these activities the quantities and units to be used as a metric for physical progress and the ones needed for scheduling. The right-most column gives the production rates. Finally, **Figure 3** shows the RSM graph. **Figure 3**: Repetitive Schedule for the Production of LHC Dipole Cryo-magnets. *(note that this is the parallelogram formalism that has been used).* This figure shows some of the activities scheduled so that the activity starts right after the first unit of its preceding activity is completed. For other activities, depending on the production rate, coincidence is made with the last unit produced.

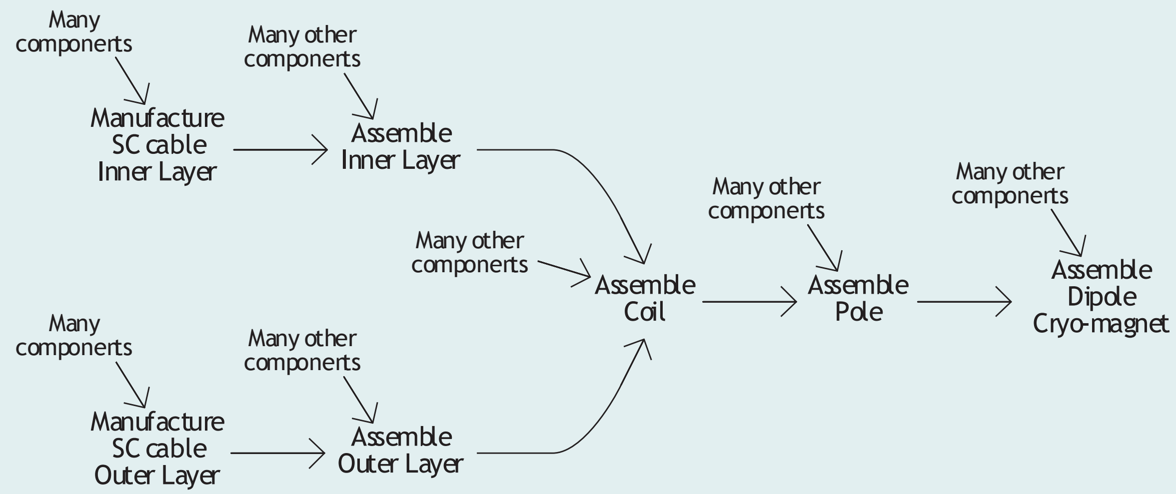


FIGURE 2. *Simplified "unit network" for the Production of LHC Dipole Cryo-magnets.*

TABLE 1. *Production Metrics for the Production of LHC Dipole Cryo-magnets.*

Components	*Physical Progress Metrics*	*Schedule Analysis Metrics*
Dipole Cryo-magnets	1240 u.	1240 u.
Poles	2480 u.	1240 u. e.e.p.
Coils	4960 u.	1240 u. e.e.p.
Inner Layers	4960 u.	1240 u. e.e.p.
Outer Layers	4960 u.	1240 u. e.e.p.
SC Cable Inner Layer	4960 u.l.	1240 u. e.e.p.
SC Cable Outer Layer	4960 u.l.	1240 u. e.e.p.

u. = units ; u.l. = unit length ; e.e.p. = equivalent end product

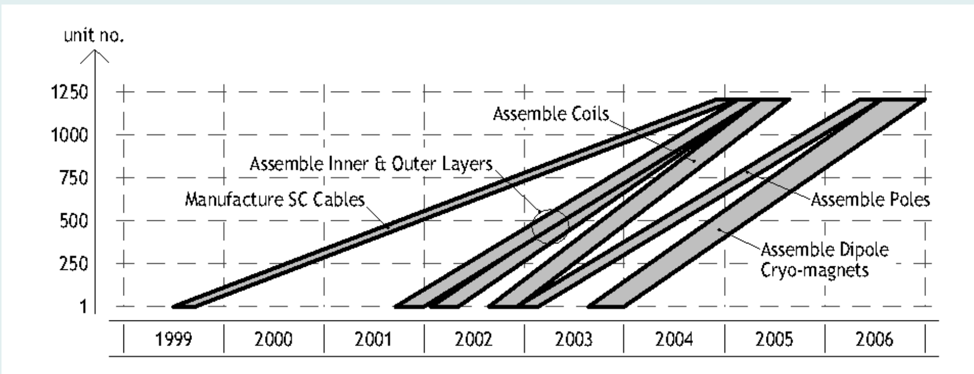


FIGURE 3. *Repetitive Schedule for the Production of LHC Dipole Cryo-magnets.*

A definition of the PDM activity was given in the previous section; the 3-tuple $(D_j, \Gamma_j^{-1}, \mathbf{r}_j)$ is sufficient to define them. The RSM activities, i.e., repetitive activities, can also be defined by a 3-tuple similar to that of the PDM activities: $((TS_j, Q_j, \theta_j)\ \Gamma_j^{-1}, \mathbf{r}_j)$ where:

- TS_j is the temporal span of activity j, i.e., the time required to produce 1 unit;
- Q_j is the total quantity of units to produce; it is generally normalized in "e.e.p." (equivalent end-product);
- θ_j is the production rate, generally expressed in number of units that can be produced per period of time: e.g., unit/week, unit/day...

If the start date S_j of activity j is known, its end date F_j can be calculated using the following expression: $F_j = S_j + (Q_j - 1)/\theta_j + TS_j$.

It can be noted that when $\theta_j \rightarrow \infty$ *(even if this has no physical meaning)*: $F_j = S_j + TS_j = S_j + D_j$.

Figure 4 shows a visual definition of all these parameters.

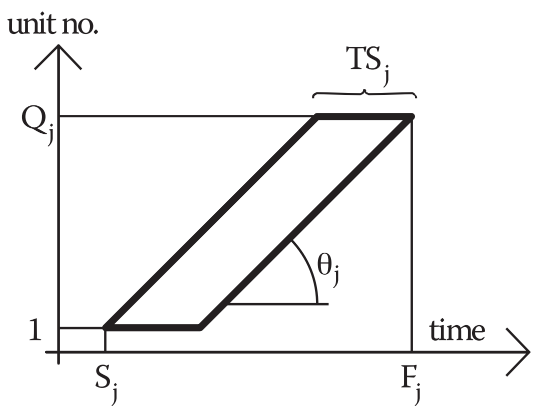


FIGURE 4. *Naming conventions for repetitive activity j.*

Let's use an example to find out where scheduling difficulties are located. Let's consider two repetitive activities j and k, with $\Gamma_j^{-1} = \emptyset$ and $\Gamma_k^{-1} = \{(j, FS, 0)\}$. One could schedule them as if they were PDM activities. To do so, D_j and D_k must be known: $D_j = (Q_j - 1)/\theta_j + TS_j$ and $D_k = (Q_k - 1)/\theta_k + TS_k$.

Figure 5a presents a possible resulting schedule analyzed according to the standard PDM approach, assuming a finish-start constraint. One can see that it is not optimal. The schedules presented in **Figure 5b** *(case featuring $\theta_j > \theta_k$)* or in **Figure 5c** *(case featuring $\theta_j < \theta_k$)* give optimal solutions.

ES_k and LF_j can be calculated from ES_j and LF_k as follows:

$$ES_k = \begin{cases} ES_j + TS_j + L_{jk} & \text{if } \theta_j \geq \theta_k \\ ES_j + TS_j + \dfrac{(Q_j - 1)}{\theta_j} - \dfrac{(Q_k - 1)}{\theta_k} + L_{jk} & \text{if } \theta_j < \theta_k \end{cases} \quad (3)$$

$$LF_j = \begin{cases} LF_k - TS_k - L_{jk} & \text{if } \theta_j \leq \theta_k \\ LF_k - TS_k - \dfrac{(Q_k - 1)}{\theta_k} + \dfrac{(Q_j - 1)}{\theta_j} - L_{jk} & \text{if } \theta_j > \theta_k \end{cases} \quad (4)$$

These formulae are applicable when two or more repetitive activities are constrained by a finish-start constraint. By considering the special case where Q = 1 we fall back to the standard PDM formulae *(1)* and *(2)* with $D_k = TS_k$.

As can be easily derived from **Figure 5b**, in the case of RSM the finish-start constraint can be modeled in "standard PDM" as a start-start constraint with a lag equal to the time span of the preceding activity. Similarly, when the production rate of the preceding activity is lower than the succeeding activity *(as in **Figure 5c**)* the finish-start constraint can be modeled as a finish-finish constraint, with lag equaling the time span of the succeeding activity. While this does yield optimal scheduling results, it has the disadvantage that the resulting precedence constraints are introduced for reasons which do not reflect reality *(which in this case is really the finish-start constraint)*. Also if, for some reason, the production rate of one of the activities changes, it may be needed to review the artificial precedence constraints. For all these reasons, it does not seem sound to introduce basic PDM constraints to reflect constraints between pairs of repetitive activities.

3. Linear Scheduling Basics

The projects related to the linear scheduling method are projects that mainly consist of linear activities. Harmelink & Rowling *(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 projects that typically belong to the family of linear development projects.

Typically, linear activities consume a spatial resource, which is the physical location where they are carried out. This spatial resource is a renewable resource: as soon as the work is completed in a given area, the place is freed for another activity.

There are two types of space-constrained activities: the linear space-constrained activities, and the discrete space-constrained activities. The linear space-constrained activities are in fact those activities that consume a spatial resource and that progress along a path. The discrete space-constrained activities are a special case of the latter: these are activities that typically occupy a physical area, but that do not progress along a path.

Linear space-constrained activities can be graphically presented either as slanted lines (**Figure 6a**) or as parallelograms (**Figure 6b**). Because discrete space-constrained activities benefit by being represented as rectangles, the parallelogram representation is preferred.

In our preferred linear scheduling vocabulary, all space-constrained activities are called "blocks." The blocks can then be of two types: parallelogram-shaped blocks referring to linear activities and rectangle-shaped blocks that correspond to discrete activities.

Linear space-constrained activities. (**Figure 7a**). Let j be a linear space-constrained activity, in addition to precedence and resource information, such an activity is characterized by a 3-tuple $((x_j, y_j, TS_j, \theta_j), \Gamma_j^{-1}, r_j)$ where:

- x_j is the start station of activity j

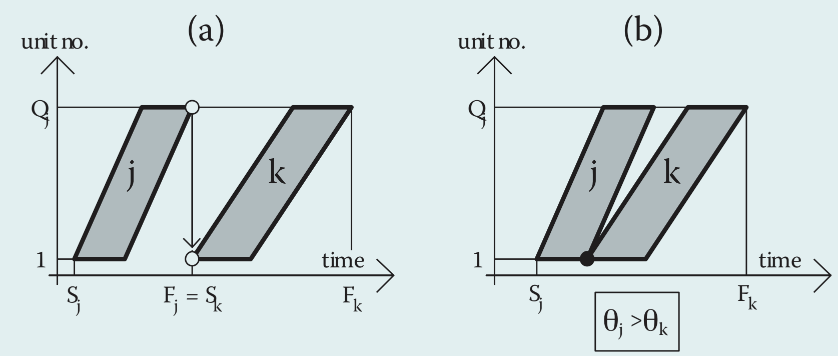


FIGURE 5. *Scheduling of two repetitive activities j and k.*

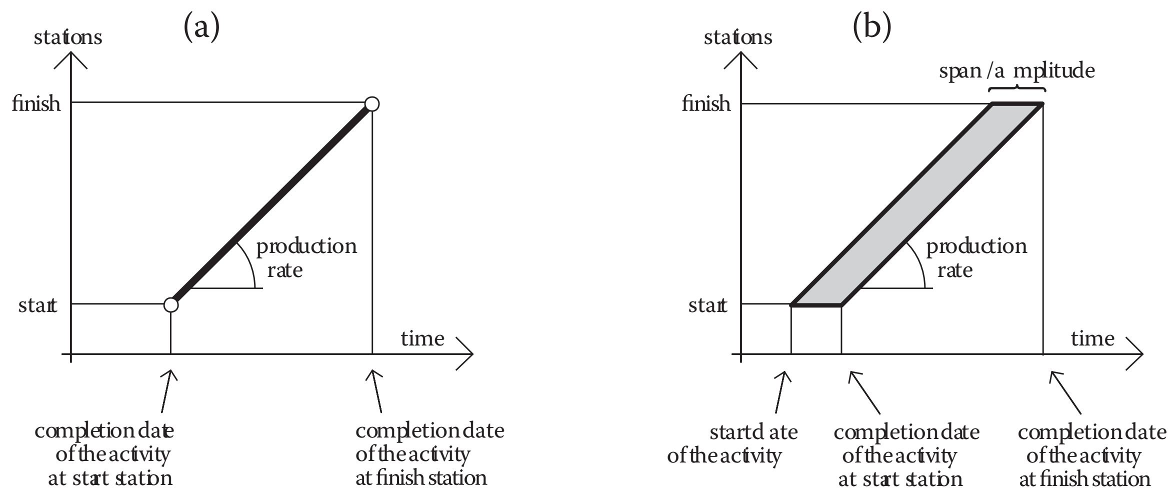


FIGURE 6. *Linear activity drawn as a slanted line (a) or as a parallelogram (b).*

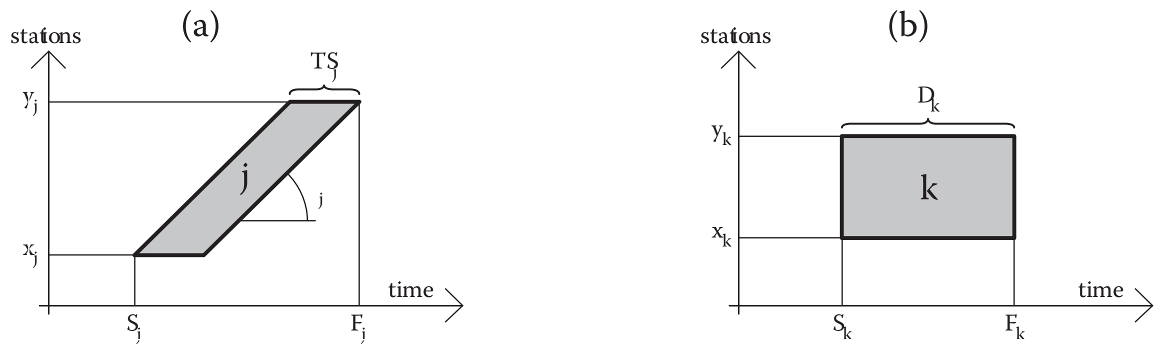


FIGURE 7. *Naming conventions for linear space-constrained activity j and discrete space-constrained activity k.*

- y_j is the finish station of activity j
- TS_j is the temporal span of activity j
- θ_j is the production rate of activity j; which is the spatial progress foreseen per unit of time: typically, production rates can be expressed in feet/hour, km/week, units/day... θ_j is positive if j progresses along the path, i.e., if $x_j < y_j$, otherwise θ_j is negative. With such a convention, it should be noted that the true production rate capacity is based on $|\theta_j|$ and not on θ_j.

If the start date S_j of activity j is known, the finish date F_j of this activity can be obtained as follows: $F_j = S_j + (y_j - x_j)/\theta_j + TS_j$.

It should also be stated that the definition of predecessor activity in a linear scheduling context differs from the one of predecessors in a CPM or PDM context. To illustrate this, let j and k be two activities, such as $\Gamma_k^{-1} = \{j\}$. In a CPM or PDM context, this means that k cannot start until j has not ended. In a linear scheduling environment, precedence constraints generally mean that an activity can start, as soon as its predecessor activities have ended in the vicinity of its start station.

Discrete space-constrained activities. — (**Figure 7***b*). These are a special case of the latter; they are also characterized by a 3-tuple $((x_j, y_j, D_j, \infty), \Gamma_j^{-1}, r_j)$ where:

- x_j and y_j are the end stations of activity j
- D_j is the duration of activity j: $F_j = S_j + D_j$

A discrete space-constrained activity is a linear space-constrained activity without production rate *(mathematically* $\theta_j = \infty$*)*, and $TS_j = D_j$. In the remainder of this article, we will use the term "block activities" to denote linear and discrete space-constrained activities.

Linear scheduling. Several approaches can be found to solve a linear scheduling problem. As an example, let j and k be two block activities such that $j = ((x_4, x_2, TS_j, \theta_j), \varnothing, \varnothing)$ and $k = ((x_1, x_3, TS_k, \theta_k), \{j\}, \varnothing)$, with $x_1 < x_2 < x_3 < x_4$, i.e. $\theta_j < 0$ and $\theta_j > 0$.

Figure 8a shows how these two activities should be scheduled optimally *(assuming the finish-start constraint)*. It can be observed that the finish date F_j of activity j is scheduled later than the start date S_k of activity k. **Figure 8b** shows how "standard PDM" would have scheduled these two activities. The result of **Figure 8b** is not optimal.

Earliest and latest date calculations. It is rather laborious to find the formulae corresponding to expressions *(1)* and *(2)* of PDM earliest and latest date calculations! Actually, one shall consider the 117 different combinations of xj, yj, xk, yk, θj and θk *(see Figure 10)*. By forming some sets, the problem is reduced to 22 cases: 11 for the forward pass *(marked from 1 to 11 on the top left hand side corner of each combination featured in* **Figure 9***)* and 11 for the backward pass *(marked from A to L on the bottom right hand side corner of each combination).*

FIGURE 9. *The 117 different combinations for two activities such that* $j_{LR}k$. *Check it at http://www.journalmodernpm.com/public/issue02/Paper01Figure9.pdf*

Considering the finish-start constraint, the earliest start date ES_k for activity k can be calculated as **Figure 10**.

FIGURE 10. *Check figure at http://www.journalmodernpm.com/public/issue02/Figure10.pdf*

For other types of precedence constraints *(SS, FF, SF)*, the formulae can be derived in similar way. For constraints on multiple tasks, the maximum value *(i.e., latest date)* will be used as shown in formula *(1)*.

The corresponding latest finish date ES_j for activity j can be calculated as **Figure 11**.

FIGURE 11. *Check figure at http://www.journalmodernpm.com/public/issue02/Figure11.pdf*

Again, the formulae for SS, FF, SF constraints can be derived in similar way and in case of constraints with multiple tasks, the minimum value *(i.e., earliest date)* will be used as demonstrated in formula *(2)*.

4. Merging PDM, RSM and LSM in a unique system

Real-world construction projects are generally made-up of the three types of activities featured in this paper, i.e., standard *(one-of-a-kind)* project activities, repetitive activities and linear activities. Project management practitioners need the means for modeling and analyzing activity networks made-up of a mixture of all these activities. Based on the definitions and on the formulae given in the three previous sections, expression *(3)*, i.e., RSM calculation formulae for earliest start dates, can be embedded into expressions *(1)*, PDM formulae. *(see* **Figure 12A***)*.

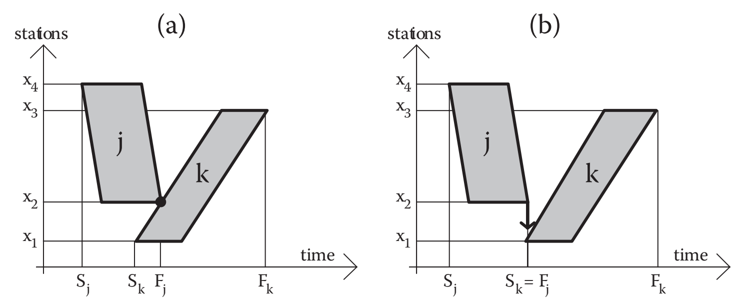


FIGURE 8. *Optimally Calculated vs. PDM Calculated Diagram.*

Similarly, expression *(4)* for latest finish date can be embedded into expressions *(2)*. *(see* **Figure 12B***)*.

We could go ahead and merge the LSM formulae *(5)* and *(6)* with the ones obtained here before. But one can easily see that PDM and RSM are just special cases of the LSM formulae:

- PDM formulae are a special case of LSM with $x_j = y_j$, so that $(y_j - x_j)/\theta_j = 0$, and $TS_j = D_j$.
- RSM formulae are a special case of LSM with $x_j = x_k = 1$, $y_j = y_k = Q_j = Q_k$, $\theta_j > 0$ and , $\theta_k > 0$, so that:

$$\frac{y_j - x_j}{\theta_j} = \frac{Q_j - 1}{\theta_j} \text{ and}$$

$$\frac{x_j - x_k}{\theta_j} = \frac{x_j - x_k}{\theta_k} = \frac{y_j - y_k}{\theta_j} = \frac{y_j - y_k}{\theta_k} = 0 .$$

5. Example

A wide range of projects use linear scheduling approaches: construction of highways and railways, re-pavement of runways, excavation of tunnels, installation of pipelines, and so on. The construction of large particle accelerators also involves such scheduling approaches. At a coordination level, the LHC uses a scheduling approach that is quite similar to the UPSS approach described in this paper *(see Appendix 1)*. This UPSS approach is a spin-off of the scheduling effort needed to coordinate this large-scale scientific project over a decade.

A Microsoft Excel spreadsheet implementation is given here to demonstrate that the UPSS approach can be implemented with widespread means. For the sake of simplicity, the sub-set used as an example is made-up of 16 activities extracted from the LHC project installation coordination schedule: the installation of the general services of sector 7-8 *(one eighth of the main ring)*. Also for the sake of simplicity, repetitive activities were considered in this example. These activities are described in Appendix 2 that is a snapshot of the Microsoft Excel project data entry sheet.

Spreadsheet implementation

The principle used for entering project data is the following: rows can be of two types, activity rows or constraint rows. The activity rows are those with "wu" *(work unit)* in column C; the constraint rows are those with "pc" *(prece-*

FIGURE 12A.

$$
{}_{k} = \begin{cases} ES_{\alpha} \text{ if } \Gamma_j^{-1} = \varnothing \\ \max\limits_{j \in \Gamma_k^{-1}} \left\{ \begin{array}{ll} ES_j + TS_j + L_{jk} + \begin{cases} \frac{Q_j - 1}{\theta_j} - \frac{Q_k - 1}{\theta_k} & \text{if } j,k = \text{repetitive s and } \theta_j < \theta_k \\ 0 & \text{otherwise} \end{cases} & \text{if } \sigma_{jk} = FS \\ ES_j + L_{jk} + \begin{cases} \frac{Q_j - 1}{\theta_j} - \frac{Q_k - 1}{\theta_k} & \text{if } j,k = \text{repetitive s and } \theta_j < \theta_k \\ 0 & \text{otherwise} \end{cases} & \text{if } \sigma_{jk} = SS \\ ES_j - TS_k + L_{jk} + \begin{cases} \frac{Q_j - 1}{\theta_j} - \frac{Q_k - 1}{\theta_k} & \text{if } j,k = \text{repetitive s and } \theta_j < \theta_k \\ 0 & \text{otherwise} \end{cases} & \text{if } \sigma_{jk} = SF \\ ES_j + TS_j - TS_k + L_{jk} + \begin{cases} \frac{Q_j - 1}{\theta_j} - \frac{Q_k - 1}{\theta_k} & \text{if } j,k = \text{repetitive s and } \theta_j < \theta_k \\ 0 & \text{otherwise} \end{cases} & \text{if } \sigma_{jk} = FF \end{array} \right\} \text{ otherwise} \end{cases} , \forall k \qquad (7)
$$

FIGURE 12B.

$$
LF_j = \begin{cases} LF_{\omega} \text{ if } \Gamma_j = \varnothing \\ \min\limits_{k \in \Gamma_j} \left\{ \begin{array}{ll} LF_k - TS_k - L_{jk} - \begin{cases} \frac{Q_k - 1}{\theta_k} - \frac{Q_j - 1}{\theta_j} & \text{if } j,k = \text{repetitive s and } \theta_k > \theta_j \\ 0 & \text{otherwise} \end{cases} & \text{if } \sigma_{jk} = FS \\ LF_k - TS_k + TS_j - L_{jk} - \begin{cases} \frac{Q_k - 1}{\theta_k} - \frac{Q_j - 1}{\theta_j} & \text{if } j,k = \text{repetitive s and } \theta_k > \theta_j \\ 0 & \text{otherwise} \end{cases} & \text{if } \sigma_{jk} = SS \\ LF_k + TS_j - L_{jk} - \begin{cases} \frac{Q_k - 1}{\theta_k} - \frac{Q_j - 1}{\theta_j} & \text{if } j,k = \text{repetitive s and } \theta_k > \theta_j \\ 0 & \text{otherwise} \end{cases} & \text{if } \sigma_{jk} = SF \\ LF_k - L_{jk} - \begin{cases} \frac{Q_k - 1}{\theta_k} - \frac{Q_j - 1}{\theta_j} & \text{if } j,k = \text{repetitive s and } \theta_k > \theta_j \\ 0 & \text{otherwise} \end{cases} & \text{if } \sigma_{jk} = FF \end{array} \right\} \text{ otherwise} \end{cases} , \forall j \qquad (8)
$$

dence constraint) or "tc" *(temporal constraint)* in this column. The constraints listed right below an activity row correspond to this activity. Each task received an identification number *(column B)*. The start and finish stations of tasks of linear or discrete space-constrained activities are entered in columns H and I; their production rates in column J. These rates have been given in distance over time to avoid dealing with infinity for discrete space-constrained activities; the latter are identified with a production rate that is equal to zero. The calculation formulae have been adapted accordingly.

The precedence constraints *("pc" rows)* are entered as follows: the identification number of the predecessor activity is given in column F, the one to which the constraint is applied to is given in column D. The type of precedence constraint is given in column E: it must be one of the following "fs" *(finish-start)*, "ss" *(start-start)*, "sf" *(start-finish)*, "ff" *(finish-finish)* or "sc" *(space-constrained)*. A delay can be associated to precedence constraint; entered in column G.

The temporal constraints *("tc" rows)* are entered as follows: the nature of the constraint is given in column E and the date in column G. The nature of the constraint can be of two types: "nb" *(not before)* or "na" *(not after)*.

The schedule calculations are done by using the schedule spreadsheet calculation formulae given by Seal *(2001)* and Ragsdale *(2003)*, using the classical forward-pass - backward-pass approach. Calculations are done for each of the constraints of an activity; then the earliest date *(forward-pass)* or latest date *(backward-pass)* obtained is kept at activity level. Due to the heaviness of the calculation formulae, they have been coded as four Visual Basic for Excel functions *(see Appendix 4 and Appendix 5)*. The formula has been conceived so that the formulae needed for "wu" rows are easily copied from that of row 5, and also the formulae for the constraint rows. The calculation results are given in columns L to O. Critical activities are immediately highlighted with a red background.

The project Gantt diagram is given on a second sheet *(see Appendix 3)*. It is straightforwardly obtained by plotting parallelograms made of five-point four-segment open polygons with similar start and end points. Each point has two coordinates: a horizontal temporal coordinate and a vertical special coordinate. The vertical position of bar-type activities is forced by entering a figure in column R. To simplify the plotting of the parallelograms, some formulae have been added to columns Q to AF. The color of the parallelogram is the one given in column P.

The Microsoft Excel spreadsheet can be obtained upon request at upss@bonnal.eu.

Comments on the activities

Activities 003 and 005 are not carried out at the installation site *(i.e., the LHC tunnel)*. For that very reason these are bar type activities. Activities 001 and 002 are activities that require the whole tunnel in order to be performed; due to this characteristic they are the discrete space-constrained type. Activities 004, 006, 007, 013 and 014 are spot activities. They correspond to very specific works to be performed in punctual locations in the LHC tunnel; set up along a few meters of the tunnel. All other activities are linear space-constrained activities; all their θ_j have been set to positive values in order to provide a pre-optimization of the schedule.

6. Conclusion

Linear scheduling techniques help to keep the number of activities down in the context of large-scale projects, as described earlier. They also essentially result in a good starting base to efficiently allocate resources and work continuity for the crews in place.

Possible linear scheduling approaches including more "Visual" scheduling methods, such as Line of Balance, have been tried but are not easily maintained throughout the lifetime of a project where production rates differ from the initial schedules. Replacing space constraints with "artificial precedence constraints" results in a working solution, but again fragile in terms of change during the execution of the project.

In our opinion only an analytical extension based on standard PDM yields a scheduling technique that can be consistently used throughout the lifecycle of large-scale projects. While the set of formulae and the number of cases to examine appear daunting, the link to standard PDM is maintained: earliest start dates are calculated by propagation in a forward pass and latest finish dates are obtained during a backward pass. Total and free floats are then calculated straightforwardly, as well as critical activities. Resource-constrained project scheduling optimization algorithms can also be used. One-of-a-kind, repetitive and linear activities are equally considered in the project scheduling model.

FIGURES AND TABLES

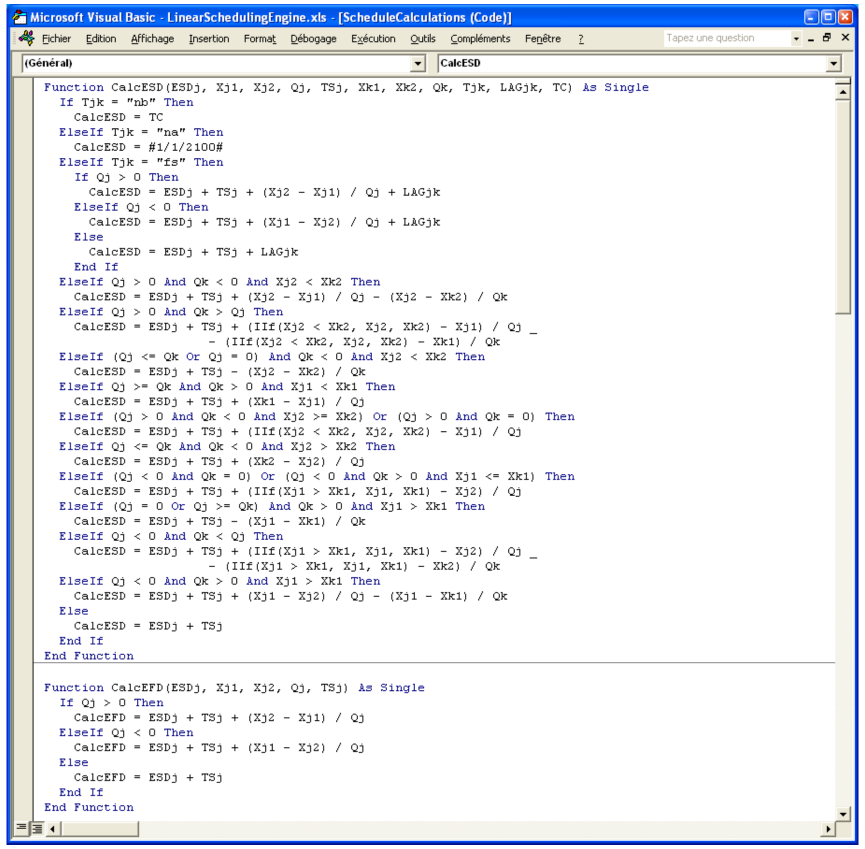
The LHC will provide particle physics community with a tool to reach the energy frontier above 1 TeV. To deliver 14 TeV proton-proton collisions, it will operate with about 1700 cryo-magnets using NbTi super-conductors cooled at 1.8 K. These cryo-magnets will be installed in the 27-km long, 100-m underground 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 manufacturated 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. The LHC co-ordination schedule can be looked at from www.cern.ch.

APPENDIX 1. *The Large Hadron Collider.*

APPENDIX 2. *Microsoft Excel spreadsheet; project data entry sheet.*
Check it at http://www.journalmodernpm.com/public/issue02/Paper01Appendix02.pdf

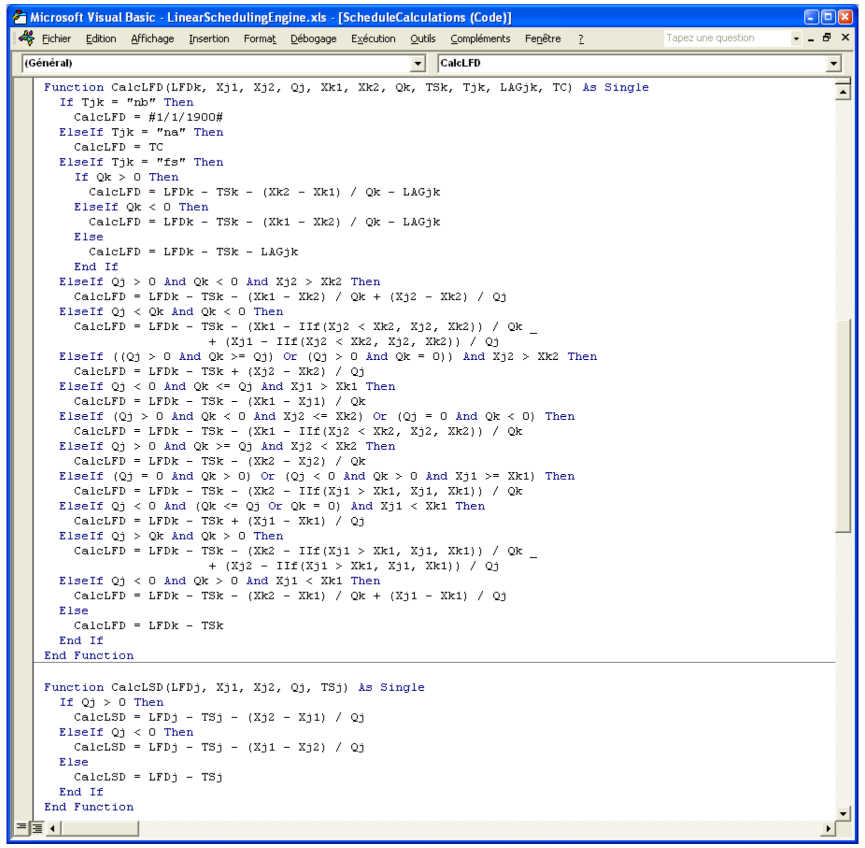
APPENDIX 3. *Microsoft Excel spreadsheet; linear Gantt diagram.*
Check it at http://www.journalmodernpm.com/public/issue02/Paper01Appendix03.pdf




```
Function CalcESD(ESDj, Xj1, Xj2, Qj, TSj, Xk1, Xk2, Qk, Tjk, LAGjk, TC) As Single
  If Tjk = "nb" Then
    CalcESD = TC
  ElseIf Tjk = "na" Then
    CalcESD = #1/1/2100#
  ElseIf Tjk = "fs" Then
    If Qj > 0 Then
      CalcESD = ESDj + TSj + (Xj2 - Xj1) / Qj + LAGjk
    ElseIf Qj < 0 Then
      CalcESD = ESDj + TSj + (Xj1 - Xj2) / Qj + LAGjk
    Else
      CalcESD = ESDj + TSj + LAGjk
    End If
  ElseIf Qj > 0 And Qk < 0 And Xj2 < Xk2 Then
    CalcESD = ESDj + TSj + (Xj2 - Xj1) / Qj - (Xj2 - Xk2) / Qk
  ElseIf Qj > 0 And Qk > Qj Then
    CalcESD = ESDj + TSj + (IIf(Xj2 < Xk2, Xj2, Xk2) - Xj1) / Qj _
                  - (IIf(Xj2 < Xk2, Xj2, Xk2) - Xk1) / Qk
  ElseIf (Qj <= Qk Or Qj = 0) And Qk < 0 And Xj2 < Xk2 Then
    CalcESD = ESDj + TSj - (Xj2 - Xk2) / Qk
  ElseIf Qj >= Qk And Qk > 0 And Xj1 < Xk1 Then
    CalcESD = ESDj + TSj + (Xk1 - Xj1) / Qj
  ElseIf (Qj > 0 And Qk < 0 And Xj2 >= Xk2) Or (Qj > 0 And Qk = 0) Then
    CalcESD = ESDj + TSj + (IIf(Xj2 < Xk2, Xj2, Xk2) - Xj1) / Qj
  ElseIf Qj <= Qk And Qk < 0 And Xj2 > Xk2 Then
    CalcESD = ESDj + TSj + (Xk2 - Xj2) / Qj
  ElseIf (Qj < 0 And Qk = 0) Or (Qj < 0 And Qk > 0 And Xj1 <= Xk1) Then
    CalcESD = ESDj + TSj + (IIf(Xj1 > Xk1, Xj1, Xk1) - Xj2) / Qj
  ElseIf (Qj = 0 Or Qj >= Qk) And Qk > 0 And Xj1 > Xk1 Then
    CalcESD = ESDj + TSj - (Xj1 - Xk1) / Qk
  ElseIf Qj < 0 And Qk < Qj Then
    CalcESD = ESDj + TSj + (IIf(Xj1 > Xk1, Xj1, Xk1) - Xj2) / Qj _
                  - (IIf(Xj1 > Xk1, Xj1, Xk1) - Xk2) / Qk
  ElseIf Qj < 0 And Qk > 0 And Xj1 > Xk1 Then
    CalcESD = ESDj + TSj + (Xj1 - Xj2) / Qj - (Xj1 - Xk1) / Qk
  Else
    CalcESD = ESDj + TSj
  End If
End Function

Function CalcEFD(ESDj, Xj1, Xj2, Qj, TSj) As Single
  If Qj > 0 Then
    CalcEFD = ESDj + TSj + (Xj2 - Xj1) / Qj
  ElseIf Qj < 0 Then
    CalcEFD = ESDj + TSj + (Xj1 - Xj2) / Qj
  Else
    CalcEFD = ESDj + TSj
  End If
End Function
```

APPENDIX 4. *Microsoft Excel spreadsheet; Visual Basic for Excel functions to calculate earliest start dates and earliest finish dates.*

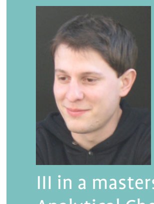



```
Function CalcLFD(LFDk, Xj1, Xj2, Qj, Xk1, Xk2, Qk, TSk, Tjk, LAGjk, TC) As Single
  If Tjk = "nb" Then
    CalcLFD = #1/1/1900#
  ElseIf Tjk = "na" Then
    CalcLFD = TC
  ElseIf Tjk = "fs" Then
    If Qk > 0 Then
      CalcLFD = LFDk - TSk - (Xk2 - Xk1) / Qk - LAGjk
    ElseIf Qk < 0 Then
      CalcLFD = LFDk - TSk - (Xk1 - Xk2) / Qk - LAGjk
    Else
      CalcLFD = LFDk - TSk - LAGjk
    End If
  ElseIf Qj > 0 And Qk < 0 And Xj2 > Xk2 Then
    CalcLFD = LFDk - TSk - (Xk1 - Xk2) / Qk + (Xj2 - Xk2) / Qj
  ElseIf Qj < Qk And Qk < 0 Then
    CalcLFD = LFDk - TSk - (Xk1 - IIf(Xj2 < Xk2, Xj2, Xk2)) / Qk _
                  + (Xj1 - IIf(Xj2 < Xk2, Xj2, Xk2)) / Qj
  ElseIf ((Qj > 0 And Qk >= Qj) Or (Qj > 0 And Qk = 0)) And Xj2 > Xk2 Then
    CalcLFD = LFDk - TSk + (Xj2 - Xk2) / Qj
  ElseIf Qj < 0 And Qk <= Qj And Xj1 > Xk1 Then
    CalcLFD = LFDk - TSk - (Xk1 - Xj1) / Qk
  ElseIf (Qj > 0 And Qk < 0 And Xj2 <= Xk2) Or (Qj = 0 And Qk < 0) Then
    CalcLFD = LFDk - TSk - (Xk1 - IIf(Xj2 < Xk2, Xj2, Xk2)) / Qk
  ElseIf Qj > 0 And Qk >= Qj And Xj2 < Xk2 Then
    CalcLFD = LFDk - TSk - (Xk2 - Xj2) / Qk
  ElseIf (Qj = 0 And Qk > 0) Or (Qj < 0 And Qk > 0 And Xj1 >= Xk1) Then
    CalcLFD = LFDk - TSk - (Xk2 - IIf(Xj1 > Xk1, Xj1, Xk1)) / Qk
  ElseIf Qj < 0 And (Qk <= Qj Or Qk = 0) And Xj1 < Xk1 Then
    CalcLFD = LFDk - TSk + (Xj1 - Xk1) / Qj
  ElseIf Qj > Qk And Qk > 0 Then
    CalcLFD = LFDk - TSk - (Xk2 - IIf(Xj1 > Xk1, Xj1, Xk1)) / Qk _
                  + (Xj2 - IIf(Xj1 > Xk1, Xj1, Xk1)) / Qj
  ElseIf Qj < 0 And Qk > 0 And Xj1 < Xk1 Then
    CalcLFD = LFDk - TSk - (Xk2 - Xk1) / Qk + (Xj1 - Xk1) / Qj
  Else
    CalcLFD = LFDk - TSk
  End If
End Function

Function CalcLSD(LFDj, Xj1, Xj2, Qj, TSj) As Single
  If Qj > 0 Then
    CalcLSD = LFDj - TSj - (Xj2 - Xj1) / Qj
  ElseIf Qj < 0 Then
    CalcLSD = LFDj - TSj - (Xj1 - Xj2) / Qj
  Else
    CalcLSD = LFDj - TSj
  End If
End Function
```

APPENDIX 5. *Microsoft Excel spreadsheet; Visual Basic for Excel functions to calculate latest finish dates and latest start dates.*

authors



Mathieu Baudin I was born in 1988, and graduated in 2011 from the French Engineering school Centrale Marseille. This degree is equivalent to a masters degree in engineering. In 2011 as well, I graduated from the University Paul Cézanne, Aix-Marseille III in a masters degree in Theoretical Chemistry, and Analytical Chemistry for which I performed a six month master thesis at the University of Nottingham in translational medical imaging, with Prof. Thomas Meersmann.
I am now performing my PhD at CERN, Geneva, Switzerland, in industrial engineering. My grad school is at Ecole Nationale Supérieure d'Arts et Métiers ParisTech (ENSAM ParisTech, Paris, France). I am working on intervention models suited to collaborative intervention planning and scheduling, and their optimization based on radiation protection data.

Pierre Bonnal Dr. Pierre Bonnal holds engineering degrees in mechanical engineering from the l'École nationale d'ingénieurs de Tarbes, France, and in industrial engineering from the École centrale Marseille, France, a M.Sc. in project management from the University of Quebec at Trois-Rivières in Canada and a Ph.D. in industrial management from the Institut national polytechnique de Toulouse, France.Pierre claims a solid industrial background including work with Airbus Industrie (F); Fatigue Technology Inc. (USA); Doris Engineering (F), Areva-SGN (F) as well as more than sixteen years with CERN (CH) mainly involved with the LHC project management. In 2008, he benefited of a two-year sabbatical to join the University of Applied Sciences of Western Switzerland-Geneva as a professor and dean of the Department of Business Administration. As from September 2010, he is back to CERN, involved in project management support to the many studies and projects on-going at CERN. Pierre is also the CERN's scientist-in-charge of a European funded initiative that aims at developing enhanced means (processes, hardware systems and software) for preventing human intervention for increased safety in infrastructures emitting ionizing radiation. Pierre is a former chairperson of the IPMA affiliated Swiss Project Management Association of Western Switzerland and a usual senior lecturer in project management and R&D management with the business schools of the Universities of Geneva and Lausanne in Switzerland.

Jurgen De Jonghe CERN, Geneva, Switzerland

BIBLIOGRAPHIC references

Arditi, D. ; Albulak, M. Z. *(1986)* "Line-of-balance scheduling in pavement construction." *J. Constr. Eng. Manage.*, ASCE, **112** *(3)* 411–424.

Arditi, D.; Psarros, M. K. *(1987)* "SYRUS — System for repetitive unit scheduling." *Proc. of the NORDNET/INTERNET/PMI'87 Conf. on Project Management*, Verkefnastjomun, the Icelandic Project Management Society, Reykjavik, Iceland.

Arditi, D.; Tokdemir, O. B.; Suh, K. *(1999)* "Effect of learning on line-of-balance scheduling." *Int. J. Proj. Manage.*, **15** *(5)* 265–277.

Ashley, D. B. *(1980)* "Simulation of repetitive-unit construction." *J. Constr. Div.*, ASCE, **106** *(2)* 185–194.

Barrie, D. S. and Paulson, B. C. Jr. *(1978) Professional Construction Management.* N.-Y.: McGraw-Hill, pp. 232-233.

Birrell, G. S. *(1980)* "Construction Planning - Beyond the Critical Path." *J. Constr. Div.*, ASCE, **106** *(CO3)* 389-407.

Borcherding, J. D. *(1989) Factors that have an adverse effect on productivity.* Discussion Paper of Construction Operation, University of Texas at Austin, Austin, TX.

Carr, R. I.; Meyer, W. L. *(1974)* "Planning Construction of Repetitive Building Units." *J. Constr. Div.*, ASCE, **100** *(CO3)* 403-412.

Chrzanowski, E. N.; Johnston, D. W. *(1986)* "Application of Linear Scheduling." *J. Constr. Eng. Manage.*, ASCE, **112** *(4)* 476-491.

Construction Industry Cost Effectivenes Task Force *(1980)* "Scheduled Overtime Effect on Construction Projects." Report. The Business Roundtable, New York.

Dressler, J. *(1980)* "Construction Management in West Germany." *J. Constr. Div.*, ASCE, **106** *(CO4)* 447-487.

Eldin, N. N.; Senouci, A. B. *(1994)* "Scheduling and control of linear projects." *Canadian Journal of Civil Engineering*, **21** *(2)* 219–230.

Eldin, N. N.; Senouci, A. B. *(2000)* "Scheduling of linear projects with single loop structures." *Advances in Engineering Software*, **31**. 803–814.

El-Rayes, K.; Moselhi, O. *(1998)* "Resource-driven scheduling of repetitive activities." *Constr. Manage. Econom.*, **16** *(4)* 433–446.

El-Rayes, K.; Moselhi, O. *(2001a)* "Optimizing resource utilization for repetitive construction projects." *J. Constr. Eng. Manage.*, ASCE, **127** *(1)* 18–27.

El-Rayes, K. *(2001b)* "Object oriented model for repetitive construction scheduling." *J. Constr. Eng. Manage.*, ASCE, **127** *(3)* 99–205.

El-Rayes, K.; Ramanathan, R.; Moselhi, O. *(2002)* "An object-oriented model for planning and control of housing construction." *Constr. Manage. Econom.*, **20** *(3)* 201-210.

Gorman, J. E. *(1972)* "How to Get Visual Impact on Planning Diagrams," *Roads and Streets*, **115** *(8)* 74-75.

Halpin, D. W.; Woodhead, R. W. *(1976) Design of Construction Process Operations*, N.-Y.: John Wiley & Sons, 30-36.

Handa, V.; Barcia R. *(1986)* "Linear scheduling using optimal control theory." Journal of Construction Engineering and Management, ASCE, 112 *(3)*, May/June 1986, pp. 387–393.

Harmelink, D. J. *(2001)* "Linear Scheduling Model: float characteristics." *J. Constr. Eng. Manage.*, ASCE, **127** *(4)* 255–260.

Harmelink, D. J.; Rowings, J. E. *(1998)* "Linear scheduling model: Development of controlling activity path." *J. Constr. Eng. Manage.*, ASCE, **124** *(4)* 263–268.

Harris, F. C.; Evans, J. B. *(1977)* "Road Construction - Simulation Game for Site Managers." *J. Constr. Div.*, ASCE, **103** *(CO3)* 405-414.

Harris, R. B.; Ioannou, P. G. *(1998)* "Scheduling Projects with Repeating Activities." *J. Constr. Eng. Manage.*, ASCE, **124** *(4)* 269-278.

Hartmann, S. *(2001)* Project scheduling under limited resources: models, methods and applications. Belin: Springer.

Hegazy, T.; Moselhi, O.; Fazio, P. *(1993)* "BAL: An algorithm for scheduling and control of linear projects." *AACE Transactions*, Morgantown, WV, 8.1–8.14.

Johnston, D. W. *(1981)* "Linear Scheduling Method for Highway Construction." *J. Constr. Div.*, ASCE, **107** *(CO2)* 241-261.

Kavanagh, D. P. *(1985)* "SIREN: a repetitive construction simulation model." *J. Constr. Eng. Manage.*, ASCE, **111** *(3)* 308–323.

Leu, S.; Hwang, S. *(2001)* "Optimal repetitive scheduling model with shareable resource constraint." *J. Constr. Eng. Manage.*, ASCE, **127** *(4)* 270–280.

Lumsden, P. *(1968) The line-of-balance method.* Tarrytown, N.-Y.: Pergamon.

Lutz, J. D.; Halpin, D. W. *(1992)* "Analyzing linear construction operations using simulation and line-of-balance." Proc. of the 71st Annual Meeting of the Transportation Research Board, Record 1351, National Academy Press, Washington, D.C., 48–56.

Lutz, J. D.; Halpin, D. W.; and Wilson, J. R. *(1994).* "Simulation of learning development in repetitive construction." *J. Constr. Eng. Manage.*, ASCE, **120** *(4)* 753–773.

Moselhi, O.; El-Rayes, K. *(1993)* "Scheduling of repetitive projects with cost optimization." *J. Constr. Eng. Manage.*, ASCE, **119** *(4)* 681–697.

Moselhi, O.; Hassanein, A. *(2003)* "Optimized scheduling of linear projects." *J. Constr. Eng. Manage.*, ASCE, **129** *(6)* 664–673.

O'Brien, J. J. (ed.) *(1969) Scheduling Handbook.* N.-Y.: McGraw-Hill.

O'Brien, J. J. *(1975)* "VPM Scheduling for High-Rise Buildings." *J. Constr. Div.*, ASCE, **101** *(CO4)* 895-905.

O'Brien, J. J.; Kreitzberg, F. C.; Mikes, F. W. *(1985)* "Network scheduling variation for repetitive work." *J. Constr. Eng. Manage.*, ASCE, **111** *(2)* 105–116.

Peer, S. *(1974)* "Network Analysis and Construction Planning." *J. Constr. Div.*, ASCE, **100** *(CO3)* 203-210.

Psarros, M. E. *(1987) SYRUS: A program for repetitive projects.* Master's thesis, Dept. of Civil Engineering, Illinois Institute of Technology, Chicago, IL.

Ragsdale, C. T. *(2003)* "A new approach to implementing project networks in spreadsheets." *INFORMS Transaction on Education.* 3 *(3)* 76–85.

Reda, R. *(1990).* "RPM: Repetitive Project Modelling." *J. Constr. Eng. Manage.*, ASCE, 116 *(2)* 316–330.

Russell, A. D.; Caselton, W. F. *(1988)* "Extensions to Linear Scheduling Optimization." *J. Constr. Eng. Manage.*, ASCE, 114 *(1)*, 36-52.

Russell, A. D.; Wong, W. C. M. *(1993)* "New generation of planning structures." *J. Constr. Eng. Manage.*, ASCE, 119 *(2)* 196–214.

Seal, K. C. *(2001)* "A generalized PERT/CPM implementation in a spreadsheet." *INFORMS Transaction on Education.* 2 *(1)* 16–26.

Selinger, S. *(1980)* "Construction Planning for Linear Projects." *J. Constr. Div.*, ASCE, 106 *(CO2)* 195-205.

Senouci, A. B.; Eldin, N. N. *(1996)* "Dynamic programming approach to scheduling of non-serial linear project." *J. Comput. Civ. Eng.*, 10 *(2)* 106–114.

Srigungvarl, P. *(1992)* "A knowledge-based integrated system for line-of-balance applications in high-rise building construction." Doctoral thesis. Div. of Structural Engineering and Construction, Asian Institute of Technology, Bangkok, Thailand.

Stradal, O.; Cacha, J. *(1982)* "Time Space Scheduling Method." *J. Constr. Div.*, ASCE, **108** *(CO3)* 445-457.

Suhail, S. A.; Neale, R. H. *(1994)* "CPM/LOB: New methodology to integrate CPM and Line of Balance." *J. Constr. Eng. Manage.*, ASCE, **120** *(2)* 667–684.

Thabet, W. Y.; Beliveau, Y. J. *(1994)* "HVLS: Horizontal and Vertical Logic Scheduling for Multistory Projects." *J. Constr. Eng. Manage.*, ASCE, **120** *(4)* 875-892.

Wang, C.; Huang, Y. *(1998)* "Controlling activity interval times in LOB scheduling." *Constr. Manage. Econom.*, **16** *(1)* 5–16.

Whiteman, W. E.; Irwig, H. G. *(1988)* "Disturbance Scheduling Technique for Managing Renovation Work." *J. Constr. Eng. Manage.*, ASCE, **114** *(2)* 191-213.