

Chapter 8

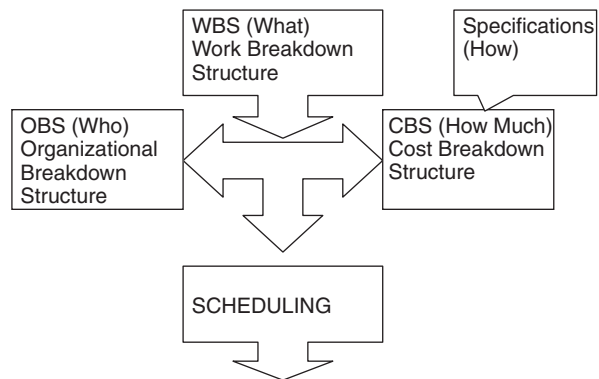
Planning and Scheduling

8.1 Project Planning: Breakdown Structuring

Construction project planning is a method of determining “What” is going to be done, “How” things are going to be done, “Who” will be doing activities and “How much” activities will cost.

In this sense planning does not cover scheduling, which addresses the “When”, but once planning is complete scheduling can be done (Fig. 8.1).

Fig. 8.1 General framework for the planning process



8.1.1 Work Breakdown Structure – “What”

When projects are simple, consisting of few defined activities, it might be possible for a single person to grasp the total construction effort with little difficulty. Unfortunately, most projects for which formal plans are prepared tend to be defined with dozens or even hundreds or thousands of activities: the larger the project, the greater the number of activities and higher the level of detail managers have to handle.

When a project plan consists of numerous activities, it is often advisable to organize the activities in some way to allow communication of plan information to others and to maintain an understanding of the various aspects of the project. While there are many ways that a plan can be organized, one common practice is the Work Breakdown Structure (WBS).

The WBS is a convenient method for decomposing the project complexity in a rational manner into work packages and elementary activities. Some firms prefer to use a standard means of identifying work packages common to all similar projects. These work packages are then coded so that both costs and the schedule can be controlled. A common numerical accounting system is then applied to the activities, so that the coding indicates factors such as the type of material involved or the physical location within the project.

In essence, the WBS divides and subdivides a project into different components, whether by area, phase, function, or other considerations. The highest level in the WBS consists of a single element, the project. At the next level, there may be only a few elements or items. Naturally, the further one goes down within the WBS, the greater the granularity of decomposition and the amount of detail. Regardless of the means used to define the elements, individual tasks are to be defined for the lowest level in the hierarchy or at the greatest level of detail that is required to adequately manage and control the construction process. The level of detail used will be determined by the scheduling needs and the roles of the people viewing the WBS. For example, if one is a homeowner and having a house built, one is interested in the completion date of the project, but a subcontractor will be primarily interested in information related specifically to the task this has direct responsibility.

Commonly there are three main types of WBS, namely, the Project WBS, Standard WBS and Contract WBS.

The Project WBS is an operational tool usually prepared by contractors to monitor and control the work (Fig. 8.2 is an example for a new stadium construction project).

A standard WBS is a breakdown structure of activities carried out in the past for a similar project: the past project WBS can that can be used as a template for the new one. Figure 8.3 shows the highest levels of a sample template WBS that might be used for turn-key construction of an ordinary industrial building.

A contract WBS is agreed between owner and contractor. This is a decomposition of the scope of work into the main elements that will be used for progress measurement, control and payment of the contract price. It may include less detail than a Project WBS.

To summarize, WBS is a deliverable-oriented decomposition of the project scope (Project Management Institute 2008) until a sufficient level of granularity enables easy definition of all information required to execute and manage detailed tasks.

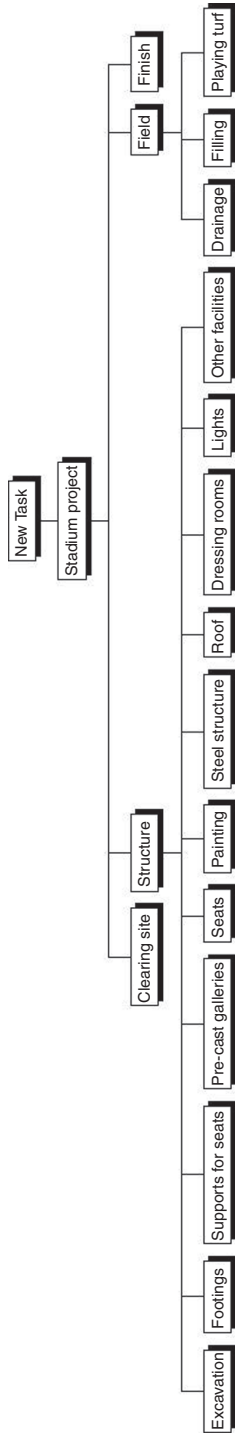


Fig. 8.2 Example of Project WBS of a new stadium construction project. WBS chart based on the case story by E. Turban and J.R. Meredith “The Sharon Construction Corporation” from Meredith and Mantel (2006)

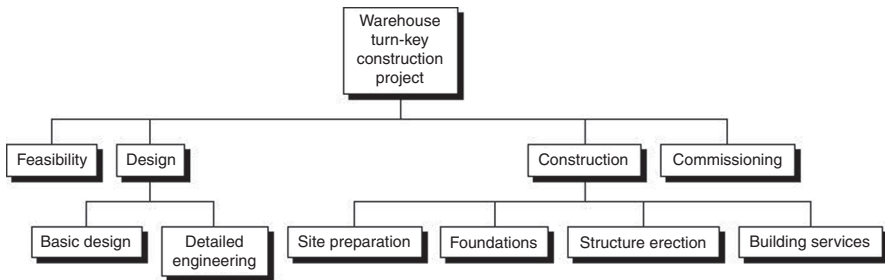


Fig. 8.3 Example of standard WBS for a warehouse construction project

8.1.2 Organizational Breakdown Structure – “Who”

Once what needs to be done is defined, it is necessary that all human resources required to perform the project are identified. Depending on the portions of work scope, the project may need engineering skills, procurement capabilities, construction labor, management staff, etc.

The Organization Breakdown Structure is a practical method to decompose the pool of human resources needed to execute all of the tasks into different competence areas and then into project roles, independently of the number of individuals that will be assigned the specified role (Fig. 8.4).

The OBS is prepared with the idea that each task in the WBS must be assigned to a role or committee of roles. In other words, roles are allocated to detailed tasks

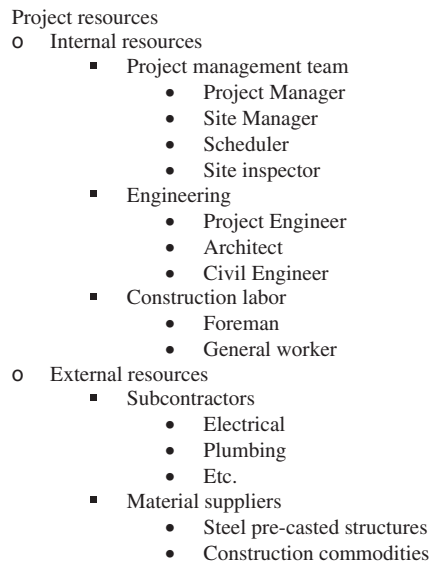


Fig. 8.4 Example of OBS

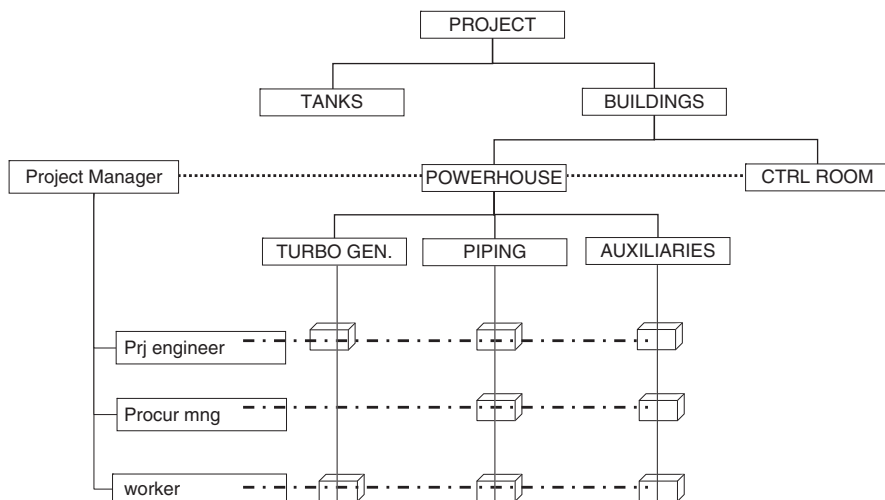


Fig. 8.5 Example of WBS/OBS allocation matrix for a power station construction project

with a specified number of resources and related estimated work load required to perform the task.

Figure 8.5 is a simplified depiction of how a WBS and OBS correlate. It is imperative during the monitoring and controlling phase of the construction project that there is a responsible person for each activity.

To summarize the planning process thus far, we have created the WBS and have now incorporated a responsible committee or person to each element of the WBS. In essence, from activities developed in the WBS we have allocated resources through the OBS.

8.1.3 Cost Breakdown Structure – “How Much”

Now that we have discussed “what” is going to be accomplished through the WBS and “who” is going to perform activities through the OBS, owners and contractors want to know how much things will cost.

Determining the cost is done through the Cost Breakdown Structure (CBS). The CBS is a system for dividing a project into hardware elements and sub elements, functions and sub functions and cost categories. It is a hierarchical structure that classifies resources into cost accounts, typically labor, materials, and other direct costs. In addition it represents the economic breakdown of the project into budgets per work package. This will allow the project manager to track project progress and expenditure according to planning breakdown of activities and responsibilities.

A CBS includes all direct full cost of labor, material, as well as the so-called project overhead, which is still a direct cost required to execute the project. Project

overhead embraces the cost of construction equipment (usually under the terms of average amortization of construction assets), project management, design services, permits and insurance fees. CBS does not have to include the company's overhead not associated with the project, such as general office salaries, utilities, insurance, taxes, interest, and other expenses out of the direct control of the project team, but rather inherent with corporate top management's action.

There are two main approaches to direct cost breakdown structuring. Which is used in a particular circumstance depends on the different purposes of cost accounting.

The first one makes use of the WBS as the project cost control structure, so that the CBS and WBS are the same structure and each cost account is consistent with a work package or detailed task. In other words, the accounting structure is the same WBS that has been filled with cost information: the end result is a hierarchical structure of cost to be used by the project team for both budgeting, accounting and control. With this kind of CBS, Activity Based Costing (ABC) method drives both estimation of budget and accounting of actual expenditures. The advantage is that project budgeting and tracking develop on the WBS exactly in the way the facility is going to be built, with detailed analysis at the final level of decomposition of the WBS: the cost of an elementary activity may include a combined summation of full cost of labor, quantity of material, equipment, and lump-sum cost of subcontract or service.

To define the budget, a different methodology may apply to parts of the breakdown depending on the specific nature of items or elements. Subcontractor quotes are of practical use when a specialized subcontractor is assigned a job. Quantity takeoffs are obtained by multiplying the measured quantities by the unit cost, which includes material, equipment and labor as a whole. Challenges here are the tremendous detail complexity of line items, the dependence of the estimated quantities on construction methods, and the determination of unit cost based on historical data. Material takeoff estimation is needed when data about unit costs for complete installation of materials are unknown. For each line item in the cost breakdown, a quantity of material required, Q , must be determined. For each item the unit cost of material, M , can be estimated using quotes from local material suppliers. For most line items equipment is involved in the construction process, and an equipment rate of cost, E_M (cost per unit of material), must be determined. In addition, labor costs – which are often greater than material cost – must be incorporated by multiplying the hourly wage rate, W , and the labor cost per unit of material (productivity) L . Combining these factors in the following equation produces an estimate of the direct cost for a given item:

$$\text{Total cost \$} = Q * (M + E_M + W * L) \quad (8.1)$$

Regardless of the method applied, careful consideration of wages and productivity has to be taken into account for appropriate detailed budgeting. Labor cost estimation W) is affected by several components, namely wages, insurance, social security, benefits and premiums. Productivity (L) impacts a project in many ways.

At the beginning of a job workers will typically have lower productivity on account of inexperience with the particular routine to be followed. As time progresses they become more efficient in their work with repetition due to the effects of learning: an effect expressed in learning curves (Kerzner 2001).

However, some projects have little repetitive tasks, and therefore must account for this factor in the project estimate. When productivity is less than initially expected a project may begin to fall behind schedule. As a result the project manager may increase pressure in order to finish more quickly. However, as hours per day of work increase, worker productivity per hour is known to decrease. Productivity also suffers greatly over the medium- and long-term as workers become fatigued and lose motivation. This reciprocal process can be damaging to the success of a project if it is not realized. Productivity can be measured, but the results of corrective actions are highly uncertain. In this realm, a project manager with good experience and a good understanding of his personnel can identify problems and attempt to remedy them – ideally before the time such problems begin to be evident in project reports and failure to meet the schedule of values. Lost time due to low productivity can be incorporated into an updated cost estimation, but prior to construction this additional cost is most easily calculated as a contingency. Applying probabilistic models to estimation calculations allows planners to gain a deeper insight into the effects of uncertainty in costs.

Even if the probabilistic distribution is not fully known, the effects of changing the range of outcomes can help planners see where major problems may occur. Finding the variance of just one portion of a project can give insight into the effects of increased costs will have on the total project cost (see Section 8.5 “Uncertainty”).

This practical first way of accounting for cost based on project activities is usually adopted when a firm does not have a specific cost control accounting system.

A second approach to CBS budgeting is to use the corporate multiple-project cost control structure as the project cost accounting system. With this method, each WBS activity has to be associated with a cost account by the means of a cost code.

The coding system may be a firm-specific or a common standardized one, such as the Master Format developed by the Construction Specifications Institute of the United States, the ISO UniClass, the German KKS valuable for power plant construction, or the Construction and the Engineering Information Classification System.

An illustration of how a cost code is often represented is below in Fig. 8.6. The cost code reflects the WBS decomposition and contains several subfields: the first is the project code for the first level of the WBS, the second code physically identifies areas or sub-facilities, then the Masterformat code describes the activity, and the final digit represents the distribution code (0 = Total, 1 = Labor, 2 = Material, 3 = Equipment, 4 = Subcontract).

CBS is also utilized in different approaches by means of delivery.

In case of a Design Bid Build delivery system, the Contract WBS is the same as the Contract CBS because schedules of values are paid unit price by performed units. As a result, most often the contractor’s own CBS used for cost accounting is quite different from the contract CBS; the project operating WBS will also differ

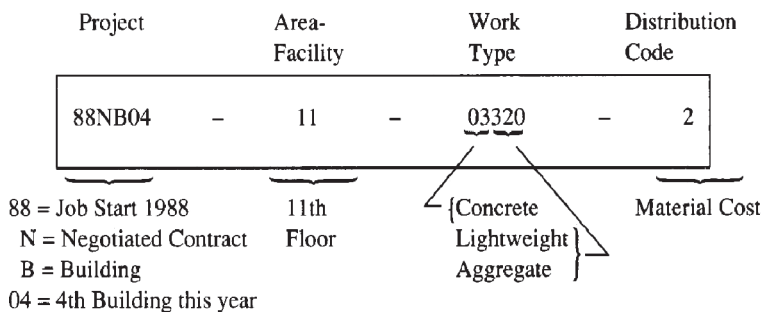


Fig. 8.6 Example of cost code integrating the UCI/CSI MASTERFORMAT

from the Contract WBS. In such circumstances, the solution is to keep the revenue and cost separate.

Instead, in a Design-Build or Turnkey project, the Contract WBS is prepared by the contractor himself and therefore it is equal to the Project WBS. The sum of the contract work packages is paid cost plus and the contract price is paid on a project progress basis. Since the revenue is a function of cost, then the project WBS should reflect the CBS, if corporate cost control is required. If this is required, then it is recommended to use the higher level of CBS codes, and then break down according to the job needs.

In summary, planning tasks include scope of work definition and budgeting, as a fundamental precursor to scheduling the estimated time to perform a project, as discussed in the following paragraphs.

8.2 Deterministic Scheduling Principles

Deterministic scheduling is just one of the many tools available to project managers during the planning stages of a project. However, it may be one of the most important because it both lowers chance of delay and assists in recovering from delay, resolving responsibility. Indeed, delays often result simply from poor planning.

Accurate scheduling assists in reasoning about a huge number of details (e.g. thousands of activities), and determines a lot of things, including expenditure estimates for crews and materials, expected opening dates (there may be situations where a strict opening date is highly important, such as a new production facility), scheduling changes with sufficient flexibility to not affect the completion date, and others.

Scheduling also allows for accountability. Setting milestones from the beginning allows for the project managers or the owners to pinpoint exactly what went wrong and who or what was responsible for a delay.

A schedule is also a good communication tool, between the managers, the owners, investors, and the general public. Schedules give an overall sense of the project's expected progress. Without schedules, it's much more difficult to explain to someone unfamiliar with the project what is expected to take place.

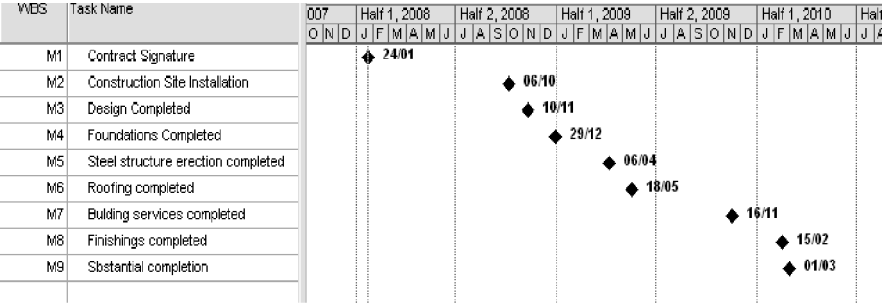


Fig. 8.7 Example of master schedule under the form of a milestone chart

A schedule can also be used a contractual tool. Some payment schemes are based on scheduling. Some offer incentives for finishing the job on time or ahead of schedule. With an accurate schedule, these sorts of incentives can be offered fairly in the contract from the very beginning. Also, in the case of a lawsuit, a good schedule can serve as great evidence in support of the parties.

To put a schedule into effect it is recommended to avoid any imbalanced use (such as to use it early on and discarding later), to game for liability reasons (i.e.: schedule as a biased document to support the originator’s rights), or to use for central PM office only. In contrast, schedules should be used as shared management tools to get to an integrated point of view for both the owner and the contractor.

Schedule documents can be subsumed mainly in two types. One is the *Master Schedule* that is used as the contract baseline, usually under the form of a milestone chart, as in Fig. 8.7.

The other is the *Project Schedule* which is used to monitor and control the actual progress of the project. This schedule is usually based on the WBS and is very meticulous. It usually includes detailed plans, such as engineering schedules, construction sequencing, quality-assurance activities, as well as procurement plans.

For example, a procurement detailed schedule involves trying to schedule when materials will be ready and available on site for installation. This is often difficult to estimate, especially for custom built items, though it is very important to keep work on pace. Without the proper materials on site, workers may be sitting around and money will be spent on entertaining them.

For the project schedule, typically there are revisions performed on a weekly, monthly, or other periodic system. Then, these revisions are used to track progress against the original schedule. This allows for the managers to make any changes, if necessary, to the work (see later Chap. 9).

8.3 Scheduling Systems

So how do we schedule? There are several forms of schedules and several methods used to determine accurately the schedule. The following methods will be discussed in greater detail in the following: task matrix, Gantt chart, network diagram, and line-of-balance scheduling.

8.3.1 Matrix Scheduling

Matrix scheduling is fairly simple. It is usually used for small, less complex projects because of this simplicity. It also doesn't have a clear way of showing interactions between different tasks. Table 8.1 shows an example of matrix scheduling.

Table 8.1 Example of matrix scheduling

Task	Original schedule	Review week 8	Review week 16	Actual week 22
Mechanical design – Start	1	1	1	1
Mechanical design – Finish	10	10	10	10
Electrical design – Start	8	8	8	8
Electrical design – Finish	16	16	18 (delay 2)	18
Software dev – Start	14	14	16 (delay 2)	18 (delay 4)
etc.				

This example shows an original schedule and then makes comparisons based on the reviews every 8 weeks. In the week 16 review, the delays began in the electrical design. However, it is unclear whether the design review caused the delay in the SW development, or whether that was due to something else. The next method we discuss shows more clearly those relationships.

8.3.2 Gantt Chart Scheduling

Figure 8.8 shows a basic Gantt chart. Here we begin to see a more clear relationship between tasks, though not completely. For example, we know that design has to take place before construction, but construction could begin before the design is completed. So there is some intuition as to which tasks are related, but not an explicit statement of dependencies. Each bar represents the amount of time that its respective task will take.

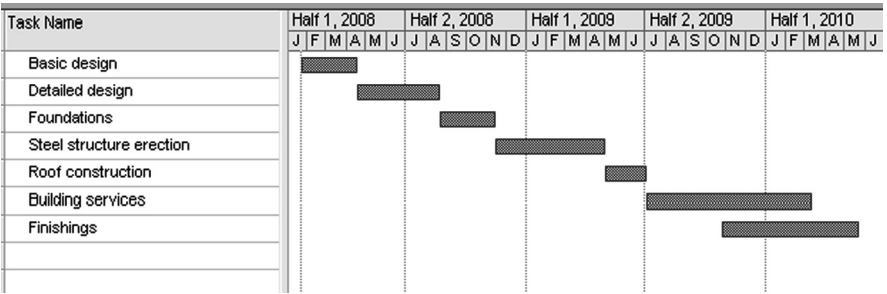


Fig. 8.8 Example of a Gantt chart

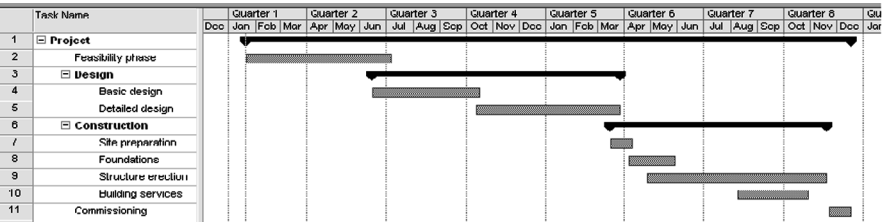


Fig. 8.9 Gantt chart schedule of Fig. 8.2 WBS

This form of scheduling is far superior to that of the matrix scheduling in that it’s more effective as a communication tool. This type of chart is very easy for anyone to understand and allows for the owner or manager to more effectively communicate how the project will proceed.

There may also be WBS levels of scheduling. Figure 8.9 illustrates that idea. However, we need a more detailed way of showing relationships of activities.

8.3.3 Network Diagramming

This method is a most robust way of showing and calculating a schedule. Using this method of scheduling, it is fairly easy to use software tools to calculate project duration and optimize allocation of labor and resources. It is also relatively easy to find the areas in the schedule which are more flexible to change.

Basically, the process of constructing a network system is composed of the following stages:

- First, the tasks are drawn from WBS work packages and assigned expected deterministic duration, estimate cost, and resources as discussed in Section 8.1. The method for obtaining the deterministic durations may vary depending on the task, but mostly it’s a factor of amount of work to be performed, productivity, number of resources and equipment used. Costs can also be assigned to each task based on the original cost estimates or through assignment of human resources, materials and equipment to each task. In any case, the common assumption in deterministic estimation is that all activity attributes can be determined as certain values with very little margin of error (in a later section we will discuss about probabilistic estimation of task attributes).
- Second, each task is assigned precedence relationships with other tasks. In other words, if task B cannot be started until task A is finished, that relationship is defined in this method.
- Then, the network diagram is solved and optimized using various ways, such as Critical Path Method, Precedence Diagramming Method and Program Evaluation Review Technique. This often implies iteration: if the solution of the network acceptable in terms of total project duration and resource allocation, then terminate. If it is not acceptable, it is needed to impose dependencies or added/reduced resources.

Table 8.2 Task list from project breakdown

Task name	ID
Feasibility study	1
Basic design	2
Detailed design	3
Site preparation	4
Foundations	5
Structure erection	6
Building services	7
Commissioning	8

Let us first discuss the precedence relationship process. The first step is to list the activities that need to be performed. This is done by taking the tasks defined in the WBS and listing them. The following shows an illustration of listing the activities from WBS detailed in Table 8.2.

Once the tasks are listed, one has to assign precedence relationships. Sometimes activities can overlap; sometimes they have to occur in series. So we define a matrix of precedence to capture this idea (Table 8.3).

As per the above diagram, the relationships between activities reflect the constraints in sequencing the tasks, such as regulatory or contractual, physical or functional, financial, managerial, and environmental constraints. Also, resource availability may restrain multiple tasks in parallel: for example, if only one crew is available to perform the job all construction tasks have to be performed in series.

Finally, representation is required to capture the above relationship matrix in diagram form, again to allow the scheduler to clearly understand how activities will unfold. There are two ways of graphical diagramming: Activities on Arrows (AOA) or Activities on Nodes (AON).

AOA representation keeps similarities to a Gantt format. In this method, Nodes represent start and finish events for each activity. Arrows represent the tasks that need to be done to get to the next activity. The diagram in Fig. 8.10 shows a sample task depicted with AOA mode.

A problem that sometimes arises using this method is that we need to create “dummy nodes”. These nodes arise when one task has two or more precedent activities, as in the example. Because a node may only have one incoming arrow, dummy nodes need to be created. Figure 8.11 illustrates that idea.

Once all the nodes are accurately represented, one may construct the final diagram. Figure 8.12 shows an AOA representation of the previous schedule.

AON is the method most popularly used in today’s project planning software programs, such as Microsoft Project or Primavera. A task is represented in Fig. 8.13.

Fig. 8.10 Activities on arrows representation (AOA)



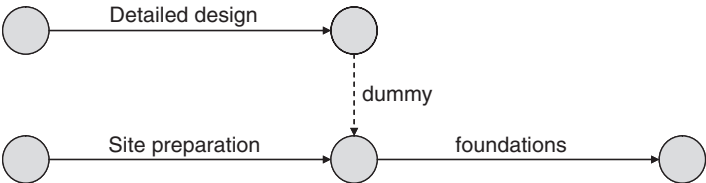


Fig. 8.11 Dummy activities are necessary for AOA representation

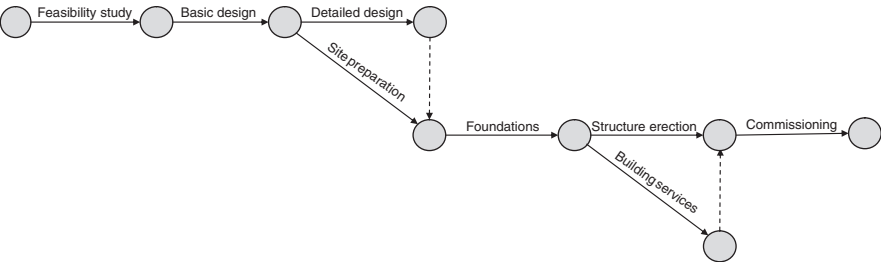


Fig. 8.12 Example of a network diagram with AOA representation

Thus, from the example in Fig. 8.13, the resulting AON diagram is graphed in Fig. 8.14.

Also, as with the Gantt chart, we are able to illustrate a hierarchy of networks, as in Fig. 8.15. So setting different levels of hierarchy may help in presentation, where a client or a top manager may not need to know the details of the construction, but may just want an overall view of the process.

AON representation is also closely related to the so-called “Precedence Diagram Method” (PDM) or “Bubble Diagram Method” that allow for representing richer



Fig. 8.13 Activities on nodes representation

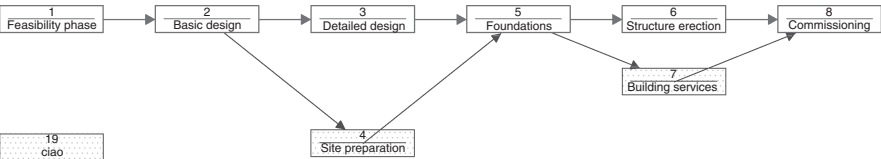


Fig. 8.14 Precedence diagram for concrete footing construction

Table 8.3 Matrix of precedence relationships between tasks

Task name	ID	Feasibility study	Basic design	Detailed design	Site preparation	Foundations	Structure erection	Building services	Commissioning
Feasibility study	1	1							
Basic design	2	X	2						
Detailed design	3		X	3					
Site preparation	4		X						
Foundations	5			X	X				
Structure erection	6					X			
Building services	7					X			
Commissioning	8						X	X	

Fig. 8.15 Hierarchy in network diagram representation

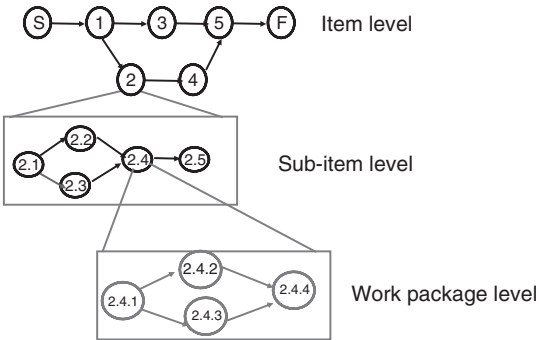


Fig. 8.16 Node representation of a task using the precedence diagramming method

Early start	Duration	Early Finish
Task name		
Late Start	Float	Late Finish

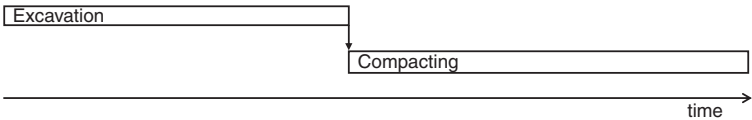


Fig. 8.17 Node bar chart with added graphical precedences

semantics, such as early/late start and finish events of activities (Fig. 8.16) and varied possibilities for setting diverse constraints between tasks (start-to-start, finish-to-finish, start-to-finish, finish-to-start). PDM nuances will be better discussed in the following paragraph coping with critical paths and time floats.

Another way of modeling network dependencies is using bar charts with precedence notation, as shown in Fig. 8.17. By adding arrows to a Gantt chart, it is possible to capture the AON precedence relationships, while being able to maintain the easy-reading of the Gantt chart.

8.3.4 Line-of-Balance Scheduling

Finally, another way for graphical representation of scheduling is the Line-of-Balance (LOB) method otherwise called *Chemin-de-Fer* from the French national railroad company (SNCF Société Nationale des Chemins de Fer) who widely uses this technique to schedule linear works such as railroad tracks, roads, and tunnels.

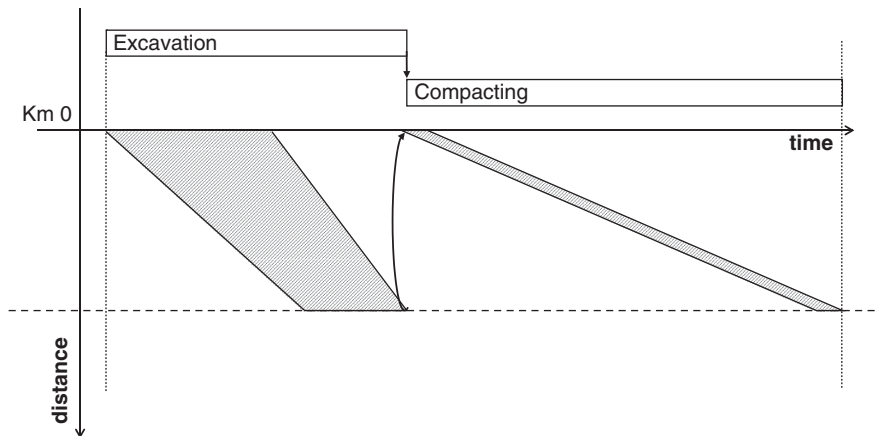


Fig. 8.18 Line-of-balance representation of scheduling

In a LOB graph, time is usually plotted on the horizontal axis and space on the vertical one. This diagram allows for representing the production rate of an activity: the slope of the production line is expressed in terms of units of distance per time (i.e. km/day).

From the example in Fig. 8.18, it is also clear that the production rate for each kilometer of excavation is variable depending on several parameters: as distance increases, time to perform excavation decreases. This may depend on several factors such as use of more resources, decrease in volumes of excavation (the dig may be less deep or narrower), or/and more efficient technologies.

8.4 Critical Path Method

There are different scheduling practices depending on whether the duration of activities is considered to be deterministic or probabilistic. Under the deterministic assumption, the most used is the Critical Path Method (CPM) and its strictly derived Precedence Diagramming Method (PDM).

The CPM consists of specifying the activities to be carried out and its associated information (such as duration) and running a scheduling algorithm in order to yield some scheduling recommendations and constraints.

The CPM runs on a network-based scheduling system. The basic steps to follow are: define activities from WBS work packages, estimate the cost, duration and resources for each one of the activities and define the precedence relationships between them. Once all is clearly defined, the system needs to be iterated in order to optimize and manage the network, using the CPM algorithm. If the results obtained are acceptable, the iteration must stop. Otherwise, some extra dependencies need to be added or some additional resources need to be considered.

The CPM algorithm runs either on AOA diagrams or on AON diagrams and it computes Early and Late Finish as well as Early and Late Start for each node. Late

Table 8.4 Precedence table of a roof construction project

#	Task name	Duration (weeks)	Predecessor
1	Site preparation	2	
2	Excavation and Foundations	7	1
3	Structures	2	2
4	Roofing	2	3
5	Enclosures	4	3
6	Building services	5	5
7	Finishing	6	4; 6

Start and Late Finish for each activity is defined as those latest dates to start or complete an activity without delaying the project duration as a whole.

For each activity, the difference between the Late Start and the Early Start (as well as between Late Finish and Early Finish) constitutes the so-called “Float”.

The CPM algorithm consists of two phases or passes:

- Forward pass determines Early Start and Finish of activities. Because all preceding activities must finish before a successor, early start of a given node is the maximum of early finishes of preceding nodes. As a practical example, the forward pass determines the shortest time to complete a sequence of tasks.
- Backward pass determines Late Start and Finish dates. Because preceding activity must finish before any following activity, late finish of a given activity is minimum of late starts of successors. In practice, given the final completion time of a sequence of tasks, the backward pass allows calculating the latest point in time the sequence has to be initiated.

Both notions are quite common-sense reasoning that we use all the time for daily life tasks (e.g. we use the forward pass to figure out what is the earliest time we could meet someone, or use the backward pass to know at what time we need to leave for making an airplane on time).

Below is an example for the construction of a small residential unit. Consider the project described with the precedence matrix in Table 8.4.

With these tasks and their predecessors in mind, the network diagram looks like the one in Fig. 8.19:

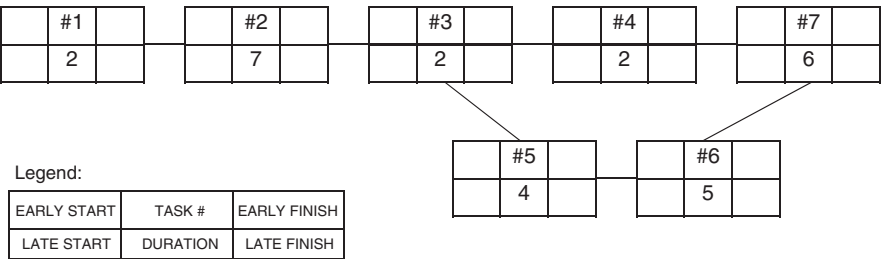


Fig. 8.19 Network diagram of the roof construction project presented in Table 8.3

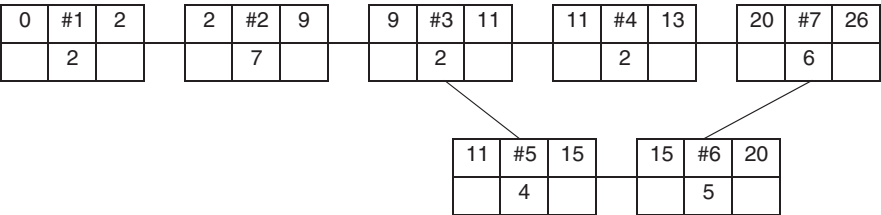


Fig. 8.20 Forward pass allows for calculating early start and finish dates of activities

Some conclusions can be extracted from application of the forward pass principle in Fig. 8.20: the Early Finish date of the project is 26 weeks.

The next Fig. 8.21 shows the second phase of the CPM algorithm: the backward pass. Now, as we know the durations of the activities, we subtract them from the Late Finish to get their Late Starts.

With also Late Start and Finish dates in hand, it is possible to calculate floats for each one of the activities. In the project above, for example, activity #7 has no float, while activity #6 has a 7-week float.

After all the network is solved, we just need to look at that path whose activities have no float. This path is defined as the *Critical Path* (CP) and it is the longest of all paths in the network system. In the example above, the CP is the one comprised of activities #1-2-3-5-6-7.

In all projects where the total finish date is calculated as the late duration of the network, there is at least one critical path, and the activities in this path must be completed on time, otherwise the entire project will be delayed.

Sometimes, projects have a later contract deadline than the one obtained from solving the network. In such fortunate circumstances, there is no critical path in a strict sense. Yet, it is opportune that a new project timeline is set to be finished with the longest path of activities, so that a time buffer, from timeline completion to contract deadline, is available as a contingency.

The CP determines the minimum time required to execute a project. However, two aspects of this algorithm need to be considered: first, we have to pay special attention to near-critical paths (those paths with low floats), and second, the critical path evolves over time as activity actual durations unfold. Finally, since there is no

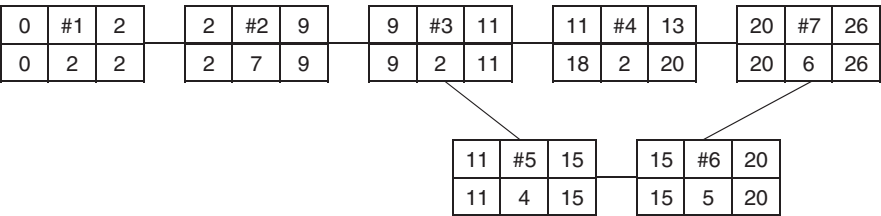


Fig. 8.21 Backward pass allows for calculating late start and finish dates of activities

float in the critical path, there is no flexibility and, thus, some contingency buffer should be planned ahead.

Therefore, the notion of float assumes great importance. Intuitively, the float measures the leeway in scheduling: it is somewhat a degree of freedom in timing for performing a task.

There are two different types of float:

1. the *Total Float* of a path, represents the maximum amount of time that will not delay the overall project;
2. the *Free Float*, for each activity, represents the amount of time an activity can be delayed without delaying the start of its successors. Closely similar is the Independent Float, which is defined as the Free Float in the worst-case finish of all its predecessors.

In light of this definition, a critical path is that with a total float equal to 0. Those paths with a total float greater than 0 are called sub-critical and those with a float less than 0 are called hyper-critical. In this latter case, it is necessary, either by increasing the number of resources and the productivity rate or by changing the equipment and the technology, to expedite the network and bring the hyper-critical paths to critical, at least.

One way to rank all the paths in order to know which ones need more attention is by using the priority index, defined as:

$$\lambda = \frac{\alpha_2 - \beta}{\alpha_2 - \alpha_1} (100\%)$$

where α_1 is the minimum total float, α_2 is the maximum total float and β is the float of the considered path. In this way, we can classify all paths and pay attention as λ is high.

Consider this Example. A Project has 4 Paths with the Following Total Floats:

Path 1: $b_1 = 0$ days, which is equal to minimum total float α_1

Path 2: $b_2 = 10$ days, which is equal to maximum total float α_2

Path 3: $b_3 = 5$ days

Path 4: $b_4 = 2$ days

The priority indexes for the four paths will be:

$\lambda_1 = (10-0)/(10-0) = 100\%$, Critical Path

$\lambda_2 = (10-10)/(10-0) = 0\%$, the less critical path of the project

$\lambda_3 = (10-5)/(10-0) = 50\%$, medium critical

$\lambda_4 = (10-2)/(10-0) = 80\%$, near-critical

8.4.1 Float Ownership

Tensions and disputes often occur between owners and contractors over the “ownership” of the float. The problem arises when, on the one hand, owners seek to push contractors on a tight (and sometimes unrealistic) schedule, while, on the other hand, contractors seek flexibility in their projects.

Thus, the owner seeks lower risks by getting the work done the earliest (because too many late starts may jeopardize the overall project duration) and, in this endeavor, the owner may impose unrealistic short schedule to the contractor. The owner may also use the contract to limit the flexibility of the contractor by specifying the owner rights to use the float, to select the scheduling procedures or to object to unreasonable durations.

On the other side, the contractor will try to artificially create a schedule with many critical and near-critical paths by deliberately inflating durations (so that they can charge extra money if the owner requires them to speed up) or by inserting artificial precedence constraints (so that the contractor can charge an extra amount of money if the owner requires them to change “the way of doing things”).

Fisk (2003, pp. 362–364) proposes a proper distribution of floats that may help in solving tensions and better understand who is responsible for delays, as follows.

There are two common ways of distributing the available float all over the non-critical activities: straight-linear and distributed. To present those methods, consider the example in Fig. 8.22.

The critical path (marked grey in the figure) has duration of 30 days. The black-marked path has a total duration of 24 days and it is comprised of activities #1 (duration 3 days) and #2 (21 days). Therefore, the total float for the black path is 6 days (30–24 days).

A first way to distribute the total float is by using a straight-line method. That is, distribute the float proportionally to the duration of each activity of the path. The formula for this case is:

$$\text{Distributed Float} = \text{Activity Duration} / \text{Path Duration} \times \text{Total Float}$$

So the distributed float of activity #1 is: $3/24 \times 6 = 0.75$ days; for activity #2, the distributed float is: $21/24 \times 24 = 5.25$ days. Thus, the new Late Finish for activity

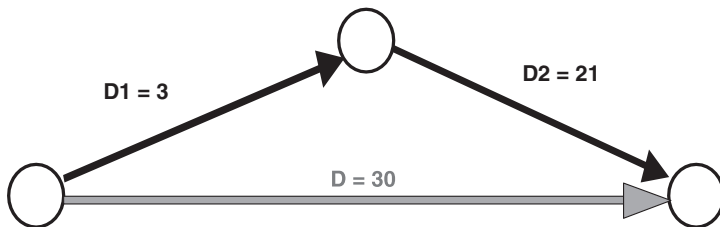


Fig. 8.22 Backward pass allows for calculating late start and finish dates of activities

#1 would be 3.75 days and the new Early Start for activity #2 would be 24.75 days (30 days, which is the project duration, minus the distributed float).

Another way to distribute the total float is by using a float-sensitive distribution. That is, considering the length of the activities as well as the inherent risk in the activity itself. The formula in this case would be:

$$\text{Distributed Float} = \text{average (Activity Duration / Path Duration; } f(\text{risk})) \times \text{Total Float}$$

In the example above, if activity #1 is the design phase of a project, with an 80% risk of delay, and activity #2 is the construction phase of the same project, with a 20% risk of delaying the project then:

$$\text{DF(A1)} = \text{average (3/24; 0.8)} \times 6 = 2.775$$

$$\text{DF(A2)} = \text{average (21/24; 0.2)} \times 6 = 3.225$$

So, the new Late Finish for activity #1 is 5.775 days, whereas the new Early Finish for activity #2 is 26.775 days (30 days minus the distribution float).

8.5 Precedence Diagramming Method

As previously discussed, PDM is an AON network method and goes beyond the CPM by including other inter-activities relationships such as Start-to-Start (SS), Start-to-Finish (SF) and Finish-to-Finish (FF) apart from the conventional Finish-to-Start (FS).

It also includes the possibility of adding “lags” or “leads” (negative “lags”) between activities. If we consider that there is a relationship XY (SS, SF, FS or FF) with lag “t” between activities A and B, then event Y of activity B can occur no earlier than t units after event X occurs for activity A.

Figure 8.23 illustrates different situations of leads and lags.

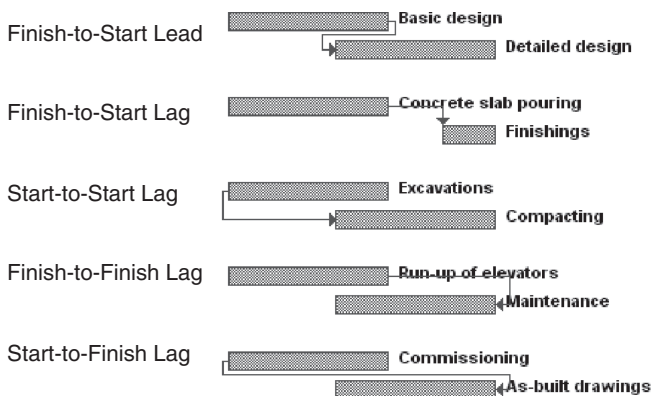
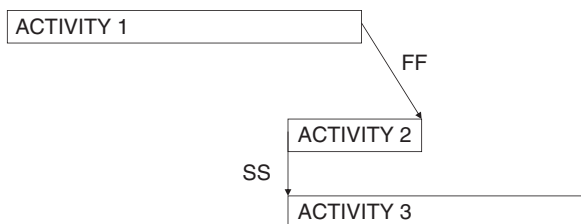


Fig. 8.23 Relationships, lags and leads in a CPM schedule

Fig. 8.24 Example of a reverse critical activity



Nodes now are no longer simply vertices in the graph. Here, an arrow on the left side of the node indicates a Start Relationship, whereas an arrow on the right side of the node indicates a Finish relationship.

In the PDM, the user can also add some constraints as in the CPM by assigning a fix date to a particular activity (it works as a milestone). One just needs to remember that milestones are given priority over relationships or other kind of links, so pay special attention to give “reachable” milestones. Otherwise, the links one may propose will be broken. Also, the user can set dates under the form of “must start/finish” constraints or as-late-as-possible calculations (e.g. must start on, no early than, etc.).

Some caveat of PDM need to be pointed out. It is important that the user clearly understands all the different relationships between activities, especially concerning the “lead” and “lag” concepts, which usually lack a specific standard and change from software to software. It is also important to stress that for a same activity there may be two differing floats: the Start Float (Late Start – Early Start) and the Finish Float (Late Finish – Early Finish).

As far as the CP under a PDM notation is concerned, choices on the relationships between activities clearly impact the critical path and tracing the critical path may be difficult for various reasons. For example, non-critical activities may have a critical start or finish date. Also, the critical path of the network may go backward through an activity, with the result that increasing the activity time may actually decrease the project completion time. Such an activity is called “reverse critical” and this happens when the critical path enters the completion of an activity through a finish constraint, continues backward through the activity, and leaves through a start constraint, as in the example drawn in Fig. 8.24 (the longer Activity 2 is, the smaller the critical path duration – and the quicker the project can be completed):

Furthermore, as far as different software packages display the critical path differently, it is of great importance for the scheduler to use the software package as a tool and not to completely rely on its outcomes (e.g.: Microsoft Project displays as-late-as-possible constrained activities as critical if the project is scheduled from the start date).

8.6 Resource-Based Scheduling

This section completes the CPM technique, according to which an optimal duration can be determined as a result of the optimization of the time-cost tradeoff. This is still an open problem and involves the application of heuristic algorithms to find the minimum total cost consistent with the project optimal duration.

This section also addresses situations that involve resource optimization. Resource-constrained scheduling applies whenever there are limited resources available and the competition for these resources among the project activities is keen. In short, the time-cost optimized schedule can provide a bad utilization of resources with high peaks and under loaded periods. Resource leveling aims to minimize the period-by-period variations in resource loading by shifting tasks within the allowed slacks.

Another problem is about resolving periods with over allocated resources: heuristic models require priority rules to establish which activity takes precedence in resource usage and which one can be postponed or get a longer duration.

8.6.1 Time-Cost Schedule Optimization with CPM

Let us recall the critical path method: once activities are defined from WBS work packages and durations for each activity as well as cost and resources are estimated, then it is possible to plot the network and perform the CPM scheduling to estimate time, cost, and resource usage over the whole project.

If the total duration is compliant with the contract baseline, the schedule is terminated. If it is not acceptable, it is needed to impose other dependencies or added resources in order to reduce the project total duration (*“project crashing”*). Indeed, so far, scheduling has been referred to as time allocation; but, since time is a function of resource usage and the inherent related cost, possible tradeoffs exist between time and cost, and, more generally, between time and resources.

There are several ways to crash a project: supplying a higher number of human resources, using overtime or multiple shifts, and changing the technology.

Adding additional resources may not be possible or effective for several reasons. First, the available supply of a limited resource might be exhausted. Second, the wage for addition resources may be higher, or the resources might come in packages, such as a crew of 3 electricians. Thirdly, the productivity of additional resources might not be as high as the original resources. Training may be required, or limitations such as space or the nature of the task at hand might cause a slowdown of work.

Increasing the number of shifts avoids the problem of reduced productivity due to crowding, but has problems of its own, such as the increased cost of labor at night, and the natural fact that people are less productive over night.

Overtime is an option, but worker productivity drops dramatically after 40 h a week. Productivity rebounds slightly for a few weeks, but then drops off again. Overtime wages are also more costly than standard wages.

A change in technology can also reduce time and costs, but also has some drawbacks. More efficient equipment is most likely more expensive. Changing technology in a project might also create the need for some redesign or rework. In the end, the time saved by a technology change might not be linear to the additional costs incurred. One thing that needs to be considered when scheduling is the type of task at hand. If the task has a fixed duration, such as the curing time of concrete, it

cannot be crashed. In order to save time in this scenario, the technology might have to be changed to quick setting concrete.

As a result, project crashing inevitably increases the cost of the project: we call “crashed”, or accelerated, cost the cost associated with a crashed, or accelerated, duration of the network.

With this notion in hand, it is possible to optimize the network using the CPM.

The first task is to schedule the project using a “normal” time frame and associated “normal” cost. The second step is to crash the project. This is done for two reasons: to reduce the normal finish date to less than the contract deadline if needed, and to establish the length of the project at minimum costs.

Crashing a project consists of reducing the time that it takes to complete the project. Usually this raises the cost of the project.

In most cases, there are a few portions of the project that can be crashed, resulting in a high reduction in project time, but relatively small increases in cost. As more and more tasks get crashed, the relative gain in time to the increase in costs gets smaller. At some point it is no longer valuable to trade time for costs. The chart of Fig. 8.25 gives an example of this.

When crashing a project, it is important to look at the critical path. There is no reason to crash tasks not on the critical path, because no time on the project will be saved, resulting in more cost with no time benefit. It is also important to watch how the critical path changes during crashing. After crashing a few tasks, the critical path might change, and then tasks not originally on the critical path will need to be crashed to reduce the project time. This makes a big difference in construction project management, as many managers would crash all the tasks in a project to save time, when it is unnecessary to crash many of the tasks, as in the R point illustrated in Fig. 8.26.

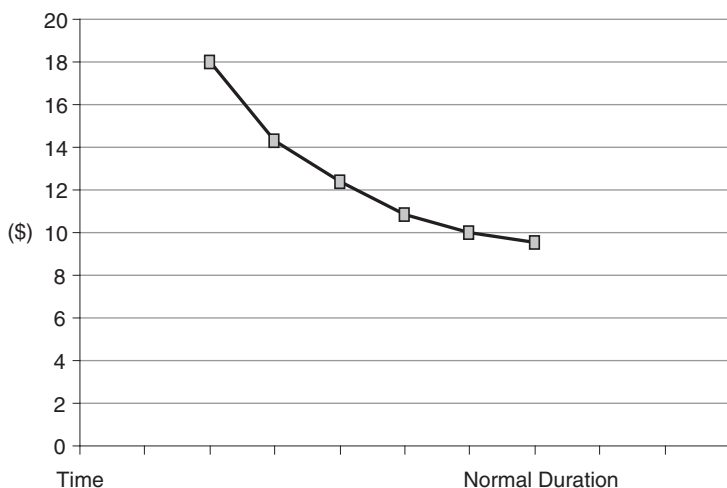


Fig. 8.25 Crashing curve

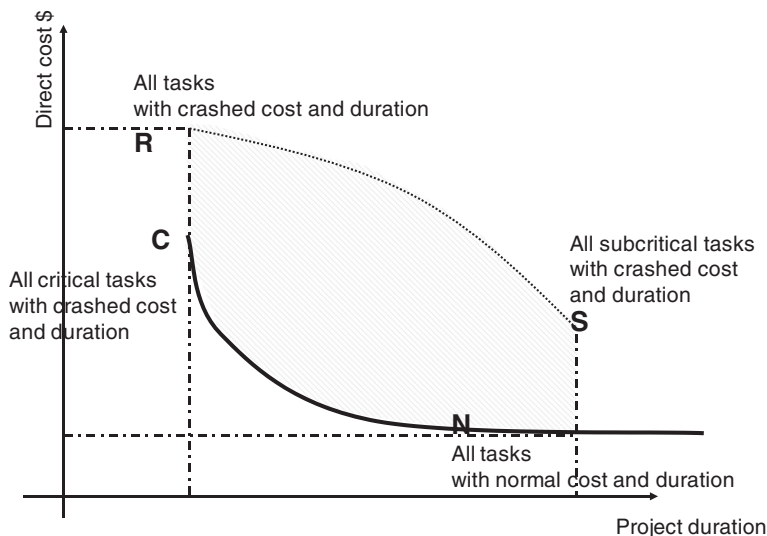


Fig. 8.26 Time-cost configuration space resulting from all possible crashings of the project duration

More generally, Fig. 8.26 shows the time-cost configuration space resulting from all possible crashing simulations. It illustrates the importance of speeding up only the tasks that are on the critical path, since there may be ineffective situations when a shorter duration of the project may be obtained with lower cost (e.g. point R versus point C of the graph).

In other words, the proper “crashing curve” is the one that is not dominated by more efficient curves: the crashing curve is a Pareto-optimal solution because it minimizes the direct cost associated with a given duration of the project. Also, it is worth to note that the crashing curve slope increases as the duration is crashed up to its shortest date; in fact, crashing is limited by technology and resources to a minimum duration. On the opposite side, a longer duration than the one calculated with a minimum/normal usage of resources does not lead to lower costs.

At this point, it is convenient to take a look at the curve of the total cost, which sums the direct cost with project and corporate overhead costs. Because overhead increases as the project continues off, there is a minimum point of the total cost curve, as shown in Fig. 8.27. The minimum point determines the “*optimal duration*” referred to as the length of the project consistent with the minimum total cost for the firm.

A better insight of the graph above suggest that there may be reasons to pay for penalties up to the optimal duration of the project: from the date when contract penalties are due to the optimal finish date it is less expensive to sustain overhead and pay the liquidated damages than to afford the cost of crashing. In general, later than the optimal duration the daily crashing cost is less than daily overhead.

The optimal duration allows define the proper amount of crashing. But, whether or not accelerating the entire project, crashing is suitable for different algorithms. In

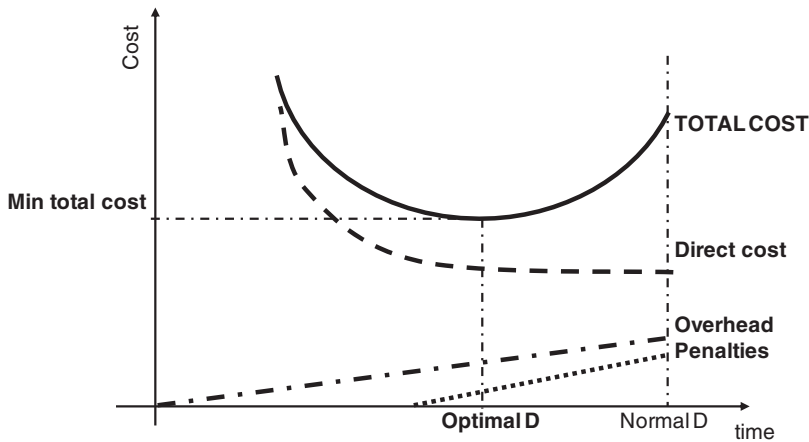


Fig. 8.27 Optimal duration is the one who minimizes the total cost of the project

particular, during the planning phase it may be simply opportune to crash the initial activities of the critical path, while during the project execution there is no other possibility than crash the remaining activities, from time now forward. In any case, the definition of a proper crashing algorithm is required.

If activity time-cost curves are linear, then finding the optimal duration of the project is a linear programming problem. Unfortunately, in most cases there is no straight-linear relationship between time and cost. This ends up to a non-linear programming problem for which the definition of an algorithm based on heuristics is required.

As a general rule, basic recommendations apply to schedule crashing heuristics:

- focus on critical path, which means that only critical activities should be crashed (note that as the crash time amount increases, the number of critical activities increases as well);
- select the less expensive way to do it, that is crash first activities that result in a smaller increase in costs
- trade time for money on non-critical activities: the activity time should be lengthened to reduce costs if possible. Non-critical paths can be extended within the available float, reducing the costs of the task. As long as the task is not extended beyond the available float, the project duration is not lengthened, and the indirect costs will not go up.

One of the most used crashing heuristic algorithms is the one by Kelly and Walker (1959). It states the following steps:

1. solve CPM with normal durations;
2. for critical activities find marginal cost of crashing (i.e., additional cost of shortening duration 1 time unit);

3. reduce by one time step the critical activity with the lowest marginal cost of crashing;
4. record resulting project duration and cost;
5. repeat step 3 until another path becomes critical.

8.6.2 Resource Leveling

Mainly, limitations to the schedule regard the tradeoff cost-time-resources. If the project is budget limited, then it must have duration and resource usage based on the required preset cost. In this case, resources must be leveled to reduce indirect costs and then total costs. In a time limited project, it must finish within a scheduled date thus requiring resource usage at best with minimum possible cost.

Finally, if a project is resource limited, it must not exceed a specific level of resource usage or overcome resource constraints (such as crew sequencing) so that the project duration is the shortest possible time associated with the limitation. If the normal schedule was based on some resource limitations, then crashing the schedule may not be possible, or might greatly increase costs due to working around the resource limitations. In short, the time-cost optimized schedule can suggest an impossible schedule or provide a bad utilization of resources with high peaks and under loaded periods.

To solve the problem, “*resource leveling*” may help to reduce the period-by-period oscillations in resource loading by shifting tasks within the allowed float. During the course of a project, a more steady usage of resources leads to lower resource costs. This is due to reducing the costs of hiring, training, and firing human labor, material storage, and equipment rental and storage. Resource leveling is done by moving tasks around in the schedule and reorganizing the floats.

Figure 8.28 shows an example of this. The original resource-load profile (dashed line profile) can be leveled if the non-critical activity #2 is anticipated (black-marked profile). This provides a double advantage: it avoids a later short resource peak and keeps the maximum amount of resources within 25 units over the project.

In some cases the situation arises where the resources cannot be leveled within the available float. At that point, a decision needs to be based on what is more cost-effective, whether acquiring more resources or lengthening the schedule. When resources are limited, the schedule has to be lengthened or reworked to accommodate the situation. One thing to keep in mind is that performing tasks when possible is not always the best approach. When resources are limited, the entire project schedule needs to be evaluated to find the optimal way to work around the constraints.

The following is an example of “manual workaround” (i.e. adding precedence relationships to the original network links) that is needed to respect resource constraints. Suppose that there is only one crane available to perform two overlapping activities requiring a crane. The solution is either to add a precedence link between

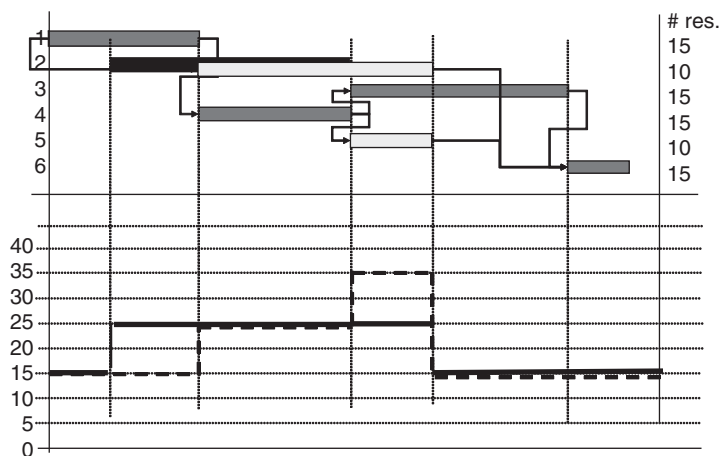


Fig. 8.28 Resource leveling within available floats

the two activities to have them performed in sequence (with twice the original duration of the path) or to buy/rent an extra crane with added cost.

8.6.3 Heuristic Scheduling Approaches

A wider aspect of resource scheduling is concerned with resource leveling under limited-resource allocation. This combined problem can be solved either through optimization or by applying heuristics algorithms.

Finding the “optimal” configuration for leveling a resource-constrained schedule is a computationally-expensive combinatorial problem. In principle, it would need to compare all possible orderings of conflicting activities. Applications on specific projects exist based on approaches such as linear programming, explicit enumeration and “Branch and Bound” methods.

Heuristics algorithms, though inconsistent with finding the optimal allocation, yet provide useful configuration of leveled resource-constrained schedules. Heuristics use some “rules of thumb” to get answer in an acceptable time. They typically reach a local minimum and do what is locally-best, but not necessarily globally-best. This means that heuristics may not optimize the project as a whole.

There are two types of resource-scheduling heuristics. The “serial methods” schedule activity-by-activity: the algorithm considers prioritized activities in order and schedules them as early as possible. Activities are assigned priority based on a number of attributes: length, resources required, slack, or the number or type of successors. Each activity needs to wait until its predecessors have been completed, and the required number of resources is available.

The “parallel methods” schedule activities by time step. At each time step, some activities are delayed as needed based on algorithm criteria such as Shortest Task

First or Longest Task First and rules like: “Wait until predecessors complete or adequate resources are available”.

Parallel methods are more commonly used in current software packages (i.e. Microsoft Project, Primavera) than serial methods.

References and Additional Resources About Planning and Scheduling

- Diamant L, Tumblin CR (1990) Construction cost estimates. Wiley, New York, NY
- Fisk ER (2003) Construction project administration, 7th edn. Pearson Prentice Hall, Upper Saddle River, NJ
- Hinze JW (2004) Construction planning and scheduling, 2nd edn. Pearson Prentice Hall, Upper Saddle River, NJ
- Kelly JE, Walker MR (1959) Critical path planning and scheduling. Mauchly Associates, Ambler, PA
- Kerzner H (2001) Project management: a systems approach to planning, scheduling, and controlling. Wiley, New York, NY
- Meredith J, Mantel S (2006) Project management: a managerial approach, 6th edn. Wiley, New York, NY
- Moder J, Phillips CR, Davis EW (1995) Project management with CPM, PERT, and precedence diagramming, 3rd edn. Van Nostrand Reinhold, New York, NY
- Associated General Contractors of America (2005) Construction estimating & bidding: theory/principles/process. AGC of America, Arlington, VA
- Patrick C (2004) Construction project planning and scheduling. Pearson Prentice Hall, Upper Saddle River, NJ



<http://www.springer.com/978-3-642-17091-1>

Project Management for Facility Constructions

A Guide for Engineers and Architects

De Marco, A.

2011, VIII, 189 p., Hardcover

ISBN: 978-3-642-17091-1