12

Resource Management

Chapter Outline

Chapter Objectives

After completing this chapter, you should be able to:

1. Recognize the variety of constraints that can affect a project, making scheduling and planning difficult.
2. Understand how to apply resource-loading techniques to project schedules to identify potential resource overallocation situations.
3. Apply resource-leveling procedures to project activities over the baseline schedule using appropriate prioritization heuristics.
4. Follow the steps necessary to effectively smooth resource requirements across the project life cycle.
5. Apply resource management within a multiproject environment.

PROJECT MANAGEMENT BODY OF KNOWLEDGE CORE CONCEPTS COVERED IN THIS CHAPTER

1. Estimate Activity Resources (PMBoK sec. 6.4)
2. Plan Human Resource Management (PMBoK sec. 9.1)

PROJECT PROFILE

Hong Kong Connects to the World's Longest Natural Gas Pipeline

As one of the world's most populous cities, Hong Kong needs a consistent stream of clean energy resources to supply its 7.1 million residents with power. The city relies on a mix of fuels, but most recently, has been able to make use of an environmentally friendly supply of energy to keep the city's power needs satisfied. That energy source is natural gas. Since 1996, Hong Kong's Black Point Power Station had drawn natural gas from the reserves of the Yacheng 13-1 gas field in Hainan, a nearby Chinese province. But as those reserves began to deplete, it became clear by the end of the last decade that the energy companies managing Hong Kong's resource needs required new options. The company, CLP/CAPCO (a joint venture of ExxonMobil Energy and CLP Power Hong Kong), began looking into alternative arrangements—not only to maintain a consistent supply of natural gas, but also to comply with the tightened emission standards that will be required by the Hong Kong Special Administrative Region (HKSAR) Government in 2015.

In 2008, Hong Kong officials and the Central Government of the People's Republic of China signed a memorandum of understanding on energy cooperation, which identified three new gas sources from which mainland China could supply gas to Hong Kong. One of the sources is the Second West-East Gas Pipeline. The Second West-East Gas Pipeline is the single biggest energy investment project in the history of the country, stretching nearly 9,000 kilometers. Begun in 2008, it is already powering cities across the mainland. It starts in Xinjiang, where it connects to Turkmenistan's Central Asia-China Gas Pipeline, and crosses 15 provinces, autonomous regions, and municipalities. It can carry 30 billion cubic meters of gas a year and was already supplying energy for some 500 million Chinese citizens across the country. Built by a workforce of 50,000 people, the pipeline passes through mountains, deserts, and swamps, and crosses 60 hills and mountains and approximately 190 rivers.

Just connecting the pipeline network from mainland China to Hong Kong presented numerous complex challenges to all involved.

- Regulations: Because it crossed the border between mainland China and Hong Kong, the project team had to acquire permits from both jurisdictions. The project had to fulfill differing practices and statutory approval processes between the jurisdictions.

FIGURE 12.1 Workers Inside the Natural Gas Pipeline

Source: Yan Ping/Xinhua Press/Corbis

(continued)

- Communications: The various working teams used several different languages, and all of the parties involved had different requirements for documentation and reporting. The teams predominantly used English, Putonghua, and Cantonese. However, they used English and Chinese for documents and PowerPoint presentations. The project team also had to manage a multitude of stakeholders, including over 30 authorities in both jurisdictions.
- Resource management: Acquiring and scheduling thousands of skilled workers and engineers throughout the pipeline extension project required a massively complex project plan and a clear method for scheduling resources, especially during critical parts of the project, including laying underwater piping.
- Environmental requirements: The project needed to fulfill stringent environmental requirements for the two jurisdictions. The project managers instituted a robust monitoring and audit program during the project execution phase, with intensive water quality monitoring, marine mammal monitoring, and site inspections. Mitigation measures also included the deployment of silt curtains and limitations on working speed during marine dredging and jetting operations.
- Quality control: Every section of the pipeline was meticulously checked. Each weld joint had to pass an automatic ultrasonic testing. The entire pipeline, including its coating and corrosion protection system, was thoroughly inspected before being laid into the seabed.

Laying of the undersea pipeline was a difficult undertaking due to a number of physical constraints. The project required a 20-kilometer section pass beneath three major shipping channels—Dachan Fairway, Tonggu Channel, and Urmston Road—the last of which is one of the world's busiest marine channels used by more than 400 vessels a day, including ocean-going vessels. Getting a permit to put the pipeline beneath the Urmston Road took three months of planning and discussions with Hong Kong and Shenzhen officials. Once approved, the actual laying of the pipeline took just three days. The operation involved hundreds of engineers, pipefitters, and welders working from a sophisticated water-pipe laying and lifting barge working in shallow and deep waters. For example, the branch line links Dachan Island, off Shenzhen, with Hong Kong's Black Point power station. It took 1,600 carbon steel pipes, each nearly 40 feet long and weighing approximately 13 tons.

The project cost the equivalent of $23 billion dollars and is operated by the China National Petroleum Corporation. It will continue to provide a new source of gas to Hong Kong, replacing the nearly exhausted alternatives, while improving Hong Kong's consumption of carbon fuels and ensuring a "greener" supply of energy well into the future.[1]

INTRODUCTION

As noted in Chapter 1, one of the defining characteristics of projects is the constraints, or limitations, under which they are expected to operate. The number one constraint is the availability of resources, both money and people, at the critical times when they are needed. Initial project cost estimation and budgeting—those activities that nail down resources—are extremely important elements in project management. When these two are performed well, they ensure appropriate resources for the project as it progresses downstream.

In Chapters 9 and 10 on project scheduling, we saw that network diagrams, activity duration estimates, and comprehensive schedules can all be developed without serious discussion of the availability of the resources. It was not until Chapter 11, on Agile and Critical Chain Scheduling, that resource availability came up as a prerequisite for accurate scheduling. Organizational reality, of course, is very different. If projects are indeed defined by their **resource constraints**, any attempt to create a reasonable project schedule must pass the test of resource availability. So, effective project scheduling is really a multistep process. After the actual network has been constructed, the second stage must always be to check it against the resources that drive each activity. The availability of appropriate resources *always* has a direct bearing on the duration of project activities.

In this chapter, we are going to explore the concept of resource planning and management. Gaining a better understanding of how resource management fits into the overall scheme of project planning and scheduling gives us a prominent advantage when the time comes to take all those carefully laid plans and actually make them work. The chapter will be divided into two principal sections: resource constraints and resource management.

12.1 THE BASICS OF RESOURCE CONSTRAINTS

Probably the most common type of project constraint revolves around the availability of human resources to perform the project. As we have noted, one of the key methods for shortening project durations is to move as many activities as possible out of serial paths and into parallel ones.

This approach assumes, of course, that staff is free to support the performance of multiple activities at the same time (the idea behind parallel work). In cases in which we do not have sufficient people or other critical resources, we simply cannot work in a parallel mode. When projects are created without allowing for sufficient human resources, project teams are immediately placed in a difficult, reactive position. Personnel are multitasked with their other assignments, are expected to work long hours, and may not receive adequate training. Trade-offs between the duration of project activities (and usually the project's overall schedule) and resource availability are the natural result.

In some situations, the **physical constraints** surrounding a project may be a source of serious concern for the company attempting to create the deliverable. Environmental or contractual issues can create some truly memorable problems; for example, the Philippine government contracted to develop a nuclear power plant for the city of Manila. Bizarrely, the site selected for its construction was against the backdrop of Mount Natib, a volcano on the outskirts of the city. As construction proceeded, environmentalists rightly condemned the choice, arguing that seismic activity could displace the operating systems of the reactors and lead to catastrophic results. Eventually, a compromise solution was reached, in which the energy source for the power plant was to be converted from nuclear to coal. With the myriad problems the project faced, it became known as the "$2.2 Billion Nuclear Fiasco."[2] This case is an extreme example, but as we will continue to see, many real problems can accrue from taking a difficult project and attempting to develop it in hazardous or difficult physical conditions.

Materials are a common project resource that must be considered in scheduling. This is most obvious in a situation where a physical asset is to be created, such as a bridge, building, or other infrastructure project. Clearly, having a stockpile of a sufficient quantity of the various resources needed to complete the project steps is a key consideration in estimating task duration times.

Most projects are subject to highly constrained (fixed) budgets. Is there sufficient working capital to ensure that the project can be completed in the time frame allowed? It is a safe bet to assume that any project without an adequate budget is doomed.

Many projects require technical or specific types of equipment to make them successful. In developing a new magazine concept, for example, a project team may need leading-edge computers with great graphics software to create glitz and glamour. Equipment scheduling is equally important. When equipment is shared across departments, it should be available at the precise time points in the project when it is needed. In house construction, for example, the cement mixer must be on site within a few days after the ground has been excavated and footers dug.

Time and Resource Scarcity

In the **time-constrained project**, the work must be finished by a certain time or date, as efficiently as possible. If necessary, additional resources will be added to the project to hit the critical "launch window." Obviously, the project should be completed without excessive resource usage, but this concern is clearly secondary to the ultimate objective of completing the project on time. For example, projects aimed at specific commercial launch or those in which late delivery will incur high penalties are often time constrained.

In the **resource-constrained project**, the work must not exceed some predetermined level of resource use within the organization. While the project is to be completed as rapidly as possible, speed is not the ultimate goal. The chief factor driving the project is to minimize resource usage. In this example, project completion delays may be acceptable when balanced against overapplication of resources.

The **mixed-constraint project** is primarily resource constrained but may contain some activities or work package elements that are time constrained to a greater degree. For example, if critical delivery dates must be met for some project subcomponents, they may be viewed as time constrained within the overall, resource-constrained project. In these circumstances, the project team must develop a schedule and resource management plan that works to ensure the minimization of resources overall while allocating levels necessary to achieve deadlines within some project components.

There is, for almost all projects, usually a dominant constraint that serves as the final arbiter of project decisions. Focusing on the critical constraint, whether it is resource-based or time-based, serves as a key starting point to putting together a resource-loaded schedule that is reasonable, mirrors corporate goals and objectives, and is attainable.[3]

The challenge of optimally scheduling resources across the project's network activity diagram quickly becomes highly complex. On the one hand, we are attempting to create an efficient activity network that schedules activities in parallel and ensures the shortest development cycle possible. At the same time, however, we inevitably face the problem of finding and providing the resources necessary to achieve these optimistic and aggressive schedules. We are constantly aware of the need to juggle schedule with resource availability, trying to identify the optimal solution to this combinatorial problem. There are two equally critical challenges to be faced: (1) the identification and acquisition of necessary project resources, and (2) their proper scheduling or sequencing across the project baseline.[4]

EXAMPLE 12.1 Working with Project Constraints

Here is an example that shows what project teams face when they attempt to manage project resources. Suppose we created a simple project activity network based on the information given in Table 12.1. Figure 12.2 demonstrates a partial network diagram, created with Microsoft Project 2013. Note that the first three activities have each been assigned a duration of five days, so activities B and C* are set to begin on the same date, following completion of activity A. Strictly from a schedule-development point of view, there may be nothing wrong with this sequence; unfortunately, the project manager set up the network in such a way that both these activities require the special skills of only one member of the project team. For that person to accomplish both tasks simultaneously, huge amounts of overtime are required or adjustments will need to be made to the estimated time to completion for both tasks. In short, we have a case of misallocated resources within the schedule baseline. The result is to force the project team to make a trade-off decision: either increase budgeted costs for performing these activities or extend the schedule to allow for the extra time needed to do both jobs at the same time. Either option costs the project two things it can least afford: time and money.

TABLE 12.1 Activity Precedence Table

Activity	Description	Duration	Predecessors	Member Assigned
A	Assign Bids	5 days	None	Tom
B	Document Awards	5 days	A	Jeff
C	Calculate Costs	5 days	A	Jeff
D	Select Winning Bid	1 day	B, C	Sue
E	Develop PR Campaign	4 days	D	Carol

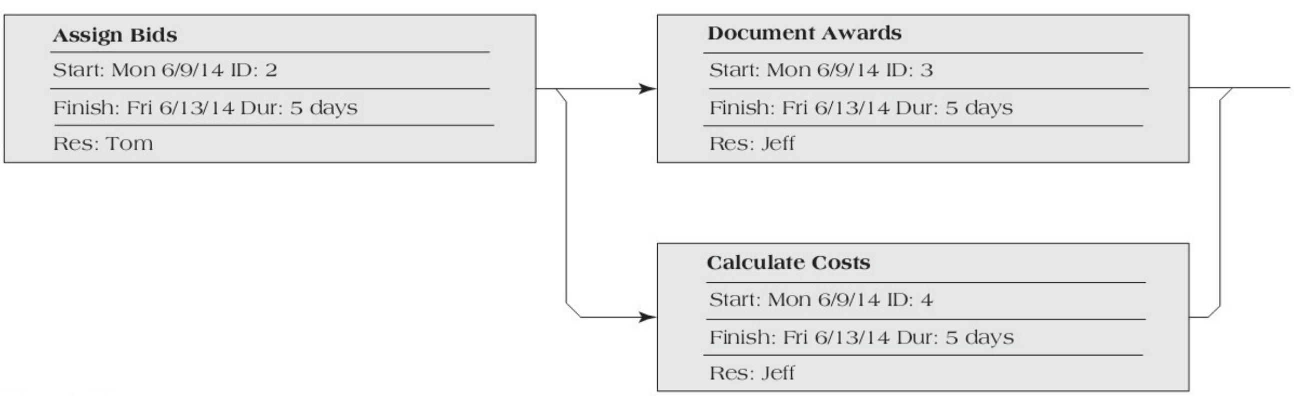
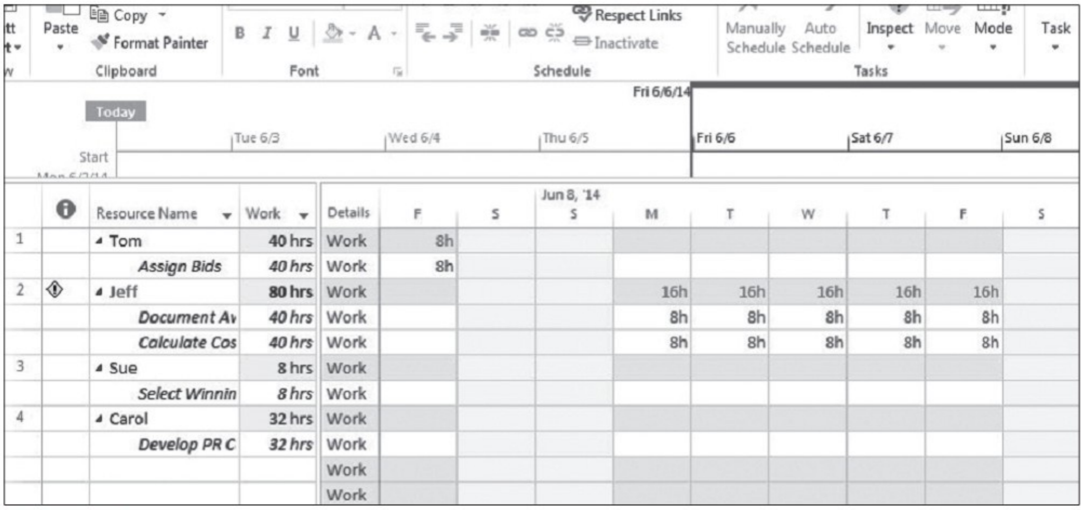


FIGURE 12.2 Sample Activity Network with Conflicts

* Microsoft Project 2013 identifies activities B and C as tasks 2 and 3, respectively.



		Resource Name	Work	Details	F	S	Jun 8, '14 S	M	T	W	T	F	S
1		Tom	40 hrs	Work	8h								
		Assign Bids	*40 hrs*	Work	8h								
2		Jeff	80 hrs	Work				16h	16h	16h	16h	16h	
		Document Av	*40 hrs*	Work				8h	8h	8h	8h	8h	
		Calculate Cos	*40 hrs*	Work				8h	8h	8h	8h	8h	
3		Sue	8 hrs	Work									
		Select Winnin	*8 hrs*	Work									
4		Carol	32 hrs	Work									
		Develop PR C	*32 hrs*	Work									
				Work									
				Work									

FIGURE 12.3 **Resource Usage Table Demonstrating Overallocation**

Source: MS Project 2013, Microsoft Corporation

The best method for establishing the existence of resource conflicts across project activities uses *resource-loading charts* (described more fully in the next section) to analyze project resources against scheduled activities over the project's baseline schedule. Resource-loading charts enable the project team, scheduling the work, to check their logic in setting resource requirements for project activities. A simplified MS Project 2013 resource usage table highlighting the resource conflict found in Figure 12.2 is shown in Figure 12.3.

Note what has happened to Jeff's resource availability. The MS Project 2013 output file highlights the fact that for a five-day period, Jeff is expected to work 16 hours each day to accomplish activities B and C simultaneously. Because the schedule in Figure 12.2 did not pay sufficient attention to competing demands for his labor when the activity chart was created, the project team is now faced with the problem of having assigned his time on a grossly overallocated basis. Although simplified, this example is just one illustration of the complexity we add to project planning when we begin to couple activity network scheduling with resource allocation.

12.2 RESOURCE LOADING

The concept of **resource loading** refers to the amount of individual resources that a schedule requires during specific time periods.[5] We can *load*, or place on a detailed schedule, resources with regard to specific tasks or the overall project. As a rule of thumb, however, it is generally beneficial to do both: to create an overall project resource-loading table as well as identify the resource needs for each individual task. In practical terms, resource loading attempts to assign the appropriate resource, to the appropriate degree or amount, to each project activity.

If we correlate the simple example, shown in somewhat greater detail in Figure 12.4, with the original project Gantt chart, we can see that these important first steps are incomplete until the subsequent resource assignments are made for each project activity. In Figure 12.4, we have temporarily fixed the problem of Jeff's overallocation by adding another resource, Bob, who has become responsible for activity C, (Calculate Costs).

Once we have developed the Work Breakdown Structure and activity networks, the actual mechanics of creating a *resource-loading form* (sometimes referred to as a *resource usage calendar*) is relatively simple. All personnel are identified and their responsibility for each task is assigned. Further, we know how many hours on a per-week basis each person is available. Again, using Microsoft Project's 2013 template, we can create the resource usage table to reflect each of these pieces of information (see Figure 12.5).

Information in the **resource usage table** shown in Figure 12.5 includes the project team members, the tasks to which they have been assigned, and the time each activity is expected to take

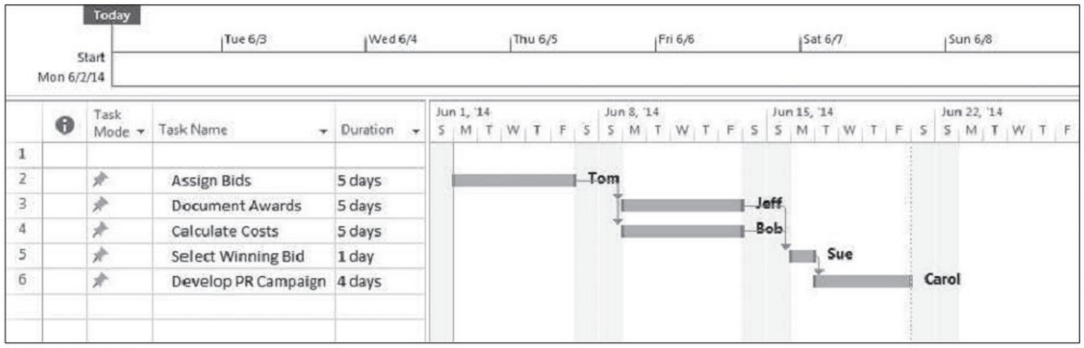


FIGURE 12.4 **Sample Project Activity Network and Gantt Charts**

Source: MS Project 2013, Microsoft Corporation

ew Clipboard Font Schedule

Today

Tue 6/3 Wed 6/4 Thu 6/5 Fri 6/6

Start Mon 6/2/14

	Resource Name	Work	Details	June 1 6/1	6/8	June 11 6/15	June 21 6/22	July 1 6/29
1	Tom	40 hrs	Work	40h				
	Assign Bids	*40 hrs*	Work	40h				
2	Jeff	40 hrs	Work		40h			
	Document Awards	*40 hrs*	Work		40h			
3	Sue	8 hrs	Work			8h		
	Select Winning Bid	*8 hrs*	Work			8h		
4	Carol	32 hrs	Work			32h		
	Develop PR Campaign	*32 hrs*	Work			32h		
5	Bob	40 hrs	Work		40h			
	Calculate Costs	*40 hrs*	Work		40h			
			Work					

FIGURE 12.5 **Resource Usage Table**

Source: MS Project 2013, Microsoft Corporation

across the schedule baseline. In this example, we have now reallocated the personnel to cover each task, thereby eliminating the overallocation problem originally uncovered in Figure 12.3. Team members are assigned to the project on a full-time (40 hours/week) basis, and the loading of their time commitments across these project activities corresponds to the project activity network, providing, in essence, a time-phased view of the resource-loading table.

The resource usage table also can provide warning signs of overallocation of project resources. For example, suppose that Jeff was again allocated to both activities B and C, as in the example from earlier in this chapter. Simply viewing the original project schedule gives no indication of this resource overallocation. When we generate the resource usage table, however, we discover the truth (see Figure 12.6). In this example, Jeff is currently scheduled to work 80 hours over a one-week period (the week of June 8)—clearly a much-too-optimistic scenario regarding his capacity for work!

The benefit of the resource-loading process is clear; it serves as a "reality check" on the project team's original schedule. When the schedule is subjected to resource loading, the team quickly becomes aware of misallocation of personnel, overallocation of team members, and, in some cases, lack of needed resources. Hence, the resource-loading process may point to obvious flaws in the original project WBS and schedule. How best to respond to resource-loading problems and other project constraints is the next question the project manager and team need to consider.

Today

Start Mon 6/2/14 | Tue 6/3 | Wed 6/4 | Thu 6/5 | Fri 6/6

		Resource Name	Work	Details	June 1 6/1	June 11 6/8	6/15	June 21 6/22	July 1 6/29
1		◢ Tom	40 hrs	Work	40h				
		Assign Bids	*40 hrs*	Work	40h				
2	◈	◢ Jeff	80 hrs	Work		80h			
		Document Awards	*40 hrs*	Work		40h			
		Calculate Costs	*40 hrs*	Work		40h			
3		◢ Sue	8 hrs	Work			8h		
		Select Winning Bid	*8 hrs*	Work			8h		
4		◢ Carol	32 hrs	Work			32h		
		Develop PR Campaign	*32 hrs*	Work			32h		
5		Bob	0 hrs	Work					
				Work					

FIGURE 12.6 Example of Resource Usage Table with Overallocation

Source: MS Project 2013, Microsoft Corporation

12.3 RESOURCE LEVELING

Resource leveling is the process that addresses the complex challenges of project constraints. With resource leveling we are required to develop procedures that minimize the effects of resource demands across the project's life cycle. Resource leveling, sometimes referred to as resource **smoothing**, has two objectives:

1. To determine the resource requirements so that they will be available at the right time
2. To allow each activity to be scheduled with the smoothest possible transition across resource usage levels

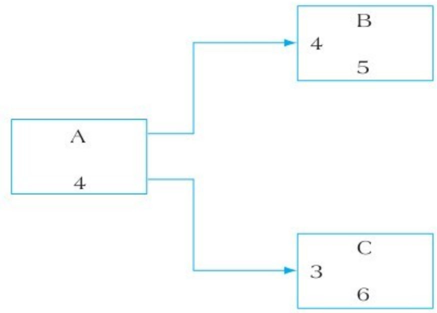
Resource leveling is useful because it allows us to create a profile of the resource requirements for project activities across the life cycle. Further, we seek to minimize fluctuations from period to period across the project. The farther in advance that we are able to anticipate and plan for resource needs, the easier it becomes to manage the natural flow from activity to activity in the project, with no downtime, while we begin searching for the resources to continue with project tasks. The key challenge consists of making prioritization decisions that assign the right amount of resources to the right activities at the right time.

Because resource management is typically a multivariate, combinatorial problem (i.e., one that is characterized by multiple solutions often involving literally dozens, hundreds, or even thousands of activity variables), the mathematically optimal solution may be difficult or infeasible to find due to the time required to solve all possible equation options. Hence, a more common approach to analyzing resource-leveling problems is to apply some **leveling heuristics**, or simplified rules of thumb, when making decisions among resource-leveling alternatives.[6]

Some simple heuristics for prioritizing resource allocation include applying resources to:

1. ***Activities with the smallest amount of slack.*** The decision rule is to select for resource priority those activities with the smallest amount of slack time. Some have argued that this decision rule is the best for making priority decisions, resulting in the smallest schedule slippage to the overall project.[7]
2. ***Activities with the smallest duration.*** Tasks are ordered from smallest duration to largest, and resources are prioritized accordingly.
3. ***Activities with the lowest activity identification number.*** (e.g., those that start earliest in the WBS). This heuristic suggests that, when in doubt, it is better to apply resources to earlier tasks first.
4. ***Activities with the most successor tasks.*** We select for resource priority those tasks that have the most tasks following behind them.
5. ***Activities requiring the most resources.*** It is common to first apply resources to those activities requiring the most support and then analyze the remaining tasks based on the availability of the additional resources

FIGURE 12.7 Sample Network Applying Resource Heuristics



Using these heuristics, let us consider a simple example and the method we would use to select the activities that get first "rights" to the resource pool. Suppose that a project has two activities (see Figure 12.7) scheduled that require the same resource at the same time. In deciding which activity should receive first priority for available resources, we can follow the heuristic logic used in the first decision rule and examine tasks B and C first in terms of their respective amount of slack time. In this case, activity C, with three days of slack, would be the best choice for prioritizing the resource. However, suppose that activities B and C both had three days of slack. Then, according to the heuristic model, we could move to the second decision rule and award the first priority to activity B. Why? Because activity B has a scheduled duration of five days as opposed to activity C's duration of six days. In the unlikely event that we discovered that a tie remained between activities B and C following the first two heuristics, we could apply the third heuristic and simply assign the resource to the task with the lowest identification number (in this case, activity B). As we will see, the implication of how resources are prioritized is significant, as it has a "ripple effect" on subsequent resource leveling throughout the remainder of the activity network.

EXAMPLE 12.2 An In-Depth Look at Resource Leveling

A more in-depth resource-leveling example illustrates the challenge project teams face when applying resource leveling to a constructed activity network. Suppose we constructed a project network diagram based on the information in Table 12.2. Using the process suggested in Chapter 9 we can also derive the early start (ES), late start (LS), early finish (EF), late finish (LF), and subsequent activity slack for each task in the network. Table 12.3 presents a complete set of data.

TABLE 12.2 Activities, Durations, and Predecessors for Sample Project

Activity	Duration	Predecessors
A	5	—
B	4	A
C	5	A
D	6	A
E	6	B
F	6	C
G	4	D
H	7	E, F
I	5	G
J	3	G
K	5	H, I, J

TABLE 12.3 Fully Developed Task Table for Sample Project

Activity	Duration	ES	EF	LS	LF	Slack
A	5	0	5	0	5	—
B	4	5	9	6	10	1
C	5	5	10	5	10	—
D	6	5	11	8	14	3
E	6	9	15	10	16	1
F	6	10	16	10	16	—
G	4	11	15	14	18	3
H	7	16	23	16	23	—
I	5	15	20	18	23	3
J	3	15	18	20	23	5
K	5	23	28	23	28	—

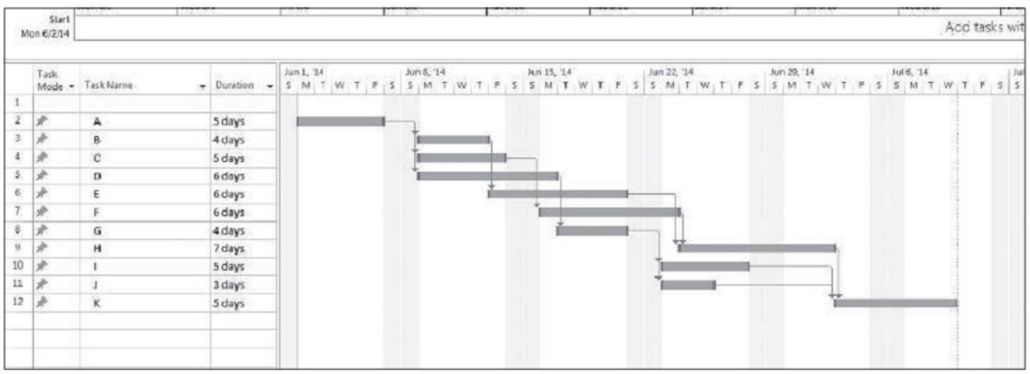


FIGURE 12.8 Gantt Chart for Sample Project

Source: MS Project 2013, Microsoft Corporation

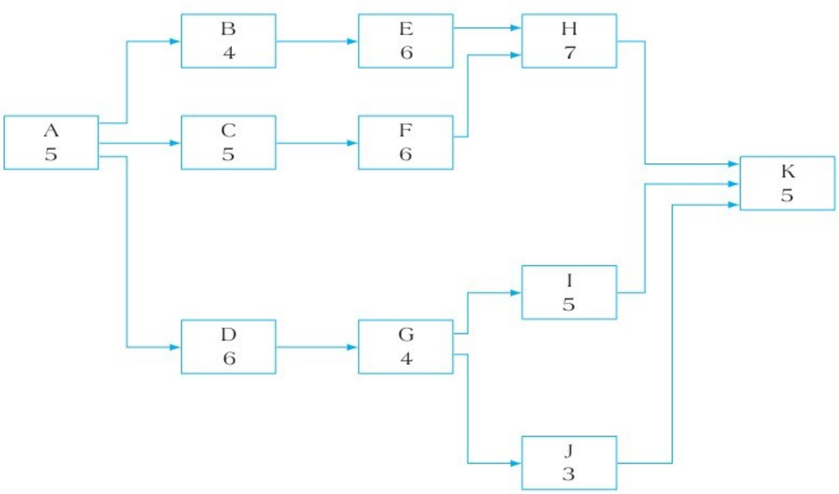


FIGURE 12.9 Sample Project Network

Table 12.3 identifies the network critical path as A – C – F – H – K. Figure 12.8 presents a simplified project Gantt chart that corresponds to the activities listed in the table, their durations, and their predecessors. This chart is based on the activity network shown in Figure 12.9. A more completely represented activity network is given in Figure 12.10, listing the ES, LS, EF and LF for each activity. It is

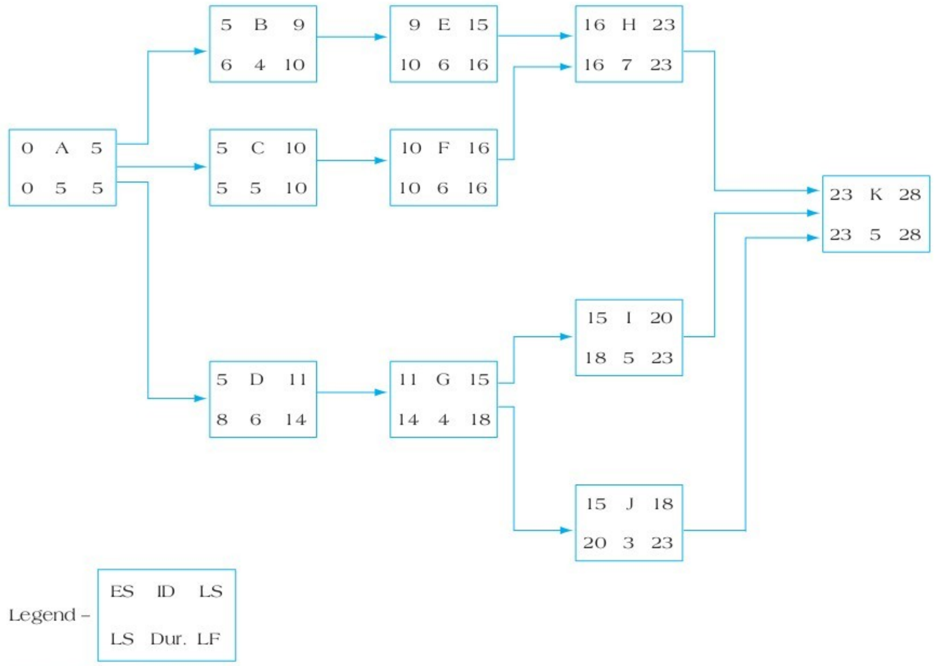


FIGURE 12.10 **Sample Project Network with Early and Late Start Indicated**

now possible to create a resource-loading table by combining the information we have in Figures 12.8 and 12.10 with one additional factor: the resources required to complete each project activity.

Naturally, there is a direct relationship between the resources we can apply to a task and its expected time to completion. For example, suppose that a task requiring one person working 40 hours per week is estimated to take two weeks (or 80 hours) to complete. Generally, we can modify the duration estimate, given adjustments to the projected resources available to work on the task. For example, if we can now assign two people to work full-time (40 hours) on the task, the new duration for the activity will be one week. Although the task will still require 80 hours of work to complete, with two full-time resources assigned, that 80 hours can actually be finished in one week of the project's scheduled baseline.

Table 12.4 identifies the activities, their durations, total activity float (or slack), and most importantly, the number of hours per week that we can assign resources to the tasks. The time value is less than full-time to illustrate a typical problem: Because of other commitments, project team

TABLE 12.4 Activity Float and Resource Needs for the Sample Network

Activity	Duration	Total Float	Resource Hours Needed per Week	Total Resource Hours Required
A	5	0	6	30
B	4	1	2	8
C	5	0	4	20
D	6	3	3	18
E	6	1	3	18
F	6	0	2	12
G	4	3	4	16
H	7	0	3	21
I	5	3	4	20
J	3	5	2	6
K	5	0	5	25
			Total	194

members may be assigned to the project on a basis that is less than full-time. So, for example, activity A is projected to take five days, given resources assigned to it at six hours per day (or a total estimated task duration of 30 hours). Activity F is projected to take six days to complete with two hours per day assigned to it. The total resources required to complete this project within the projected time frame are 194 hours. Once this information is inserted into the project, it is now possible to follow a series of steps aimed at resource-leveling the activity network. These steps will be considered in turn.

Step One: Develop the Resource-Loading Table

The **resource-loading table** is created through identifying the project activities and their resources required to completion and applying this information to the project schedule baseline. In its simplest form, the resource-loading table can be profiled to resemble a histogram, identifying hours of resource requirements across the project's life (see Figure 12.11). However, a more comprehensive resource-loading table is developed in Figure 12.12. It assumes the project begins on January 1 and the activities follow in the order identified through the project Gantt chart. Note that the resources required per day for each activity are listed against the days of the project baseline schedule when they will be needed. These total resource hours are then summed along the bottom of the table to identify the overall resource profile for the project. Note further that resource requirements tend to move up and down across the baseline, peaking at a total of 10 hours of resources required on day 10 (January 12).

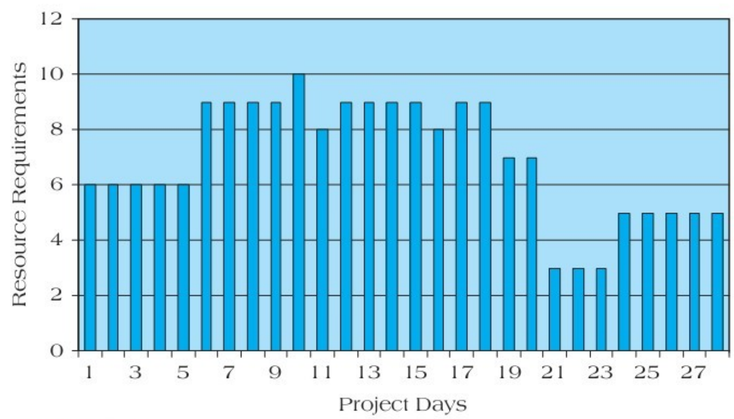


FIGURE 12.11 Resource Profile for Sample Project Network

													January													February		
Activity	1	2	3	4	5	8	9	10	11	12	15	16	17	18	19	22	23	24	25	26	29	30	31	1	2	5	6	7
A	6	6	6	6	6																							
B						2	2	2	2																			
C						4	4	4	4	4																		
D						3	3	3	3	3	3																	
E										3	3	3	3	3	3													
F											2	2	2	2	2	2												
G												4	4	4	4													
H																	3	3	3	3	3	3	3					
I																4	4	4	4	4								
J																2	2	2										
K																								5	5	5	5	5
Total	6	6	6	6	6	9	9	9	9	10	8	9	9	9	9	8	9	9	7	7	3	3	3	5	5	5	5	5

FIGURE 12.12 Resource-Loading Table for Sample Network

	January																							February				
Activity	1	2	3	4	5	8	9	10	11	12	15	16	17	18	19	22	23	24	25	26	29	30	31	1	2	5	6	7
A	6	6	6	6	6⌋																							
B						2	2	2	2	⌋																		
C						4	4	4	4	4⌋																		
D						3	3	3	3	3	3			⌋														
E										3	3	3	3	3	3	⌋												
F											2	2	2	2	2	2⌋												
G												4	4	4	4			⌋										
H																	3	3	3	3	3	3	3⌋					
I																4	4	4	4	4			⌋					
J																2	2	2					⌋					
K																								5	5	5	5	5⌋
Total	6	6	6	6	6	9	9	9	9	10	8	9	9	9	9	8	9	9	7	7	3	3	3	5	5	5	5	5

(⌋ Late Finish)

FIGURE 12.13 **Resource-Loading Table for Sample Network When Activity Float is Included**

The advantage of developing a detailed resource profile is that it provides a useful visual demonstration of the projected resource requirements needed across the entire project baseline. It is possible to use this resource profile in conjunction with the resource-loading table to develop a strategy for optimal resource leveling.

Step Two: Determine Activity Late Finish Dates

The next step in the resource-leveling process consists of applying the additional information regarding activity slack and late finish dates to the resource-loading table (see Table 12.3). This modified table is shown in Figure 12.13. Note that in this figure, we can identify the activities with slack time and those that are critical (no slack time). The late finish dates for those activities with slack are included and are represented as brackets. Hence, activities B, D, E, G, I, and J are shown with late finish dates corresponding to the slack time associated with each task, while the late finish for the activities along the critical path (A – C – F – H – K) is identical to the activities' early finish dates.

Step Three: Identify Resource Overallocation

After the resource-loading table is completed and all activity late finish dates are embedded, the process of actual resource leveling can begin with an examination of the resource profile for the project. What we are looking for here are any points across the project baseline at which resources have been allocated beyond the maximum resource level available. For example, in Figure 12.13, note that the total resources needed (the summation along the bottom row) reveals the maximum needed for any day of the project falls on January 12, when tasks requiring 10 resource units are scheduled. The question project managers need to consider is whether this resource profile is acceptable or if it indicates trouble, due to an allocation of resources that will not be available. If, for example, the project is budgeted for up to 10 resource units per day, then this resource profile is acceptable. On the other hand, if resources are limited to some figure below the maximum found in the project's resource profile, the project has an overallocation problem that must be addressed and corrected.

Certainly, at this point, the best-case scenario is to discover that resources have been allocated at or below the maximum across the project baseline. Given the nature of both time and resource project constraints, however, it is much more common to find situations of resource conflicts that require leveling. Suppose that in our sample project the maximum number of resource units available on any day is nine. We have already determined that on January 12, the project is scheduled to require 10 units, representing an overallocation. The discovery of overallocations triggers the next step in the resource-leveling process, in which we correct the schedule to eliminate resource conflict.

Step Four: Level the Resource-Loading Table

Once a determination has been made that the project baseline includes overallocated resources, an iterative process begins in which the resource-loading table is reconfigured to eliminate the resource contention points. The most important point to remember in resource leveling is that a *ripple effect* commonly occurs when we begin to rework the resource schedule to eliminate the sources of resource conflict. This ripple effect will become evident as we work through the steps necessary to level the sample project.

PHASE ONE Using Figure 12.13, examine the conflict point, January 12, for the tasks that require 10 resource units. Tasks C, D, and E are all scheduled on this day and have resource unit commitments of 4, 3, and 3 hours respectively. Therefore, the first phase in resource leveling consists of identifying the relevant activities to determine which are likely candidates for modification. Next, which activity should be adjusted? Using the priority heuristic mentioned previously, first examine the activities to see which are critical and which have some slack time associated with them. From developing the network, we know that activity C is on the critical path. Therefore, avoid reconfiguring this task if possible because any adjustment of its duration will adversely affect the overall project schedule. Eliminating activity C leaves us the choice of adjusting either activity D or activity E.

PHASE TWO Select the activity to be reconfigured. Both activities D and E have slack time associated with them. Activity D has three days of slack and activity E has one day. According to the rule of thumb, we might decide to leave activity E alone because it has the least amount of slack time. In this example, however, this option would lead to "splitting" activity D; that is, we would begin activity D on January 8, stop on the 12th, and then finish the last two days of work on January 15 and 16. Simply representing this option, we see in Figure 12.14, which shows the Gantt chart for our project, that the splitting process complicates our scheduling process to some degree. Note further that the splitting does not lengthen the overall project baseline, however; with the three days of slack associated with this task, lagging the activity one day through splitting it does not adversely affect the final delivery date.

For simplicity's sake, then, we will avoid the decision to split activity D for the time being, choosing the alternative option of adjusting the schedule for activity E. This option is also viable in that it does not violate the schedule baseline (there is slack time associated with this activity).

Figure 12.15 shows the first adjustment to the original resource-loading table. The three resource units assigned to activity E on January 12 are scratched and added back in at the end of the activity, thereby using up the one day of project slack for that activity. The readjusted resource-loading table now shows that January 12 no longer has a resource conflict, because the baseline date shows a total of seven resource units.

PHASE THREE After making adjustments to smooth out resource conflicts, reexamine the remainder of the resource table for *new* resource conflicts. Remember that adjusting the table can cause ripple effects in that these adjustments may disrupt the table in other places. This exact effect has occurred in this example. Note that under the adjusted table (see Figure 12.15), January 12 no longer shows a resource conflict; however, the act of lagging activity E by one day would create a conflict on January 22, in which 11 resource units would be scheduled. As a result, it is necessary to go through the second-phase process once more to eliminate the latest resource conflict.

Here again, the candidates for adjustment are all project tasks that are active on January 22, including activities E, F, I, and J. Clearly, activities E and F should, if possible, be eliminated as first choices given their lack of any slack time (i.e., they both now reside on a critical path).

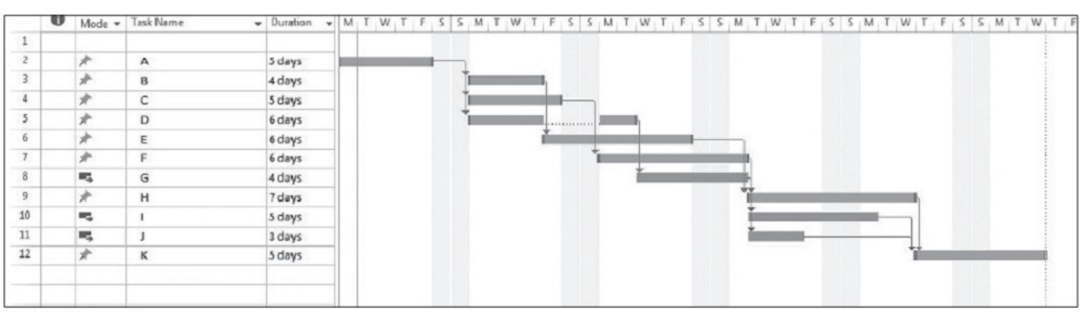


FIGURE 12.14 **Reconfiguring the Schedule by Splitting Activity D**

Source: MS Project 2013, Microsoft Corporation

	January																									February		
Activity	1	2	3	4	5	8	9	10	11	12	15	16	17	18	19	22	23	24	25	26	29	30	31	1	2	5	6	7
A	6	6	6	6	6⌋																							
B						2	2	2	2	⌋																		
C						4	4	4	4	4⌋																		
D						3	3	3	3	3	3			⌋														
E										~~3~~	3	3	3	3	3	3⌋												
F											2	2	2	2	2	2⌋												
G												4	4	4	4			⌋										
H																	3	3	3	3	3	3	3⌋					
I																4	4	4	4	4			⌋					
J																~~2~~	2	2	2				⌋					
K																								5	5	5	5	5⌋
Total	6	6	6	6	6	9	9	9	9	7	8	9	9	9	9	9	9	9	9	7	3	3	3	5	5	5	5	5

(⌋ Late Finish)

FIGURE 12.15 **Resource-Leveling the Network Table**

Adjusting (lagging) one day for either of the alternatives, activities I and J, will reduce the resource requirement to a level below the threshold, suggesting that either of these activities may be used. The earlier heuristic suggested that priority be given to activities with less slack time, so in this example we will leave activity I alone and instead lag the start of activity J by one day. Note that the resource totals summed across the bottom of the table (see Figure 12.15) now show that all activities are set at or below the cutoff level of nine resource hours per day for the project, completing our task. Further, in this example, we were able to resource-level the project without adding additional dates to the project schedule or requiring additional resources; in effect, resource leveling in this example violated neither a resource-constrained nor a time-constrained restriction.

Suppose, however, that our project operated under more stringent resource constraints; for example, instead of a threshold of nine hours per day, what would be the practical effect of resource-leveling the project to conform to a limit of eight hours per day? The challenge to a project manager now is to reconfigure the resource-loading table in such a way that the basic tenet of resource constraint is not violated. In order to demonstrate the complexity of this process, for this example, we will break the decision process down into a series of discrete steps as we load each individual activity into the project baseline schedule (see Table 12.5). Note the

TABLE 12.5 Steps in Resource Leveling

Step	Action
1	Assign Activity A to the resource table.
2	In selecting among Activities B, C, and D, employ the selection heuristic and prioritize C (critical activity) and then B (smallest amount of slack). Load C and B into the resource table. Delay Activity D.
3	On January 12, load Activity D. D had 3 days slack and is loaded four days late. Total delay for Activity D is 1 day.
4	On January 15, load Activities E and F (following completion of B and C). Prioritize F first (critical activity), and then add E. Both activities finish on January 22, so overall critical path schedule is not affected. Total project delay to date = 0.
5	Because of resource constraints, Activity G cannot begin until January 23. G had 3 days slack and is loaded five days late, finishing on January 26. Total delay for Activity G is 2 days.
6	Load Activity H on January 23, following completion of Activities E and F. H is completed on January 31, so overall critical path schedule is not affected. Total project delay to date = 0.
7	Because of resource constraints, Activity I cannot begin until January 29. I is loaded five days late. Total delay for Activity I is 2 days (new finish date = February 2).
8	Because of resource constraints, Activity J cannot begin until February 1. Even with slack time, J is delayed 3 days, completing on February 5.
9	Activity K cannot be loaded until completion of predecessors H, I, and J. K begins on February 6 and completes on February 12. Total project delay = 3 days.

January

Total Slack	Activity	1	2	3	4	5	8	9	10	11	12	15	16	17	18	19	22	23	24	25	26
0	A	⌠6	6	6	6	6															
1	B						⌠2	2	2	2	2										
0	C						⌠4	4	4	4	4										
1	D						⌠				3	3	3	3	3	3					
0	E										⌠	3	3	3	3	3	3				
0	F											⌠2	2	2	2	2	2				
2	G												⌠					4	4	4	4
	H																	⌠3	3	3	3
	I																⌠				
	J																⌠				
	K																				
	Total	6	6	6	6	6	6	6	6	6	7	8	8	8	8	8	5	7	7	7	7

February

Total Slack	Activity	29	30	31	1	2	5	6	7	8	9	12	13	14	15	16	17
	A																
	B																
	C																
	D																
	E																
	F																
	G																
0	H	3	3	3													
−2	I	4	4	4	4	4											
−3	J				2	2	2										
−3	K				⌠			5	5	5	5	5					
	Total	7	7	7	6	6	2	5	5	5	5	5					

⌠ = Original activity early start time

FIGURE 12.16 **Resource-Loading Table with Lowered Resource Constraints**

need, at times, to make sacrifices to the initial baseline schedule in order to maintain the nonviolation of the resource-loading limit.

Figure 12.16 pictures this resource-leveling example given in Table 12.4 with January and February stacked. As the steps in the table indicate, the determination of total project delay is not evident until all predecessor tasks have been loaded, resources leveled at the point each new activity is added to the table, and the overall project baseline schedule examined. Interestingly, note from this example that the project's schedule did not show a delay through the inclusion of 8 of the 11 activities (through activity H). However, once activity H was included in the resource table, it was necessary to delay the start of activity J in order to account for the project resource constraint. As a result, the project's baseline schedule was delayed through a combination of loss of project slack and the need to reassess the activity network in light of resource constraints. The overall effect of this iterative process was to delay the completion of the project by three days.

The extended example in this section illustrates one of the more difficult challenges that project managers face: *the need to balance concern for resources with concern for schedule*. In conforming to the new, restricted resource budget, which allows us to spend only up to eight resource units per day, the alternatives often revolve around making reasoned schedule trade-offs to account for limited resources. The project's basic schedule dictates that any changes to the availability of sufficient resources to support the activity network are going to involve lengthening the project's duration. Part of the reason for this circumstance, of course, lies in the fact that this example included a simplified project schedule with very little slack built into any of the project activities. As a result, major alterations to the project's resource base were bound to adversely affect the overall schedule.

In summary, the basic steps necessary to produce a resource-leveled project schedule include the following:

1. Create a project activity network diagram (see Figure 12.10).
2. From this diagram, create a table showing the resources required for each activity, the activity durations, and the total float available (see Table 12.4).
3. Develop a time-phased resource-loading table that shows the resources required to complete each activity, the activity early starts, and the late finish times (Figure 12.13).
4. Identify any resource conflicts and begin to "smooth" the loading table using one or more of the heuristics for prioritizing resource assignment across activities (Figure 12.15).
5. Repeat step 4 as often as necessary to eliminate the source of resource conflicts. Use your judgment to interpret and improve the loading features of the table. Consider alternative means to minimize schedule slippage; for example, use overtime during peak periods.

12.4 RESOURCE-LOADING CHARTS

Another way to create a visual diagram of the resource management problem is to employ resource-loading charts. **Resource-loading charts** are used to display the amount of resources required as a function of time on a graph. Typically, each activity's resource requirements are represented as a block (resource requirement over time) in the context of the project baseline schedule. Resource-loading charts have the advantage of offering an immediate visual reference point as we attempt to lay out the resources needed to support our project as well as smooth resource requirements over the project's life.

Here is an example to illustrate how resource-loading charts operate. Suppose our resource profile indicated a number of "highs" and "lows" across the project; that is, although the resource limit is set at eight hourly resource units per day, on a number of days our actual resources employed are far less than the total available. The simplified project network is shown in Figure 12.17 and summarized in Table 12.6, and the corresponding resource-loading chart is shown in Figure 12.18. The network lists the early start and finish dates for each activity, as well as the resources required for each task for each day of work.

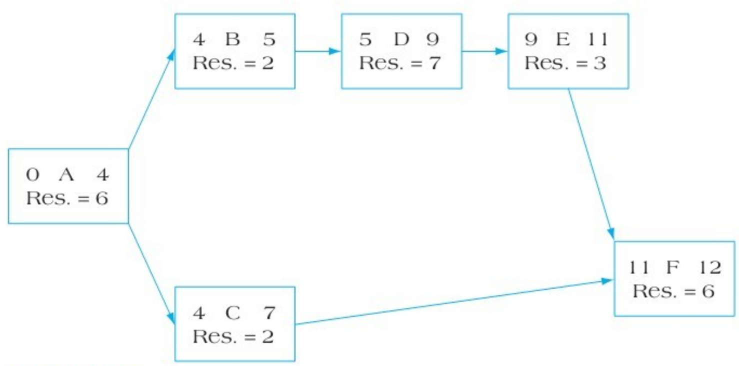


FIGURE 12.17 Sample Project Network

TABLE 12.6 Resource Staffing (Hourly Units) Required for Each Activity

Activity	Resource	Duration	Early Start	Slack	Late Finish
A	6	4	0	0	4
B	2	1	4	0	5
C	2	3	4	4	11
D	7	4	5	0	9
E	3	2	9	0	11
F	6	1	11	0	12

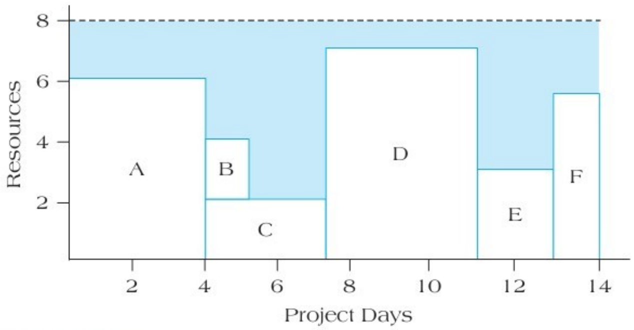


FIGURE 12.18 **Resource-Loading Chart for Sample Project**

In constructing a resource-loading chart that illustrates the time-limited nature of resource scheduling, there are six main steps to follow:[8]

1. Create the activity network diagram (see Figure 12.17).
2. Produce a table for each activity, the resource requirements, the duration, early start time, slack, and late finish time (see Table 12.6).
3. List the activities in order of increasing slack (or in order of latest finish time for activities with the same slack).
4. Draw an initial resource-loading chart with each activity scheduled at its earliest start time, building it up following the order shown in step 3. This process creates a loading chart with the most critical activities at the bottom and those with the greatest slack on the top.
5. Rearrange the activities within their slack to create a profile that is as level as possible within the guidelines of not changing the duration of activities or their dependence.
6. Use your judgment to interpret and improve activity leveling by moving activities with extra slack in order to "smooth" the resource chart across the project (see Figure 12.18).

Note that the early finish for the project, based on its critical path, is 12 days. However, when we factor in resource constraints, we find that it is impossible to complete all activities within their allocated time, causing the schedule to slip two days to a new early finish date of 14 days. Figure 12.18 illustrates the nature of our problem: Although the project allows for a total of eight hours per day for project activities, in reality, the manner in which the project network is set up relative to the resources needed to complete each task makes it impossible to use resources as efficiently as possible. In fact, during days 5 through 7, a total of only two resource hours is being used for each day.

A common procedure in resolving resource conflicts using resource-loading charts is to consider the possibility of splitting activities. As we noted earlier in the chapter, **splitting** an activity means interrupting the continuous stream of work on an activity at some midpoint in its development process and applying that resource to another activity for some period before returning the resource to complete the original task. Splitting can be a useful alternative technique for resource leveling provided there are no excessive costs associated with splitting the task. For example, large start-up or shutdown costs for some activities make splitting them an unattractive option.

To visually understand the task-splitting option, refer back to the Gantt chart created in Figure 12.14. Note that the decision there was made to split activity D in order to move the start of activity E forward. This decision was undertaken to make best use of constrained resources; in this case, there was sufficient slack in activity D to push off its completion and still not adversely affect the overall project schedule. In many circumstances, project teams seeking to make best use of available resources will willingly split tasks to improve schedule efficiency.

What would happen if we attempted to split activities, where possible, in order to make more efficient use of available resources? To find out, let us return to the activity network in Figure 12.17 and compare it with the resource-loading chart in Figure 12.18. Note that activity C takes three days to complete. Although activity C is not a predecessor for activity D, we cannot start D until C is completed, due to lack of available resources (day 5 would require nine resource hours when only eight are available). However, suppose we were to split activity C so that the task is started on day 4 and the balance is left until activity D is completed. We can shift part of this activity because it contains four days of slack. Figure 12.19 illustrates this alternative. Note that two days of

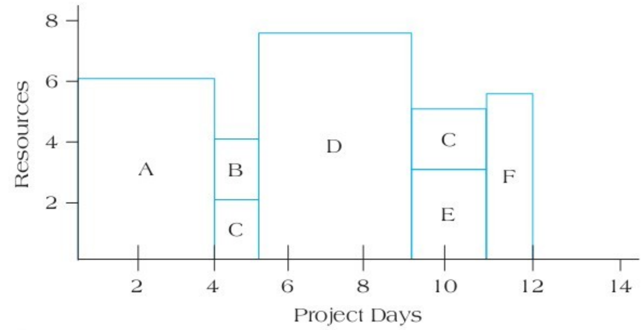


FIGURE 12.19 **Modified Resource-Loading Chart When Splitting Task C**

activity C are held until after D is completed, when they are performed at the same time as activity E. Because the final task, F, requires that both C and E be completed, we do not delay the start of activity F by splitting C. In fact, as Figure 12.19 demonstrates, splitting C actually makes more efficient use of available resources and, as a result, moves the completion date for the project two days earlier, from day 14 back to the originally scheduled day 12. This example illustrates the benefit that can sometimes be derived from using creative methods for better utilization of resources. In this case, splitting activity C, given its four days of slack time, enables the project to better employ its resources and regain the original critical path completion date.

BOX 12.1

Project Managers in Practice

Captain Kevin O'Donnell, U.S. Marine Corps

As a Marine officer, Captain Kevin O'Donnell has been working as a "project manager" for a number of years. As O'Donnell freely admits, at first glance, his duties do not seem to align with the traditional roles of project managers, and yet, the more we consider them, the more we can see that although the circumstances are unique, the principles and practices of project management remain applicable.

O'Donnell received a bachelor's degree in Criminal Justice from The Citadel, The Military College of South Carolina, and was commissioned as a second lieutenant in the U.S. Marine Corps. He is currently posted as a project officer and company executive officer at Marine Barracks, Washington, DC, and also has been posted to the presidential retreat, Camp David, as the guard officer and company executive officer.

As a second and first Lieutenant, O'Donnell served in the Second Battalion, 6th Marine Regiment, as a platoon commander and company executive officer while completing two combat deployments to Fallujah, Iraq. Although his duties have been far-ranging, including leading a number of missions and duty assignments, in O'Donnell's words, he prefers to focus on the way in which he has used project management in his career. Concepts such as a strategic vision, stakeholder management, scope of work, Work Breakdown Structure, tasks, time lines, and risk assessments are common to all projects, and the Marines use them daily during the planning of training, and while deployed and conducting combat operations.

As a platoon commander, O'Donnell was responsible for leading a force of 45 Marines during his first deployment to Iraq. They were tasked with conducting a variety of missions, and routinely engaged in vehicle and foot patrols, convoys, random house searches, and targeted raids on enemy personnel. O'Donnell notes:

> Take, for example, an intelligence-driven targeted raid on a known insurgent. This situation, viewed as a "project," required me as the platoon commander to analyze the area of operations and available intelligence, generate a five-paragraph order that contained a mission statement, tasking statements, scheme of maneuver for the operation, and logistic considerations (very similar to a project vision, scope of work, and Work Breakdown Structure). Additionally, I would need to coordinate with all adjacent and subordinate units that would be affected by the mission, and brief senior members

as a whole. We were forced to operate in a continually changing external environment, and our ability to effortlessly adapt and adjust our plan accordingly paid dividends to the mission's success. Throughout the execution of this mission, stakeholder requirements changed, mission parameters were adjusted, internal and external environment dynamics shifted, and personnel and team compositions were adjusted. However, at the end of it, through solid leadership at all levels of the chain of command, and fundamental execution of project management skills and principles, the mission was completed and dubbed a huge success. The city of Fallujah is now a self-secured and governed area of Iraq, and my battalion's actions there were utilized as the role model for pacifying and defeating the insurgency in other cities throughout Iraq.

Although O'Donnell's duties may not seem to be those of traditional project management, he is quick to point out that, in fact, the opposite is true. Carefully planned operations, defined objectives, clear strategies, and coordination and scheduling are all hallmarks of project management, and they form the critical processes for O'Donnell's duties commanding Marines in a hostile environment. "At the end of the day, regardless of industry, project management remains the same," O'Donnell concludes. "Understanding the difference between leadership and management is critical. Knowing your internal and external environments, along with how to plan, task and manage personnel, maintain a budget and time lines, have a clear understanding of your objectives, how you must meet customer and stakeholder requirements, and achieving desired results, are critical to any project manager's success."

12.5 MANAGING RESOURCES IN MULTIPROJECT ENVIRONMENTS

Most managers of projects eventually will be confronted with the problem of dealing with resource allocation across multiple projects. The main challenge is one of interdependency: Any resource allocation decisions made in one project are likely to have ramifications in other projects. What are some of the more common problems we find when faced with this sort of interdependency among projects? Some of the better known problems include inefficient use of resources, resource bottlenecks, ripple effects, and the heightened pressure on personnel to multitask.[9]

Any system used to resolve the complex problems with multiproject resource allocation has to consider the need, as much as possible, to minimize the negative effects of three key parameters: (1) schedule slippage, (2) resource utilization, and (3) in-process inventory.[10] Each of these parameters forms an important challenge across multiple projects.

Schedule Slippage

For many projects, schedule slippage can be more than simply the realization that the project will be late; in many industries, it can also result in serious financial penalties. It is not uncommon for companies to be charged thousands of dollars in penalty clauses for each day a project is delayed past the contracted delivery date. As a result, one important issue to consider when making decisions about resource allocation across multiple projects is their priority based on the impact of schedule slippage for each individual project.

Resource Utilization

The goal of all firms is to use their existing pool of resources as efficiently as possible. Adding resources companywide can be expensive and may not be necessary, depending upon the manner in which the present resources are employed. To illustrate this point, let us reconsider the example of a resource-loading chart, shown in Figure 12.21, applied to a firm's portfolio of projects rather than to just one project's activities. In this load chart, top management can assign up to eight resource units for each week of their project portfolio. Even using a splitting methodology to better employ these resources, there are still some clear points at which the portfolio is underutilizing available resources. For example, in week 5, only four resource units have been employed. The shaded area in the load chart (Figure 12.21) shows the additional available resources not employed in the current project. To maximize the resource utilization parameter, we would attempt to assign the available resources on other, concurrent projects, thereby improving the overall efficiency with which we use project resources.

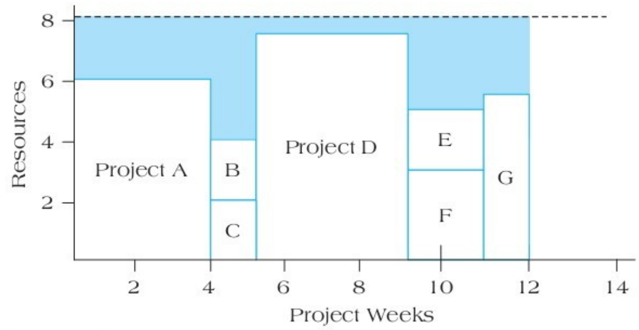


FIGURE 12.21 **Resource-Loading Chart Across Multiple Projects**

In-Process Inventory

The third standard for analyzing the optimal use of multiproject resources is to consider their impact on in-process inventory. **In-process inventory** represents the amount of work waiting to be completed but delayed due to unavailable resources. For example, an architectural firm may find several projects delayed because it only employs one checker responsible for final detailing of all blueprints. The projects stack up behind this resource bottleneck and represent the firm's in-process inventory of projects. Excessive in-process inventory is often caused by a lack of available resources and represents the kinds of trade-off decisions companies have to make in multiproject environments. Should we hire additional resources in order to reduce our in-process inventory? If this problem is only temporary, will hiring additional resources lead to inefficient resource allocation later on?

In effect, project organizations often have to strike an appropriate balance across the three parameters: schedule slippage, resource allocation, and in-process inventory. The steps necessary to improve one measure may have negative effects on one or more of the other standards. For example, steps to maximize resource allocation may provoke schedule slippage or increase in-process inventory. Any strategies we use to find a reasonable balance among these parameters must recognize the need to juggle multiple competing demands.

Resolving Resource Decisions in Multiproject Environments

The challenge of scheduling resources in multiproject environments has to do with the need to work on two levels to achieve maximum efficiency. First, with multiple projects, we have to make considered decisions regarding which projects should be assigned highest priority to resources. However, it is also vital to recognize that we are often required to schedule the activities of multiple projects simultaneously. Consider the resource-loading chart in Figure 12.21. On one level, we can see that this chart has scheduled projects A through G across 12 weeks. Project A will take the majority of resources for the first four weeks. However, during the fourth week, we have scheduled two projects at the same time (B and C). We must now work to balance their individual activity resource requirements so that both projects can be completed during the same time period. This figure illustrates the nature of the problem: On a larger level, resource allocation across multiple projects requires us to schedule projects in order to most efficiently use our resources. However, on another level, when projects compete for resources at the same time, we need to work to ensure that we can prioritize our resources across them to maximize their availability.

There are a number of potential methods for resolving resource allocation challenges in a multiproject setting, ranging from highly simplified heuristics to more complex mathematical programming options. The goal of each technique is to make the most efficient use of resources across multiple projects, often with competing requirements and priorities.

FIRST IN LINE The simplest rule of thumb for allocating resources is to prioritize on the basis of which projects entered the queue first. This "first come, first served" approach is easy to employ, because it simply follows the master project calendar. When resource allocation decisions need to be made, they can be done quickly by comparing the starting dates of the projects in question. Unfortunately, this technique ignores any other important information that may suggest the need to reorder the resource allocation process, such as strategic priorities, emergency or crisis situations, or projects with higher potential for commercial success. The first-in-line option can cause companies to underallocate resources to potentially high-payoff projects purely on the basis of when they were authorized, relative to earlier and less useful projects.

GREATEST RESOURCE DEMAND This decision rule starts by determining which projects in the company's portfolio will pose the greatest demand on available resources. Those projects that require the most resources are first identified and their resources are set aside. Once they have been prioritized and resources allocated, the company reexamines the remaining pool of projects and selects those with the next highest resource demands until the available pool is exhausted. The logic of the greatest-resource-demand approach is to recognize that resource bottlenecks are likely to spring from unexpected peaks in resource needs relative to the number of projects under development. Consequently, this approach identifies these possible bottlenecks and uses them as the starting point for additional resource allocation.

GREATEST RESOURCE UTILIZATION A variation of the greatest-resource-demand heuristic is to allocate resources in order to ensure the greatest use of project resources, or in order to minimize resource idle time. For example, an organization may seek to prioritize three projects, A, B, and C, across a resource pool made up of programmers, system analysts, and networking staff. Although project A requires the most resources for completion, it does not require any work from the system analyst resource pool. On the other hand, project B does not require as many total resources for completion, but it does need to utilize members of all three resource pool groups, that is, programmers, system analysts, and network specialists. As a result, the company may elect to prioritize project B first in order to ensure that all resources are being utilized to the greatest possible degree.

MINIMUM LATE FINISH TIME This rule stipulates that resource priority should be assigned to activities and projects on the basis of activity finish dates. The earliest late finishers are scheduled first. Remember that "late finish" refers to the latest an activity can finish without compromising the overall project network by lengthening the critical path. The goal of this heuristic is to examine those project activities that have extra slack time as a result of later late finish dates and prioritize resources to the activities with minimal slack, that is, early late finish dates.

MATHEMATICAL PROGRAMMING Math programming can be used to generate optimal solutions to resource-constrained problems in the multiproject setting, just as it can be employed for single projects. The common objectives that such models seek to maximize are:[11]

1. Minimize total development time for all projects
2. Minimize resource utilization across all projects
3. Minimize total lateness across all projects

These goals are subject to the resource constraints that characterize the nature of the problem, including: (1) limited resources, (2) precedence relationships among the activities and projects, (3) project and activity due dates, (4) opportunities for activity splitting, (5) concurrent versus nonconcurrent activity performance requirements, and (6) substitution of resources to assign to activities. Although mathematical programming is a worthy approach to resolving the constrained resource problem in either a single or multiproject setting, its use tends to be limited depending on the complexity of the problem, the large number of computational variables, and the time necessary to generate a sufficiently small set of options.

Resource management in projects is a problem that is frequently overlooked by novice project managers or in firms that have not devoted enough time to understanding the full nature of the project management challenge they face. As noted, it is common to develop project plans and schedules under the assumption of unlimited resources, as if the organization can always find the trained personnel and other resources necessary to support project activities no matter how committed they currently are to project work. This practice inevitably leads to schedule slippages and extra costs as the reality of resource availability overshadows the optimism of initial scheduling. In fact, resource management represents a serious step in creating reasonable and accurate estimates for project activity durations by comparing resources needed to undertake an activity to those available at any point in time. Further resource management recognizes the nature of time/cost trade-offs that project managers are frequently forced to make. The extra resources necessary to accomplish tasks in a timely fashion do not come cheap, and their expense must be balanced against too aggressive project schedules that put a premium on time without paying attention to the budget impact they are likely to have.

Resource management is an iterative process that can be quite time-consuming. As we balance our activity network and overall schedule against available resources, the inevitable result will be the need to make adjustments to the network plan, rescheduling activities in such a way that they have minimal negative effect on the overall activity network and critical path. Resource leveling, or smoothing, is a procedure that seeks to make resource scheduling easier through minimizing the fluctuations in resource needs across the project by applying resources as uniformly as possible. Thus, resource management can make project schedules more accurate while allowing for optimal scheduling of project resources. Although this process can take time early in the project planning phase, it will pay huge dividends in the long run, as we create and manage project plans based on meaningful resource requirements and duration estimates rather than wishful thinking.

Summary

1. **Recognize the variety of constraints that can affect a project, making scheduling and planning difficult.** Effectively managing the resources for projects is a complex challenge. Managers must first recognize the wide variety of constraints that can adversely affect the efficient planning of scheduling of projects, including technical constraints, resource constraints, and physical constraints. Among the set of significant resource constraints are project personnel, materials, money, and equipment. A reasonable and thorough assessment of both the degree to which each of these resource types will be needed for the project and their availability is critical for supporting project schedules.
2. **Understand how to apply resource-loading techniques to project schedules to identify potential resource overallocation situations.** Resource loading is a process for assigning the resource requirements for each project activity across the baseline schedule. Effective resource loading ensures that the project team is capable of supporting the schedule by ensuring that all activities identified in the schedule have the necessary level of resources assigned to support their completion within the projected time estimates. We can profile the resource requirements for a project across its life cycle to proactively plan for the needed resources (both in terms of type of resource and amount required) at the point in the project when activities are scheduled to be accomplished. One effective, visual method for resource planning utilizes resource-leveling techniques to "block out" the activities, including required resource commitment levels, across the project's baseline schedule. Resource leveling offers a useful heuristic device for recognizing "peaks and valleys" in our resource commitment over time that can make resource scheduling problematic.
3. **Apply resource-leveling procedures to project activities over the baseline schedule using appropriate prioritization heuristics.** We employ "resource-smoothing" techniques in an effort to minimize the problems associated with excessive fluctuations in the resource-loading diagram. Resource smoothing minimizes these fluctuations by rescheduling activities in order to make it easier to apply resources continuously over time. The first step in resource leveling consists of identifying the relevant activities to determine which are likely candidates for modification. The next question to resolve is: Which activity should be adjusted? Using the priority heuristic mentioned previously, we would first examine the activities to see which are critical and which have some slack time associated with them. The second step is to select the activity to be reconfigured. According to the rule of thumb, we first select the activities with the most slack time for reconfiguration.
4. **Follow the steps necessary to effectively smooth resource requirements across the project life cycle.** In constructing a resource-loading chart that illustrates the time-limited nature of resource scheduling, there are six main steps to follow: (1) Create the activity network diagram for the project; (2) produce a table for each activity that includes the resources required, the duration, and the early start time, slack, and late finish time; (3) list the activities in order of increasing slack (or in order of latest finish time for activities with the same slack); (4) draw an initial resource-loading chart with each activity scheduled at its earliest start time, building it up following the order shown in step 3, to create a loading chart with the most critical activities at the bottom and those with the greatest slack on the top; (5) rearrange the activities within their slack to create a profile that is as level as possible within the guidelines of not changing the duration of activities or their dependence; and (6) use judgment to interpret and improve activity leveling by moving

activities with extra slack in order to "smooth" the resource chart across the project.

5. **Apply resource management within a multiproject environment.** Resource management is a far more complex activity when we consider it within a multiproject environment, that is, when we try to schedule resources among multiple projects that are all competing for a limited supply of resources. In such circumstances, a number of options are available to project managers to find the optimal balance between multiple competing projects and finite resources. Among the decision heuristics we can employ in making the resource allocation decisions are those which choose on the basis of (1) which projects are first in line, (2) which projects have the greatest resource demand, (3) which projects will enable our firm to use the greatest resource utilization, (4) which will enable us to reach the goal of minimizing late finish times, and (5) through the use of mathematical programming.

Key Terms

In-process inventory *(p. 421)*
Leveling heuristics *(p. 407)*
Mixed-constraint project *(p. 403)*
Physical constraints *(p. 403)*
Resource-constrained project *(p. 403)*
Resource constraints *(p. 402)*
Resource leveling *(p. 407)*
Resource loading *(p. 405)*
Resource-loading charts *(p. 416)*
Resource-loading table *(p. 411)*
Resource usage table *(p. 405)*
Smoothing *(p. 407)*
Splitting activities *(p. 417)*
Time-constrained project *(p. 403)*

Solved Problem

Consider the resource-loading table shown here. Assume that we cannot schedule more than eight hours of work during any day of the month.

a. Can you identify any days that involve resource conflicts?
b. How would you reconfigure the loading table to resolve these resource conflicts?

June

Activity	1	2	3	4	5	8	9	10	11	12	15	16	17	18	19	22	23	24	25	26
A	4	4	4	4	4⌋															
B						4	4	4		⌋										
C						4	4	4	4	4⌋										
D						3	3	3	3	3	3			⌋						
E											3	3	3	3	3	⌋				
F											2	2	2	2	2	2⌋				
G												4	4	4	4				⌋	

SOLUTION

a. According to the resource-loading table, the dates June 8, 9, and 10 are all overallocated (11 hours), as are June 16, 17, 18, and 19 (9 hours).
b. One solution for leveling the resource-loading table is by taking advantage of slack time available in activities D and G and moving these activities later in the schedule to correspond with their late finish dates (see the resource-loading table shown below).

June

Activity	1	2	3	4	5	8	9	10	11	12	15	16	17	18	19	22	23	24	25	26
A	4	4	4	4	4⌋															
B						4	4	4		⌋										
C						4	4	4	4	4⌋										
D						~~3~~	~~3~~	~~3~~	3	3	3	3	3	3	⌋					
E											3	3	3	3	3			⌋		
F											2	2	2	2	2	2⌋				
G												~~4~~	~~4~~	~~4~~	~~4~~	4	4	4	4	⌋
Total	4	4	4	4	4	8	8	8	7	7	8	8	8	8	5	6	4	4	4	

