CHAPTER EIGHT

Scheduling Resources

Project network times are not a schedule until resources have been assigned.

There are always more project proposals than there are available resources. The priority system needs to select projects that best contribute to the organization's objectives, within the constraints of the resources available. If all projects and their respective resources are computer scheduled, the feasibility and impact of adding a new project to those in process can be quickly assessed. With this information the project priority team will add a new project only if resources are available to be formally committed to that specific project. This chapter examines methods of scheduling resources so the team can make realistic judgments of resource availability and project durations. The project manager uses the same schedule for implementing the project. If changes occur during project implementation, the computer schedule is easily updated and the effects easily assessed.

The Problem

After staff and other resources were assigned to her project, a project manager listed the following questions that still needed to be addressed:

- Will the assigned labor and/or equipment be adequate and available to deal with my project?
- Will outside contractors have to be used?
- Do unforeseen resource dependencies exist? Is there a new critical path?
- How much flexibility do we have in using resources?
- Is the original deadline realistic?

Clearly, this project director has a good understanding of the problems she is facing. Any project scheduling system should facilitate finding quick, easy answers to these questions.

The planned network and activity project duration times found in previous chapters fail to deal with resource usage and availability. The time estimates for the work packages and network times were made independently with the implicit assumption that resources would be available. This may or may not be the case. If resources are adequate but the demand varies widely over the life of the project, it may be desirable to even out resource demand by delaying noncritical activities (using slack) to lower peak demand and, thus, increase resource utilization. This process is called *resource leveling* or *smoothing.* On the other hand, if resources are not adequate to meet peak demands, the late start of some activities must be delayed, and the duration of the project may be increased. This process is called *resource-constrained scheduling.* One research study by your author of more than 50 projects found that planned project network durations were increased 38 percent when resources were scheduled.

Because the costs of failing to consider resource usage and availability are hidden or not obvious, resource scheduling in practice is often not done or does not get the attention it deserves. The consequences of failing to schedule limited resources are a costly activity and project delays usually manifest themselves midway in the project when quick corrective action is difficult. An additional consequence of failing to schedule resources is ignoring the peaks and valleys of resource usage over the duration of the project. Because project resources are usually overcommitted and because resources seldom line up by availability and need, procedures are needed to deal with these problems. This chapter addresses methods available to project managers for dealing with resource utilization and availability through resource leveling and resource-constrained scheduling.

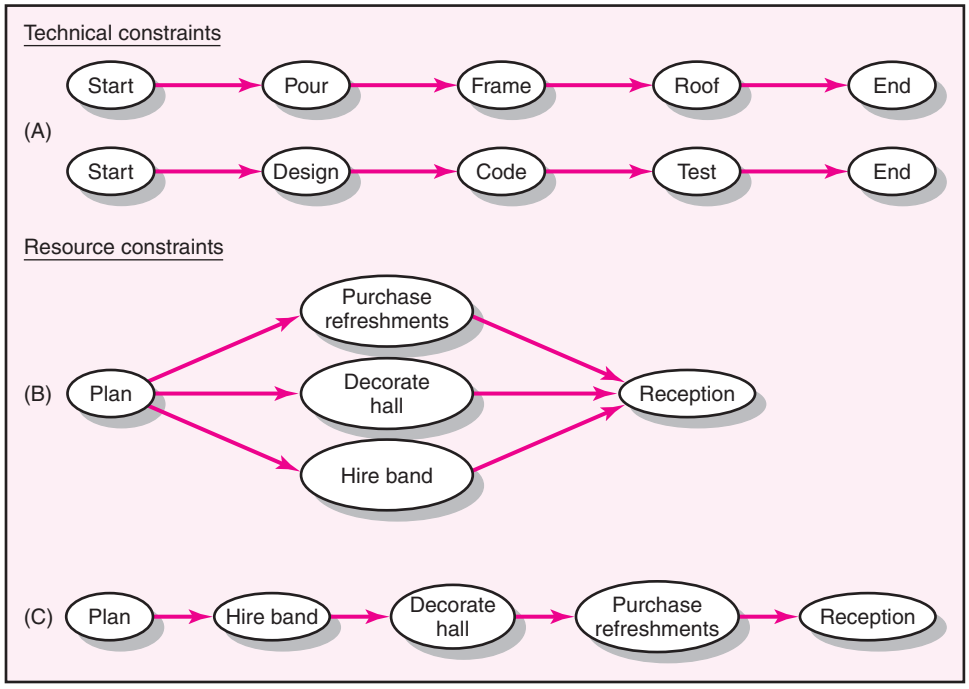
Types of Project Constraints

Project constraints impede or delay the start of activities. The result is a reduction in slack shown on the planned network, a decrease in scheduling flexibility, a possible decrease in the number of parallel activities, and an increase in the likelihood of delaying the project. Three project constraints need to be considered in scheduling.

Technical or Logic Constraints

These constraints usually address the *sequence* in which project activities must occur. The project network depicts technical constraints. A project network for framing a house might show three activities of (1) pour foundation, (2) build frame, and (3) cover roof in a sequence. A network for a new software project could place the activities of (1) design, (2) code, and (3) test in the network as a sequence. In other words, you cannot logically perform activity 2 until 1 is completed, etc. (see Figure 8.1A).

FIGURE 8.1
Constraint Examples



Physical Constraints

In rare situations there are physical constraints that cause activities that would normally occur in parallel to be constrained by contractual or environmental conditions. For example, renovation of a ship compartment might allow only one person to perform an activity because of space limitations. Another example would be erection of a tower and nearby ground work. The procedures for handling physical constraints are similar to those used for resource constraints.

Resource Constraints

The absence or shortage of resources can drastically alter technical constraints. A project network planner may assume adequate resources and show activities occurring in parallel. However, parallel activities hold potential for resource conflicts. For example, assume you are planning a wedding reception that includes four activities—(1) plan, (2) hire band, (3) decorate hall, and (4) purchase refreshments. Each activity takes one day. Activities 2, 3, and 4 could be done in parallel by different people. There is no technical reason or dependency of one on another (see Figure 8.1B). However, if one person must perform all activities, the resource constraint requires the activities be performed in sequence or series. Clearly the consequence is a delay of these activities and a very different set of network relationships (see Figure 8.1C). Note that the resource dependency takes priority over the technological dependency but *does not violate the technological dependency;* that is, hire, decorate, and purchase may now have to take place in sequence rather than concurrently, but they must all be completed before the reception can take place.

The interrelationships and interactions among time and resource constraints are complex for even small project networks. Some effort to examine these interactions before the project begins frequently uncovers surprising problems. Project managers who do not consider resource availability in moderately complex projects usually learn of the problem when it is too late to correct. A deficit of resources can significantly alter project dependency relationships, completion dates, and project costs. Project managers must be careful to schedule resources to ensure availability in the right quantities and at the right time. Fortunately, there are computer software programs that can identify resource problems during the early project planning phase when corrective changes can be considered. These programs only require activity resource needs and availability information to schedule resources.

Kinds of Resource Constraints

Resources are people, equipment, and material that can be drawn on to accomplish something. In projects the availability or unavailability of resources will often influence the way projects are managed. The major resources projects managers must marshal, schedule, and manage on a daily basis are people, materials, equipment, and working capital.

1. People This is the most obvious and important project resource. Human resources are usually classified by the skills they bring to the project—for example, programmer, mechanical engineer, welder, inspector, marketing director, supervisor. In rare cases some skills are interchangeable, but usually with a loss of productivity. The many differing skills of human resources add to the complexity of scheduling projects.

2. Materials Project materials cover a large spectrum: for example, chemicals for a scientific project, concrete for a road project, survey data for a marketing project.

Material availability and shortages have been blamed for the delay of many projects. When it is known that a lack of availability of materials is important and probable, materials should be included in the project network plan and schedule. For example, delivery and placement of an oil rig tower in a Siberian oil field has a very small time window during one summer month. Any delivery delay means a one-year, costly delay. Another example in which material was the major resource scheduled was the resurfacing and replacement of some structures on the Golden Gate Bridge in San Francisco. Work on the project was limited to the hours between midnight and 5:00 A.M. with a penalty of $1,000 per minute for any work taking place after 5:00 A.M. Scheduling the arrival of replacement structures was an extremely important part of managing the five-hour work-time window of the project. Scheduling materials has also become important in developing products where time-to-market can result in loss of market share.

3. Equipment Equipment is usually presented by type, size, and quantity. In some cases equipment can be interchanged to improve schedules, but this is not typical. Equipment is often overlooked as a constraint. The most common oversight is to assume the resource pool is more than adequate for the project. For example, if a project needs one earth-moving tractor six months from now and the organization owns four, it is common to assume the resource will not delay the pending project. However, when the earth-moving tractor is due on-site in six months, all four machines in the pool might be occupied on other projects. In multiproject environments it is prudent to use a common resource pool for all projects. This approach forces a check of resource availability across all projects and reserves the equipment for specific project needs in the future. Recognition of equipment constraints before the project begins can avoid high crashing or delay costs.

4. Working Capital In a few project situations such as construction, working capital is treated as a resource because it is limited in supply. If working capital is readily available, a project manager may work on many activities concurrently. If working capital is in short supply because progress payments are made monthly, materials and labor usage may have to be restricted to conserve cash. This situation represents a cash flow problem.

Classification of a Scheduling Problem

Most of the scheduling methods available today require the project manager to classify the project as either *time constrained* or *resource constrained.* Project managers need to consult their priority matrix (see Figure 4.2) to determine which case fits their project. One simple test to determine if the project is time or resource constrained is to ask, "If the critical path is delayed, will resources be added to get back on schedule?" If the answer is yes, assume the project is time constrained; if no, assume the project is resource constrained.

> A time-constrained project is one that must be completed by an imposed date. If required, resources can be added to ensure the project is completed by a specific date. Although time is the critical factor, resource usage should be no more than is necessary and sufficient.
>
> A resource-constrained project is one that assumes the level of resource available cannot be exceeded. If the resources are inadequate, it will be acceptable to delay the project, but as little as possible.

In scheduling terms, time constrained means time (project duration) is fixed and resources are flexible, while resource constrained means resources are fixed and time is flexible. Methods for scheduling these projects are presented in the next section.

Resource Allocation Methods

Assumptions

Ease of demonstrating the allocation methods available requires some limiting assumptions to keep attention on the heart of the problem. The rest of the chapter depends entirely on the assumptions noted here. First, splitting activities will not be allowed. This means once an activity is placed in the schedule, assume it will be worked on continuously until it is finished; hence, an activity cannot be started, stopped for a period of time, and then finished. Second, the level of resource used for an activity cannot be changed. These limiting assumptions do not exist in practice, but simplify learning. It is easy for new project managers to deal with the reality of splitting activities and changing the level of resources when they meet them on the job.

Time-Constrained Projects: Smoothing Resource Demand

Scheduling time-constrained projects focuses on resource *utilization.* When demand for a specific resource type is erratic, it is difficult to manage, and utilization may be very poor. Practitioners have attacked the utilization problem using resource leveling techniques that balance or smooth demand for a resource. Basically, all leveling techniques delay noncritical activities by using positive slack to reduce peak demand and fill in the valleys for the resources. An example will demonstrate the basic procedure for a time-constrained project. See Figure 8.2.

For the purpose of demonstration, the Botanical Garden project uses only one resource (backhoes); all backhoes are interchangeable. The top bar chart shows the activities on a time scale. The dependencies are shown with the vertical connecting arrows. The horizontal arrows following activities represent activity slack (for example, irrigation requires six days to complete and has six days slack). The number of backhoes needed for each task is shown in the shaded activity duration block (rectangle). After the land has been scarified and the plan laid out, work can begin on the walkways, irrigation, and fencing and retaining walls simultaneously. The middle chart shows the resource profile for the backhoes. For periods 4 through 10, four backhoes are needed.

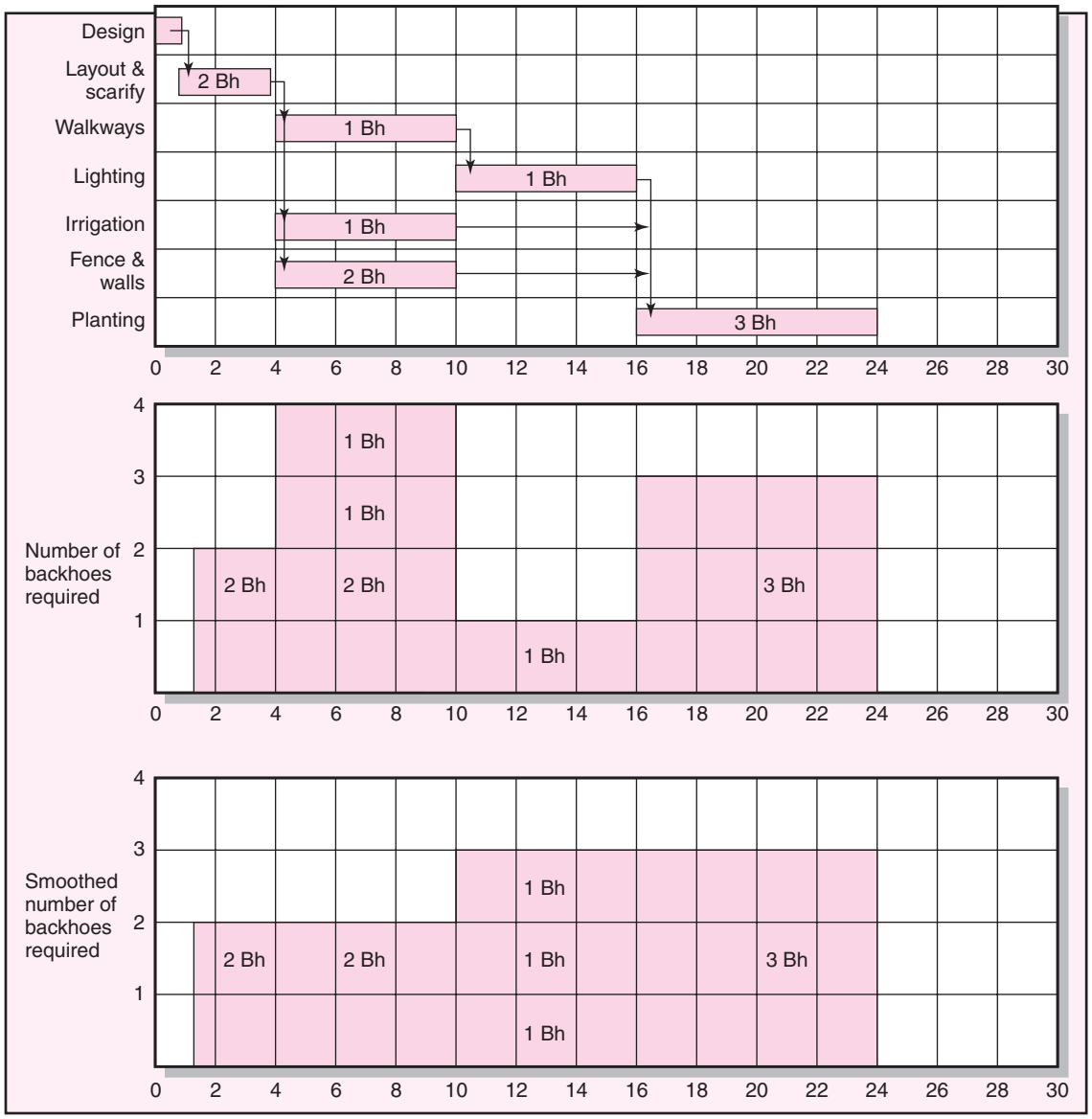
Because this project is declared time constrained, the goal will be to reduce the peak requirement for the resource and thereby increase the utilization of the resource. A quick examination of the ES (early start) resource load chart suggests only two activities have slack that can be used to reduce the peak—fence and walls provide the best choice for smoothing the resource needs. Another choice could be irrigation, but it would result in an up and down resource profile. The choice will probably center on the activity that is perceived as having the least risk of being late. The smoothed resource loading chart shows the results of delaying the fence and walls activity. Note the differences in the resource profiles. The important point is the resources needed over the life of the project have been reduced from four to three (25 percent). In addition the profile has been smoothed, which should be easier to manage.

The Botanical Garden project schedule reached the three goals of smoothing:

- The peak of demand for the resource was reduced.
- Resources over the life of the project have been reduced.
- The fluctuations in resource demand were minimized.

The latter improves the utilization of resources. Backhoes are not easily moved from location to location. There are costs associated with changing the level of resources needed. The same analogy applies to the movement of people back and forth among

FIGURE 8.2 **Botanical Garden**



projects. It is well known that people are more efficient if they can focus their effort on one project rather than multitasking their time among, say, three projects.

The downside of leveling is a loss of flexibility that occurs from reducing slack. The risk of activities delaying the project also increases because slack reduction can create more critical activities and/or near-critical activities. Pushing leveling too far for a perfectly level resource profile is risky. Every activity then becomes critical.

The Botanical Garden example gives a sense of the time-constrained problem and the smoothing approach. However, in practice the magnitude of the problem is very complex for even small projects. Manual solutions are not practical. Fortunately, the software packages available today have very good routines for leveling project

resources. Typically, they use activities that have the most slack to level project resources. The rationale is those activities with the most slack pose the least risk. Although this is generally true, other risk factors such as reduction of flexibility to use reassigned resources on other activities or the nature of the activity (easy, complex) are not addressed using such a simple rationale. It is easy to experiment with many alternatives to find the one that best fits your project and minimizes risk of delaying the project.

Resource-Constrained Projects

When the number of people and/or equipment is not adequate to meet peak demand requirements and it is impossible to obtain more, the project manager faces a resource-constrained problem. Something has to give. The trick is to prioritize and allocate resources to minimize project delay without exceeding the resource limit or altering the technical network relationships.

The resource scheduling problem is a large combinatorial one. This means even a modest-size project network with only a few resource types might have several thousand feasible solutions. A few researchers have demonstrated *optimum* mathematical solutions to the resource allocation problem but only for small networks and very few resource types. The massive data requirements for larger problems make pure mathematical solutions (e.g., linear programming) impractical. An alternative approach to the problem has been the use of heuristics (rules of thumb) to solve large combinatorial problems. These practical decision or priority rules have been in place for many years.

Heuristics do not always yield an optimal schedule, but they are very capable of yielding a "good" schedule for very complex networks with many types of resources. The efficiency of different rules and combinations of rules has been well documented. However, because each project is unique, it is wise to test several sets of heuristics on a network to determine the priority allocation rules that minimize project delay. The computer software available today makes it very easy for the project manager to create a good resource schedule for the project. A simple example of the heuristic approach is illustrated here.

Heuristics allocate resources to activities to minimize project delay; that is, heuristics prioritize which activities are allocated resources and which activities are delayed when resources are not adequate. The following scheduling heuristics have been found to consistently minimize project delay over a large variety of projects. Schedule activities using the following heuristic priority rules in the order presented:

1. Minimum slack.
2. Smallest duration.
3. Lowest activity identification number.

The parallel method is the most widely used approach to apply heuristics. The parallel method is an iterative process that starts at the first time period of the project and schedules period-by-period any activities eligible to start. In any period when two or more activities require the same resource, the priority rules are applied. For example, if in period 5 three activities are eligible to start (i.e., have the same ES) and require the same resource, the first activity placed in the schedule would be the activity with the least slack (rule 1). However, if all activities have the same slack, the next rule would be invoked (rule 2), and the activity with the smallest duration would be placed in the schedule first. In very rare cases, when all eligible activities have the same slack and the same duration, the tie is broken by the lowest activity identification number (rule 3), since each activity has a unique ID number.

When a resource limit has been reached, the early start (ES) for succeeding activities not yet in the schedule will be delayed (and all successor activities not having free slack) and their slack reduced. In subsequent periods the procedure is repeated until the project is scheduled. The procedure is demonstrated below; see Figure 8.3. The shaded areas in the resource loading chart represent the "scheduling interval" of the *time constrained* schedule (ES through LF). You can schedule the resource any place *within* the interval and not delay the project. Scheduling the activity beyond the LF will delay the project.

The programmers are limited to three. Follow the actions described in Figures 8.3 and 8.4. Note how the limit of three programmers starts to delay the project.

The Parallel Method:

Period	Action
0–1	Only activity 1 is eligible. It requires 2 programmers. Load activity 1 into schedule.
1–2	No activities are eligible to be scheduled.
2–3	Activities 2, 3, and 4 are eligible to be scheduled. Activity 3 has the least slack (0)—apply rule 1. Load Activity 3 into schedule. Activity 2 is next with slack of 2; however, activity 2 requires 2 programmers and only 1 is available. Delay activity 2. Update ES = 3, slack = 1. The next eligible activity is activity 4, since it only requires 1 programmer. Load activity 4 into schedule.
	See Figure 8.4
3–4	Activity 2 is eligible but exceeds limit of 3 programmers in pool. Delay activity 2. Update ES = 4, slack = 0.
4–5	Activity 2 is eligible but exceeds limit of 3 programmers in pool. Delay activity 2. Update ES = 5, slack = −1. Delay activity 7. Update ES = 11, slack = −1.
5–6	Activity 2 is eligible but exceeds limit of 3 programmers in pool. Delay activity 2. Update ES = 6, slack = −2. Delay activity 7. Update ES = 12, slack = −2.
6–7	Activities 2, 5, and 6 are eligible with slack of −2, 2, and 0, respectively. Load activity 2 into schedule (rule 1). Because activity 6 has 0 slack, it is the next eligible activity. Load activity 6 into schedule (rule 1). The programmer limit of 3 is reached. Delay activity 5. Update ES = 7, slack = 1.
7–8	Limit is reached. No programmers available. Delay activity 5. Update ES = 8, slack = 0.
8–9	Limit is reached. No programmers available. Delay activity 5. Update ES = 9, slack = −1.
9–10	Limit is reached. No programmers available. Delay activity 5. Update ES = 10 slack = −2.
10–11	Activity 5 is eligible. Load activity 5 into schedule. (Note: Activity 6 does not have slack because there are no programmers available—3 maximum.)
11–12	No eligible activities.
12–13	Activity 7 is eligible. Load activity 7 into schedule.

FIGURE 8.3 Resource-Constrained Schedule through Period 2–3

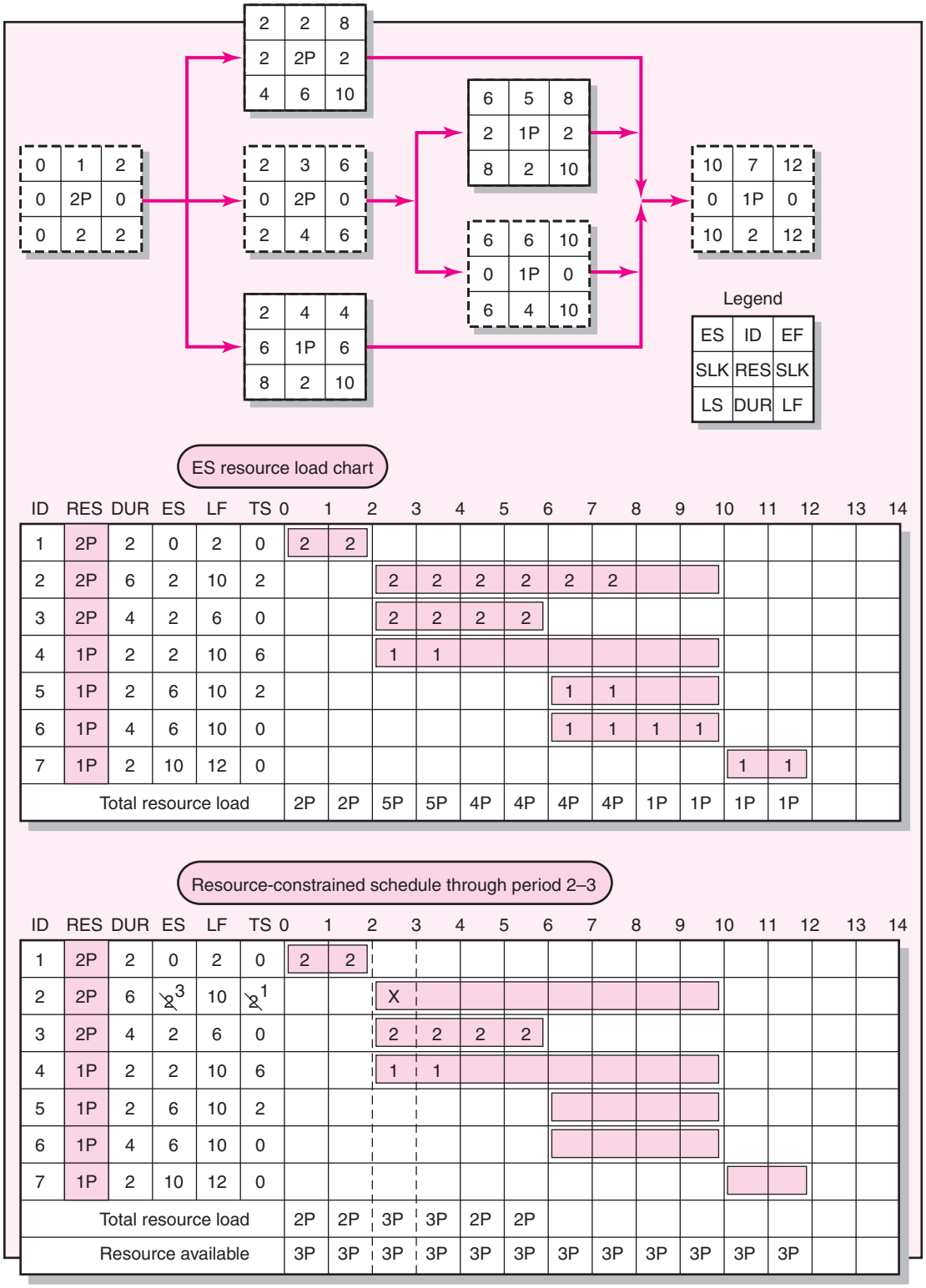


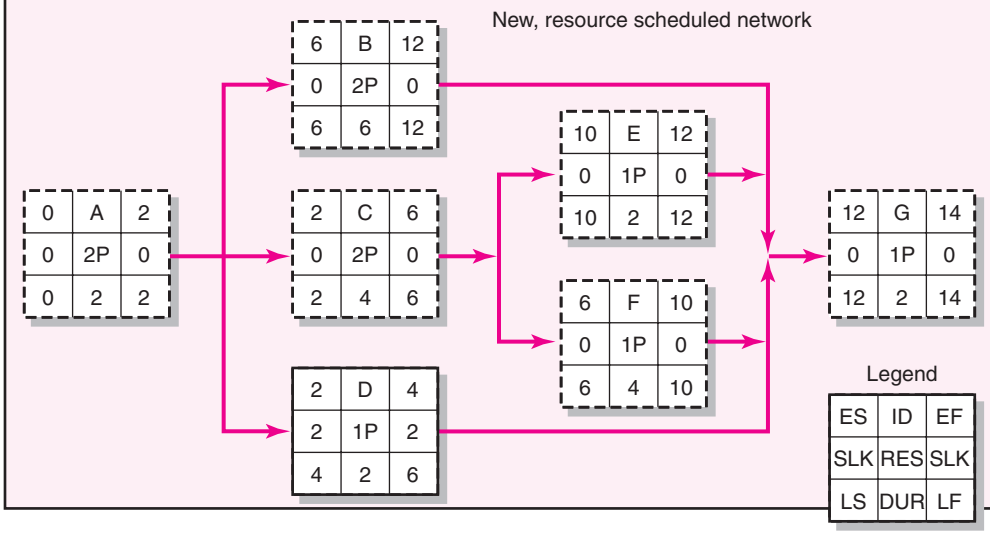
FIGURE 8.4 **Resource-Constrained Schedule through Period 5–6**

Resource constrained schedule through period 5–6

ID	RES	DUR	ES	LF	TS	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	2P	2	0	2	0	2	2													
2	2P	6	2 3 4 5 6	12	2 1 0 -1 -2			X	X	X	X									
3	2P	4	2	6	0			2	2	2	2									
4	1P	2	2	10	6			1	1											
5	1P	2	6	10	2															
6	1P	4	6	10	0															
7	1P	2	10 11 12	12 13 14	0 -1 -2											X	X			
Total resource load						2P	2P	3P	3P	2P	2P									
Resource available						3P	3P	3P	3P	3P	3P	3P	3P	3P	3P	3P	3P			

Final resource constrained schedule

ID	RES	DUR	ES	LF	TS	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	2P	2	0	2	0	2	2													
2	2P	6	2 3 4 5 6	12	2 0 -1 -2			X	X	X	X	2	2	2	2	2	2			
3	2P	4	2	6	0			2	2	2	2									
4	1P	2	2	6	6 2			1	1	S	S									
5	1P	2	6 7 8 9 10	10	2 1 0 -1 -2							X	X	X	X	1	1			
6	1P	4	6	10 11 12	0 1 2							1	1	1	1					
7	1P	2	10 11 12	12 13 14	0 -1 -2											X	X	1	1	
Total resource load						2P	2P	3P	3P	2P	2P	3P	3P	3P	3P	3P	3P	1P	1P	
Resource available						3P	3P	3P	3P	3P	3P	3P	3P	3P	3P	3P	3P	3P	3P	



Observe how it is necessary to update each period to reflect changes in activity early start and slack times so the heuristics can reflect changing priorities. The network in Figure 8.4 on the preceding page reflects the new schedule date of 14 time units, rather than the time-constrained project duration of 12 time units. The network has also been revised to reflect new start, finish, and slack times for each activity. Note that activity 6 is still critical and has a slack of 0 time units because no resources are available (they are being used on activities 2 and 5). Compare the slack for each activity found in Figures 8.3 and 8.4; slack has been reduced significantly. Note that activity 4 has only 2 units of slack rather than what appears to be 6 slack units. This occurs because only three programmers are available, and they are needed to satisfy the resource requirements of activities 2 and 5. Note that even though the project duration has increased from 12 to 14 time units, the number of critical activities (1, 2, 3, 5, 6, 7) has increased from four to six.

Computer Demonstration of Resource-Constrained Scheduling

Fortunately, project management software is capable of assessing and resolving complicated resource-constrained schedules using heuristics similar to what was described above. We will use the EMR project to demonstrate how this is done using MS Project 2002. It is important to note that the software is not "managing" the project. The software is simply a tool the project manager uses to view the project from different perspectives and conditions. See the Snapshot from Practice on page 257 for more tips on assessing resource problems.

EMR is the name given to the development of a handheld electronic medical reference guide to be used by emergency medical technicians and paramedics. Figure 8.5 contains a time-limited network for the design phase of the project. For the purpose of this example, we assume that only design engineers are required for the tasks and that the design engineers are interchangeable. The number of engineers required to perform each task is noted in the network, where 500 percent means five design engineers are needed for the activity. For example, activity 5, feature specs, requires four design engineers (400 percent). The project begins January 1, 2005, and ends February 14, 2005, a duration of 45 workdays. The time-limited (constrained) bar chart for the project is shown in Figure 8.6. This bar chart incorporates the same information used to develop the project network, but presents the project in the form of a bar chart along a time line.

Finally, a resource usage chart is presented for a segment of the project—January 15 to January 23; see Figure 8.7A. Observe that the time-limited project requires 21 design engineers on January 18 and 19 (168 hrs/8 hrs per engineer = 21 engineers). This segment represents the peak requirement for design engineers for the project. However, due to the shortage of design engineers and commitments to other projects, only eight engineers can be assigned to the project. This creates overallocation problems more clearly detailed in Figure 8.7B, which is a resource loading chart for design engineers. Notice that the peak is 21 engineers and the limit of 8 engineers is shown by the gray shaded area.

To resolve this problem we use the "leveling" tool within the software and first try to solve the problem by leveling only within slack. This solution would preserve the original finish date. However, as expected, this does not solve all of the allocation problems. The next option is to allow the software to apply scheduling heuristics and level outside of slack. The new schedule is contained in the revised, resource-limited

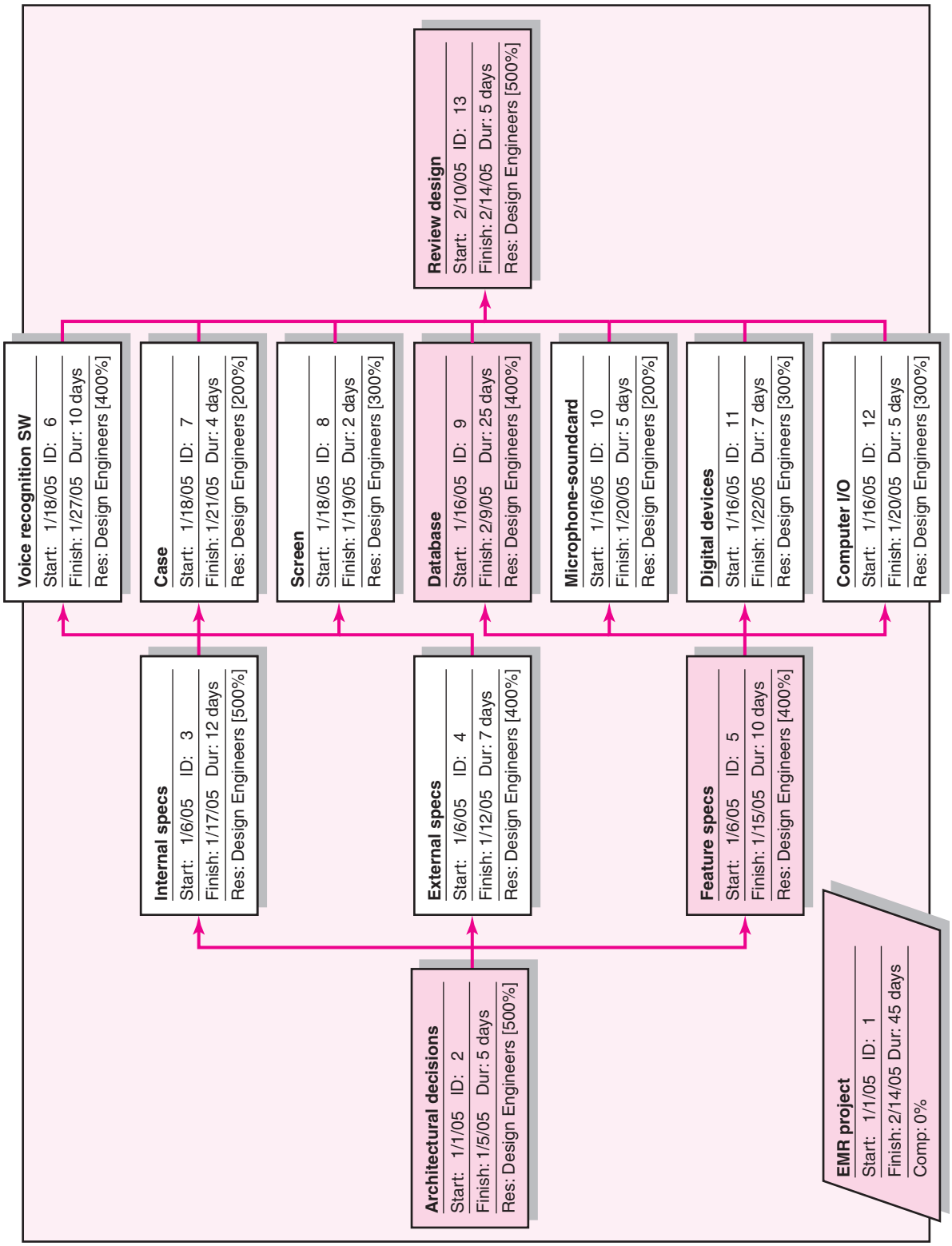


FIGURE 8.5 EMR Project Network View Schedule before Resources Leveled

FIGURE 8.6 EMR Project before Resources Added

ID	Task Name	Start	Finish	Late Start	Late Finish	Free Slack	Total Slack
1	**EMR project**	**Tue 1/1/05**	**Thu 2/14/05**	**Tue 1/1/05**	**Thu 2/14/05**	**0 days**	**0 days**
2	Architectural decisions	Tue 1/1/05	Sat 1/5/05	Tue 1/1/05	Sat 1/5/05	0 days	0 days
3	Internal specs	Sun 1/6/05	Thu 1/17/05	Sat 1/19/05	Wed 1/30/05	0 days	13 days
4	External specs	Sun 1/6/05	Sat 1/12/05	Thu 1/24/05	Wed 1/30/05	5 days	18 days
5	Feature specs	Sun 1/6/05	Tue 1/15/05	Sun 1/6/05	Tue 1/15/05	0 days	0 days
6	Voice recognition SW	Fri 1/18/05	Sun 1/27/05	Thu 1/31/05	Sat 2/9/05	13 days	13 days
7	Case	Fri 1/18/05	Mon 1/21/05	Wed 2/6/05	Sat 2/9/05	19 days	19 days
8	Screen	Fri 1/18/05	Sat 1/19/05	Fri 2/8/05	Sat 2/9/05	21 days	21 days
9	Database	Wed 1/16/05	Sat 2/9/05	Wed 1/16/05	Sat 2/9/05	0 days	0 days
10	Microphone-soundcard	Wed 1/16/05	Sun 1/20/05	Tue 2/5/05	Sat 2/9/05	20 days	20 days
11	Digital devices	Wed 1/16/05	Tue 1/22/05	Sun 2/3/05	Sat 2/9/05	18 days	18 days
12	Computer I/O	Wed 1/16/05	Sun 1/20/05	Tue 2/5/05	Sat 2/9/05	20 days	20 days
13	Review design	Sun 2/10/05	Thu 2/14/05	Sun 2/10/05	Thu 2/14/05	0 days	0 days

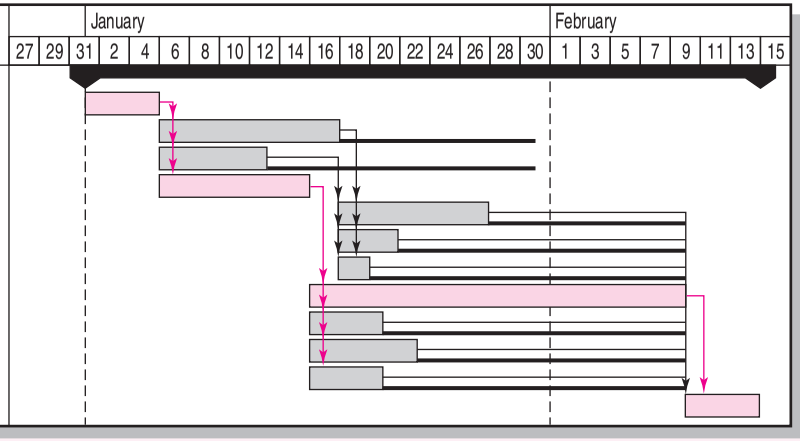
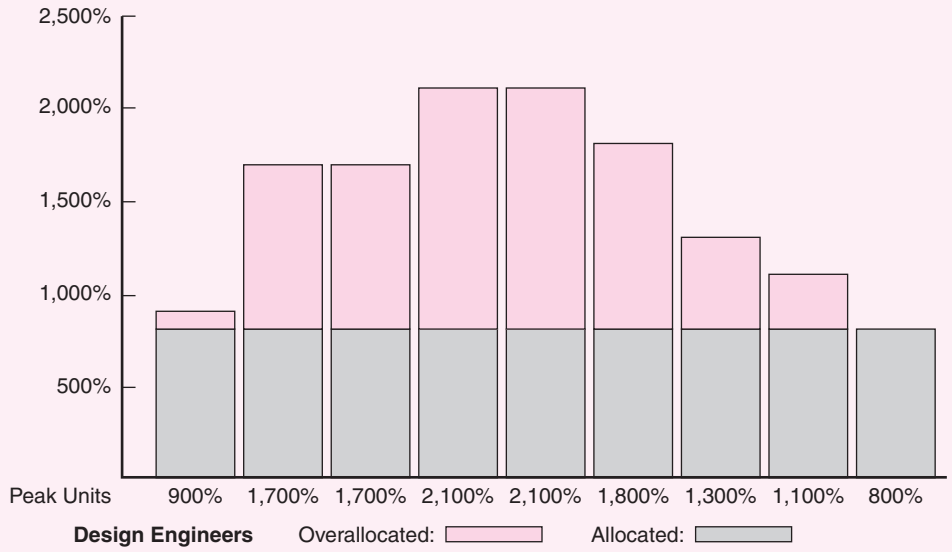


FIGURE 8.7A EMR Project—Time-Constrained Resource Usage View, January 15–23, 2005

Resource Name	Work	Jan 15, '05						Jan 21, '05		
		T	W	T	F	S	S	M	T	W
Design engineers	**3,024 hrs**	72h	136h	136h	168h	168h	144h	104h	88h	64h
Architectural decisions	200 hrs									
Internal specs	480 hrs	40h	40h	40h						
External specs	224 hrs									
Feature specs	320 hrs	32h								
Voice recognition SW	320 hrs				32h	32h	32h	32h	32h	32h
Case	64 hrs				16h	16h	16h	16h		
Screen	48 hrs				24h	24h				
Database	800 hrs		32h	32h	32h	32h	32h	32h	32h	32h
Microphone-soundcard	80 hrs		16h	16h	16h	16h	16h			
Digital devices	168 hrs		24h	24h	24h	24h	24h	24h	24h	
Computer I/O	120 hrs		24h	24h	24h	24h	24h			
Review design	200 hrs									

FIGURE 8.7B Resource Loading Chart for EMR Project, January 15–23, 2005



network chart presented in Figure 8.8. The resource-limited project network indicates the project duration has now been extended to 2/26/05, or 57 workdays (versus 45 days time limited). The critical path is now 2, 3, 9, 13.

Figure 8.9 presents the project bar chart and the results of leveling the project schedule to reflect the availability of only eight design engineers. The application of the heuristics can be seen in the scheduling of the internal, external, and feature specification activities. All three activities were originally scheduled to start immediately after activity 1, architectural decisions.

This is impossible, since the three activities collectively require 14 engineers. The software chooses to schedule activity 5 first because this activity is on the original critical path and has zero slack (heuristic # 1). Next, and concurrently, activity 4, is chosen over activity 3 because activity 4 has a shorter duration (heuristic # 2); internal specs, activity 3, is delayed due to the limitation of 8 design engineers. Notice that the original critical path no longer applies because of the resource dependencies created by having only eight design engineers.

FIGURE 8.8 EMR Project Network View Schedule after Resources Leveled

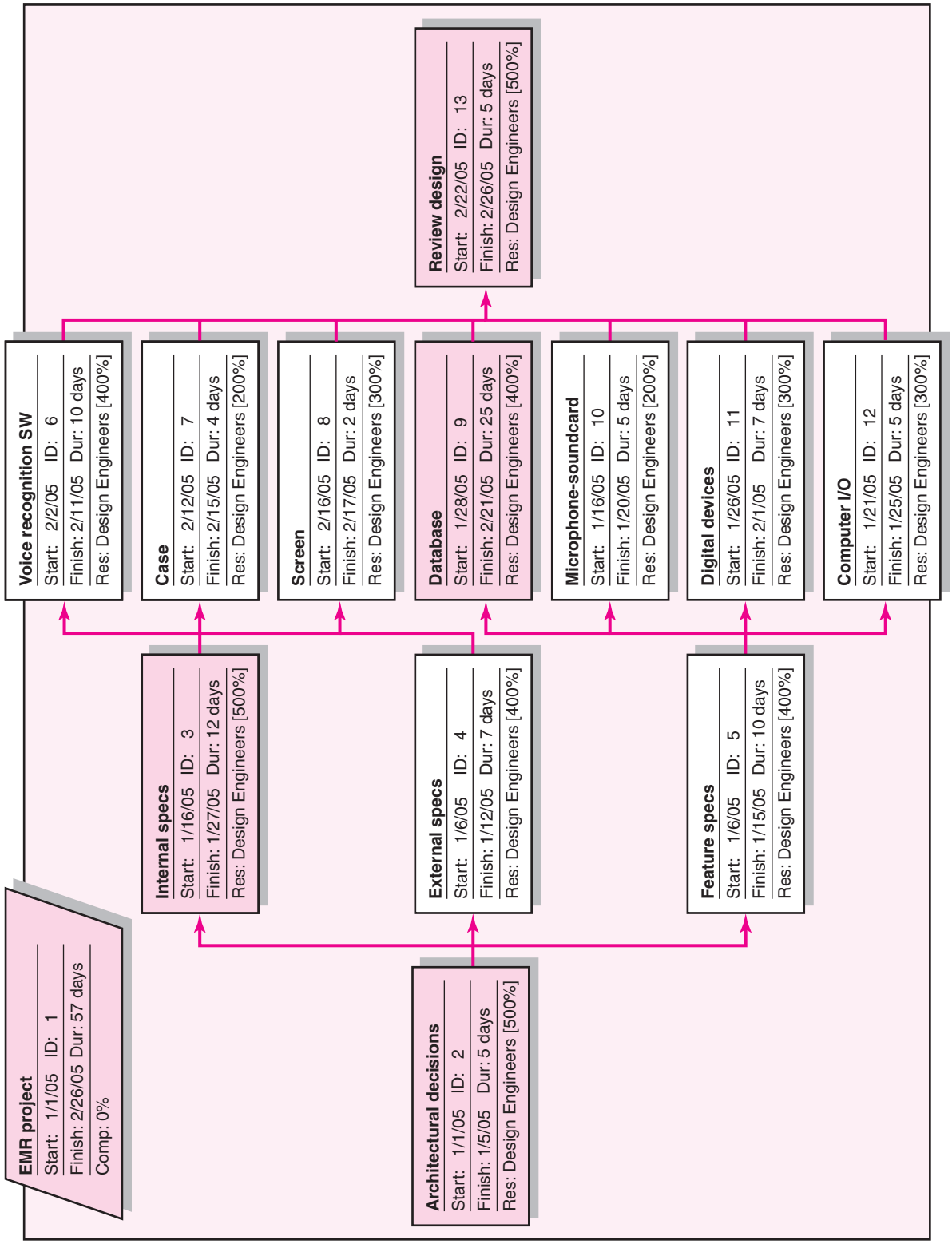
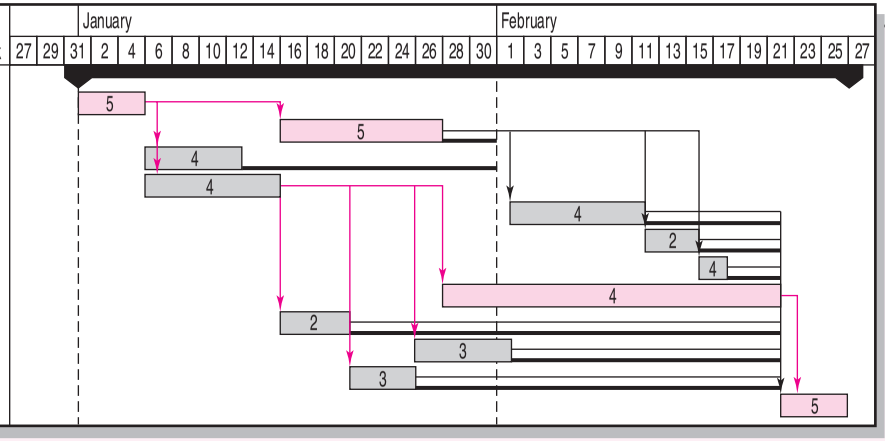


FIGURE 8.9 EMR Project Resources Leveled

ID	Task Name	Start	Finish	Late Start	Late Finish	Free Slack	Total Slack
1	**EMR project**	**Tue 1/1/05**	**Thu 2/26/05**	**Tue 1/1/05**	**Tue 2/26/05**	**0 days**	**0 days**
2	Architectural decisions	Tue 1/1/05	Sat 1/5/05	Tue 1/1/05	Sat 1/5/05	0 days	0 days
3	Internal specs	Wed 1/16/05	Sun 1/27/05	Sun 1/20/05	Thu 1/31/05	0 days	4 days
4	External specs	Sun 1/6/05	Sat 1/12/05	Fri 1/25/05	Thu 1/31/05	15 days	19 days
5	Feature specs	Sun 1/6/05	Tue 1/15/05	Sun 1/6/05	Tue 1/15/05	0 days	0 days
6	Voice recognition SW	Sat 2/2/05	Mon 2/11/05	Tue 2/12/05	Thu 2/21/05	10 days	10 days
7	Case	Tue 2/12/05	Fri 2/15/05	Mon 2/18/05	Thu 2/21/05	6 days	6 days
8	Screen	Sat 2/16/05	Sun 2/17/05	Wed 2/20/05	Thu 2/21/05	4 days	4 days
9	Database	Mon 1/28/05	Thu 2/21/05	Mon 1/28/05	Thu 2/21/05	0 days	0 days
10	Microphone-soundcard	Wed 1/16/05	Sun 1/20/05	Sun 2/17/05	Thu 2/21/05	32 days	32 days
11	Digital devices	Sat 1/26/05	Fri 2/1/05	Fri 2/15/05	Thu 2/21/05	20 days	20 days
12	Computer I/O	Mon 1/21/05	Fri 1/25/05	Sun 2/17/05	Thu 2/21/05	27 days	27 days
13	Review design	Fri 2/22/05	Tue 2/26/05	Fri 2/22/05	Tue 2/26/05	0 days	0 days



Snapshot from Practice — Assessing Resource Allocation

One of the strengths of today's project management software is the ability to identify and provide options for resolving resource allocation problems. A project manager who uses MS Project to plan projects shared with us the following checklist for dealing with resource conflicts after preliminary assignment of resources has been made.

1. Assess whether you have overallocation problems (see *Red* in the resource sheet view.)
2. Identify where and when conflicts occur by examining the resource usage view.
3. Resolve the problem by
   a. Replacing overallocated resources with appropriate resources that are available. Then ask if this solves the problem.

   If not:

   b. Use the leveling tool and choose the level within slack option.
      i. Does this solve the problem? (Are resources still overallocated?)
      ii. Check the sensitivity of the network and ask if this is acceptable.

   If not:

   c. Consider splitting tasks.
      i. Make sure to readjust task durations to take into account additional start-up and shutdown time.
4. If 3 does not work then either:
   a. Use level tool default option and ask if you can live with the new completion date.

   If not:

   b. Negotiate for additional resources to complete the project.

   If not possible

   c. Consider reducing project scope to meet deadline.

While this checklist makes specific references to MS Project, the same steps can be used with most project management software.

Compare the bar chart in Figure 8.9 with the time-limited bar chart in Figure 8.6. For example, note the different start dates for activity 8 (screen). In the time-limited plan (Figure 8.6), the start date for activity 8 is 1/18, while the start date in the resource limited schedule (Figure 8.9) is 2/26, over a month later!

While resource bar graphs are commonly used to illustrate overallocation problems, we prefer to view resource usage tables like the one presented in Figure 8.7A. This table tells you when you have an overallocation problem and identifies activities that are causing the overallocation.

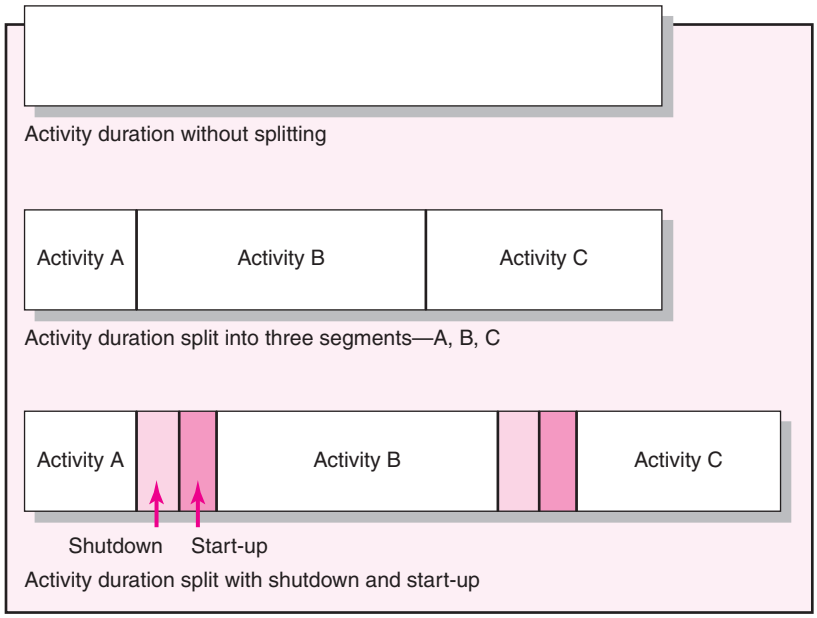
The Impacts of Resource-Constrained Scheduling

Like leveling schedules, the limited resource schedule usually reduces slack, reduces flexibility by using slack to ensure delay is minimized, and increases the number of critical and near-critical activities. Scheduling complexity is increased because resource constraints are added to technical constraints; start times may now have two constraints. The traditional critical path concept of sequential activities from the start to the end of the project is no longer meaningful. The resource constraints can break the sequence and leave the network with a set of disjointed critical activities. Conversely, parallel activities can become sequential. Activities with slack on a time-constrained network can change from critical to noncritical.

Splitting/Multitasking

Splitting or multitasking is a scheduling technique used to get a better project schedule and/or to increase resource utilization. A planner splits the continuous work included in an activity by interrupting the work and sending the resource to another activity for a period of time and then having the resource resume work on the original

FIGURE 8.10
Splitting/ Multitasking



activity. Splitting can be a useful tool if the work involved does not include large start-up or shutdown costs—for example, moving equipment from one activity location to another. The most common error is to interrupt "people work," where there are high conceptual start-up and shutdown costs. For example, having a bridge designer take time off to work on the design problem of another project may cause this individual to lose four days shifting conceptual gears in and out of two activities. The cost may be hidden, but it is real. Figure 8.10 depicts the nature of the splitting problem. The original activity has been split into three separate activities: A, B, and C. The shutdown and start-up times lengthen the time for the original activity.

Some have argued that the propensity to deal with resource shortages by multitasking is a major reason why projects fail to meet schedule. We agree. Planners should avoid the use of splitting as much as possible, except in situations where splitting costs are known to be small or when there is no alternative for resolving the resource problem. Computer software offers the splitting option for each activity; use it sparingly. See Snapshot from Practice: Assessing Resource Allocation.

Benefits of Scheduling Resources

It is important to remember that, if resources are truly limited and activity time estimates are accurate, the resource-constrained schedule *will* materialize as the project is implemented—*not* the time-constrained schedule! Therefore, failure to schedule limited resources can lead to serious problems for a project manager. The benefit of creating this schedule *before* the project begins leaves time for considering reasonable alternatives. If the scheduled delay is unacceptable or the risk of being delayed too high, the assumption of being resource constrained can be reassessed. Cost-time trade-offs can be considered. In some cases priorities may be changed. See Snapshot from Practice: U.S. Forest Service Resource Shortage.

Snapshot from Practice

U.S. Forest Service Resource Shortage

A major segment of work in managing U.S. Forest Service (USFS) forests is selling mature timber to logging companies that harvest the timber under contract conditions monitored by the Service. The proceeds are returned to the federal government. The budget allocated to each forest depends on the two-year plan submitted to the U.S. Department of Agriculture.

Olympic Forest headquarters in Olympia, Washington, was developing a two-year plan as a basis for funding. All of the districts in the forest submitted their timber sale projects (numbering more than 50) to headquarters, where they were compiled and aggregated into a project plan for the whole forest. The first computer run was reviewed by a small group of senior managers to determine if the plan was reasonable and "doable." Management was pleased and relieved to note all projects appeared to be doable in the two-year time frame until a question was raised concerning the computer printout. "Why are all the columns in these projects labeled 'RESOURCE' blank?" The response from an engineer was, "We don't use that part of the program."

The discussion that ensued recognized the importance of resources in completing the two-year plan and ended with a request to "try the program with resources included." The new output was startling. The two-year program turned into a three-and-a-half-year plan because of the shortage of specific labor skills such as road engineer and environmental impact specialist. Analysis showed that adding only three skilled people would allow the two-year plan to be completed on time. In addition, further analysis showed hiring only a few more skilled people, beyond the three, would allow an extra year of projects to also be compressed into the two-year plan. This would result in additional revenue of more than $3 million. The Department of Agriculture quickly approved the requested extra dollars for additional staff to generate the extra revenue.

Resource schedules provide the information needed to prepare time-phased work package budgets with dates. Once established, they provide a quick means for a project manager to gauge the impact of unforeseen events such as turnover, equipment breakdowns, or transfer of project personnel. Resource schedules also allow project managers to assess how much flexibility they have over certain resources. This is useful when they receive requests from other managers to borrow or share resources. Honoring such requests creates goodwill and an "IOU" that can be cashed in during a time of need.

Assigning Project Work

When making individual assignments, project managers should match, as best they can, the demands and requirements of specific work with the qualifications and experience of available participants. In doing so, there is a natural tendency to assign the best people the most difficult tasks. Project managers need to be careful not to overdo this. Over time these people may grow to resent the fact that they are always given the toughest assignments. At the same time, less experienced participants may resent the fact that they are never given the opportunity to expand their skill/knowledge base. Project managers need to balance task performance with the need to develop the talents of people assigned to the project.

Project managers not only need to decide who does what but who works with whom. A number of factors need to be considered in deciding who should work together. First, to minimize unnecessary tension, managers should pick people with compatible work habits and personalities but who complement each other (i.e., one person's weakness is the other person's strength). For example, one person may be brilliant at solving complex problems but sloppy at documenting his or her progress. It would be wise to pair this person with an individual who is good at paying attention to details. Experience is another factor. Veterans should be teamed up with new hires—not only so they can share their experience but also to help socialize the newcomers to the

Snapshot from Practice Managing Geeks*

Eric Schmidt, after a successful career at Sun Microsystems, took over struggling Novell, Inc., and helped turn it around within two years. One of the keys to his success is his ability to manage the technical wizards who develop the sophisticated systems, hardware, and software that are the backbone of electronically driven companies. He uses the term "geek" (and he can, since he is one, with a Ph.D. in computer science) to describe this group of technologists who rule the cyberworld.

Schmidt has some interesting ideas about assigning geeks to projects. He believes that putting geeks together in project teams with other geeks creates productive peer pressure. Geeks care a great deal about how other geeks perceive them. They are good at judging the quality of technical work and are quick to praise as well as criticize each other's work. Some geeks can be unbearably arrogant, but Schmidt claims that having them work together on projects is the best way to control them—by letting them control each other.

At the same time, Schmidt argues that too many geeks spoil the soup. By this he means that, when there are too many geeks on a development team, there is a tendency for intense technical navel gazing. Members lose sight of deadlines, and delays are inevitable. To combat this tendency, he recommends using geeks only in small groups. He urges breaking up large projects into smaller, more manageable projects so that small teams of geeks can be assigned to them. This keeps the project on time and makes the teams responsible to each other.

*Mitchel Russ, "How to Manage Geeks," *Fast Company* (June 1999), pp. 175–80.

customs and norms of the organization. Finally, future needs should be considered. If managers have some people who have never worked together before but who have to later on in the project, they may be wise to take advantage of opportunities to have these people work together early on so that they can become familiar with each other. Finally, see the Snapshot in Practice: Managing Geeks for some interesting thoughts about how Novell, Inc., puts together teams.

Multiproject Resource Schedules

For clarity we have discussed key resource allocation issues within the context of a single project. In reality resource allocation generally occurs in a multiproject environment where the demands of one project have to be reconciled with the needs of other projects. Organizations must develop and manage systems for efficiently allocating and scheduling resources across several projects with different priorities, resource requirements, sets of activities, and risks. The system must be dynamic and capable of accommodating new projects as well as reallocating resources once project work is completed. While the same resource issues and principles that apply to a single project also apply to this multiproject environment, application and solutions are more complex, given the interdependency among projects.

The following lists three of the more common problems encountered in managing multiproject resource schedules. Note that these are macro manifestations of single-project problems that are now magnified in a multiproject environment:

1. **Overall schedule slippage.** Because projects often share resources, delays in one project can have a ripple effect and delay other projects. For example, work on one software development project can grind to a halt because the coders scheduled for the next critical task are late in completing their work on another development project.
2. **Inefficient resource utilization.** Because projects have different schedules and requirements, there are peaks and valleys in overall resource demands. For example, a firm may have a staff of 10 electricians to meet peak demands when, under normal conditions, only 5 electricians are required.

Snapshot from Practice Multiple Project Resource Scheduling

The case for a central source to oversee project resource scheduling is well known by practitioners. Here is a synopsis of a conversation with one middle manager.

Interviewer: Congratulations on acceptance of your multiproject scheduling proposal. Everyone tells me you were very convincing.

Middle Manager: Thanks. Gaining acceptance was easy this time. The board quickly recognized we have no choice if we are to keep ahead of competition by placing our resources on the right projects.

Interviewer: Have you presented this to the board before?

Middle Manager: Yes, but not this company. I presented the same spiel to the firm I worked for two years ago. For their annual review meeting I was charged to present a proposal suggesting the need and benefits of central capacity resource planning for managing the projects of the firm.

I tried to build a case for bringing projects under one umbrella to standardize practices and to forecast and assign key people to mission critical projects. I explained how benefits such as resource demands will be aligned with mission critical projects, proactive resource planning, and a tool for catching resource bottlenecks and resolving conflicts.

Almost everyone agreed the idea was a good one. I felt good about the presentation and felt confident something was going to happen. But the idea never really got off the ground; it just faded into the sunset.

With hindsight, managers really did not trust colleagues in other departments, so they only gave half-hearted support to central resource planning. Managers wanted to protect their turf and ensure that they would not have to give up power. The culture there was simply too inflexible for the world we live in today. They are still struggling with constant conflicts among projects.

I'm glad I made the switch to this firm. The culture here is much more team-oriented. Management is committed to improving performance.

3. **Resource bottlenecks.** Delays and schedules are extended as a result of shortages of critical resources that are required by multiple projects. For example, at one Lattice Semiconductor facility, project schedules were delayed because of competition over access to test equipment necessary to debug programs. Likewise, several projects at a U.S. forest area were extended because there was only one silviculturist on the staff.

To deal with these problems, more and more companies create project offices or departments to oversee the scheduling of resources across multiple projects. One approach to multiple project resource scheduling is to use a first come–first served rule. A project queue system is created in which projects currently underway take precedence over new projects. New project schedules are based on the projected availability of resources. This queuing tends to lead to more reliable completion estimates and is preferred on contracted projects that have stiff penalties for being late. The disadvantages of this deceptively simple approach are that it does not optimally utilize resources or take into account the priority of the project. See the Snapshot from Practice: Multiple Project Resource Scheduling.

Many companies utilize more elaborate processes for scheduling resources to increase the capacity of the organization to initiate projects. Most of these methods approach the problem by treating individual projects as part of one big project and adapting the scheduling heuristics previously introduced to this "megaproject." Project schedulers monitor resource usage and provide updated schedules based on progress and resource availability across all projects. One major improvement in project management software in recent years is the ability to prioritize resource allocation to specific projects. Projects can be prioritized in ascending order (e.g., 1, 2, 3, 4, . . .), and these priorities will override scheduling heuristics so that resources go to the project highest on the priority list. (Note: This improvement fits perfectly with organizations that use project priority models similar to those described in Chapter 2.) Centralized project scheduling also makes it easier to identify resource bottlenecks that stifle

progress on projects. Once identified, the impact of the bottlenecks can be documented and used to justify acquiring additional equipment, recruiting critical personnel, or delaying the project.

Finally, many companies are using outsourcing as a means for dealing with their resource allocation problems. In some cases, a company will reduce the number of projects they have to manage internally to only core projects and outsource noncritical projects to contractors and consulting firms. In other cases, specific segments of projects are outsourced to overcome resource deficiencies and scheduling problems. Companies may hire temporary workers to expedite certain activities that are falling behind schedule or contract project work during peak periods when there are insufficient internal resources to meet the demands of all projects. The ability to more efficiently manage the ebbs and flows of project work is one of the major driving forces behind outsourcing today.

Summary

Usage and availability of resources are major problem areas for project managers. Attention to these areas in developing a project schedule can point out resource bottlenecks before the project begins. Project managers should understand the ramifications of failing to schedule resources. The results of resource scheduling are frequently significantly different from the results of the standard CPM method.

With the rapid changes in technology and emphasis on time-to-market, catching resource usage and availability problems before the project starts can save the costs of crashing project activities later. Any resource deviations from plan and schedule that occur when the project is being implemented can be quickly recorded and the effect noted. Without this immediate update capability, the real negative effect of a change may not be known until it happens. Tying resource availability to a multiproject, multiresource system supports a project priority process that selects projects by their contribution to the organization's objectives and strategic plan.

Assignment of individuals to projects may not fit well with those assigned by computer software routines. In these cases overriding the computer solution to accommodate individual differences and skills is almost always the best choice.

Key Terms

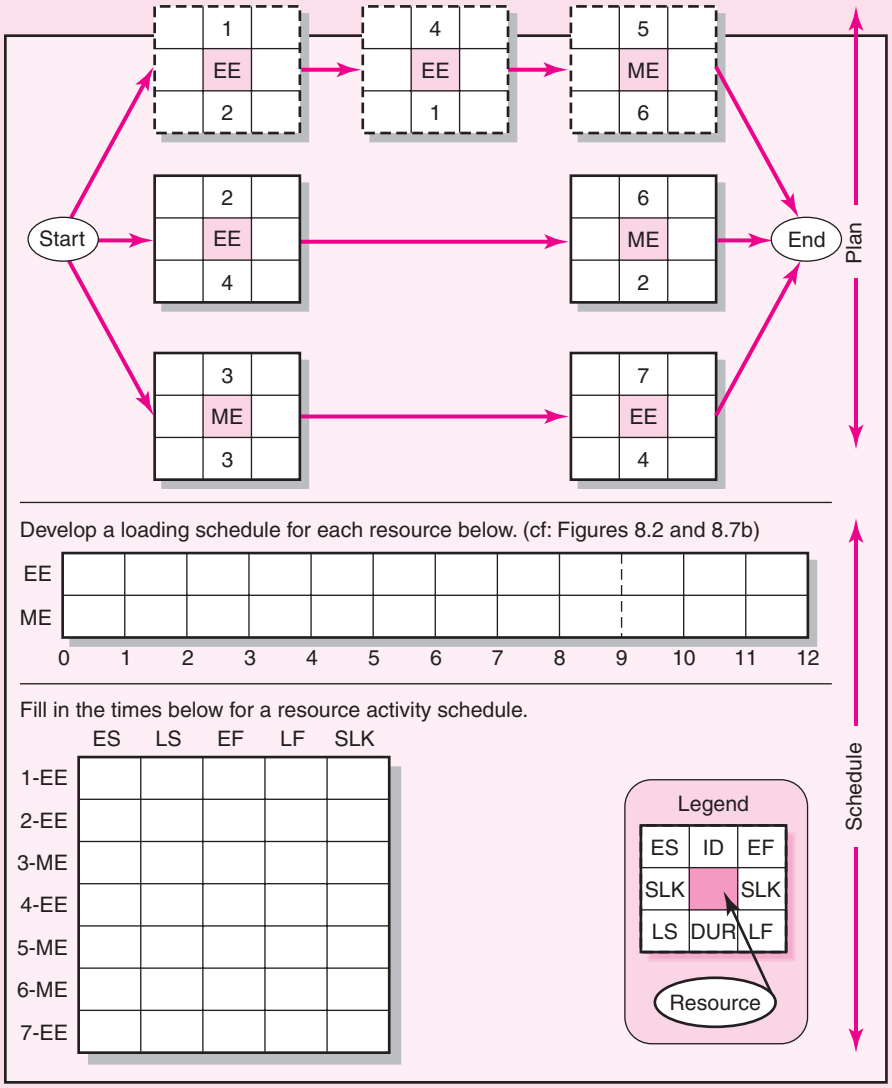
Heuristic
Leveling/smoothing
Multitasking
Resource-constrained projects
Resource profile
Splitting
Time-constrained projects

Review Questions

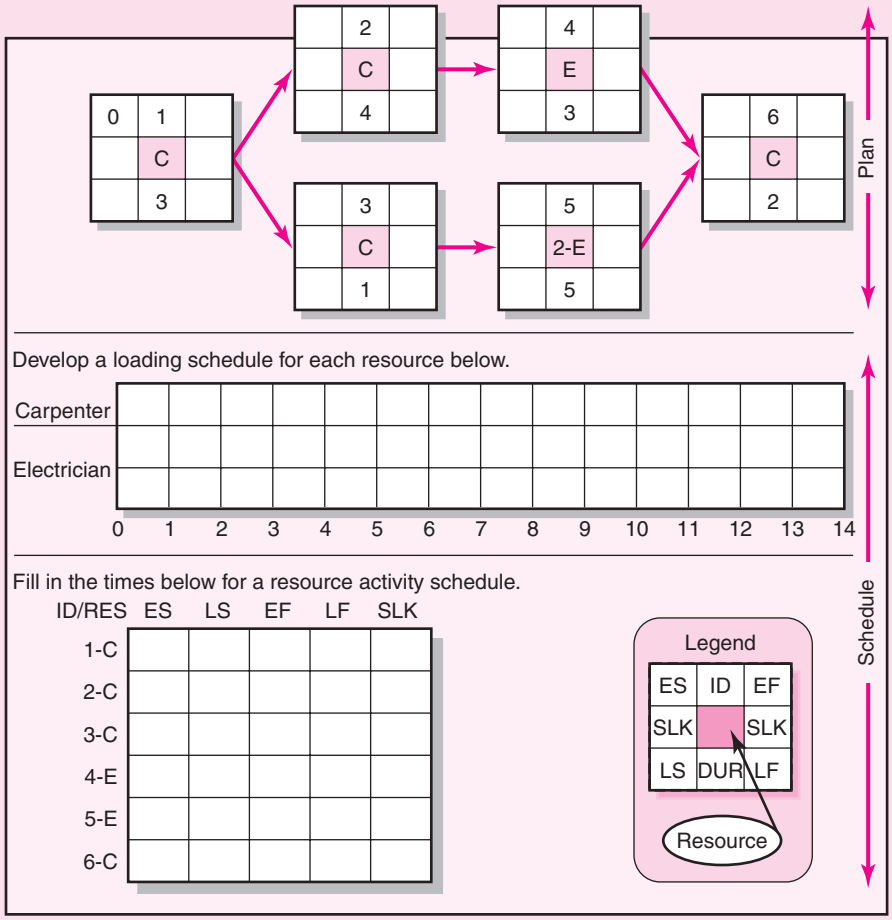
1. How does resource scheduling tie to project priority?
2. How does resource scheduling reduce flexibility in managing projects?
3. Present six reasons scheduling resources is an important task.
4. How can outsourcing project work alleviate the three most common problems associated with multiproject resource scheduling?
5. Explain the risks associated with leveling resources, compressing or crashing projects, and imposed durations or "catch-up" as the project is being implemented.

Exercises

1. Given the network plan that follows, compute the early, late, and slack times. What is the project duration? Using any approach you wish (e.g., trial and error), develop a loading chart for resources, Electrical Engineers (EE), and resource, Mechanical Engineers (ME). Assume only one of each resource exists. Given your resource schedule, compute the early, late, and slack times for your project. Which activities are now critical? What is the project duration now? Could something like this happen in real projects?



2. Given the network plan that follows, compute the early, late, and slack times. What is the project duration? Using any approach you wish (e.g., trial and error), develop a loading chart for resources Carpenters (C) and Electricians (E). Assume only one Carpenter is available and two Electricians are available. Given your resource schedule, compute the early, late, and slack times for your project. Which activities are now critical? What is the project duration now?



3. Compute the early, late, and slack times for the activities in the network that follows, assuming a time-constrained network. Which activities are critical? What is the time constrained project duration?

   Note: Recall, in the schedule resource load chart the *time constrained* "scheduling interval (ES through LF) has been shaded. Any resource scheduled beyond the shaded area will delay the project.

Assume you are a computer using a software that schedules projects by the parallel method and following heuristics. Schedule only one period at a time!

Minimum slack

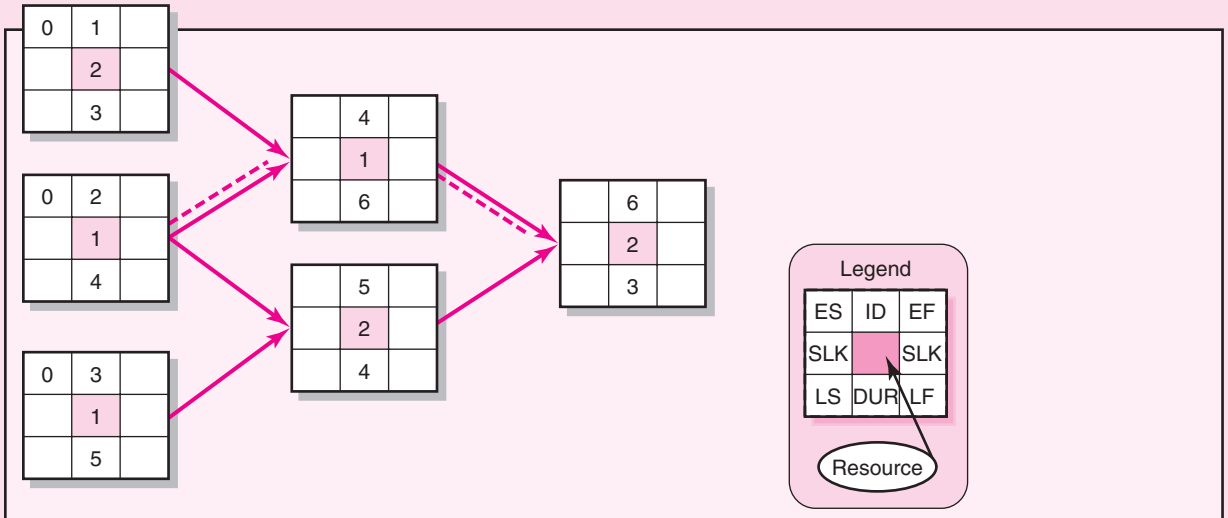
Smallest duration

Lowest identification number

Keep a log of each activity change and update you make each period—e.g., period 0–1, 1–2, 2–3, etc. (Use a format similar to the one on page 248.) The log should include any changes or updates in ES and slack times each period, activities scheduled, and activities delayed. (Hint: Remember to maintain the technical dependencies of the network.) Use the resource load chart to assist you in scheduling (see pages 249–250).

List the order in which you scheduled the activities of the project. Which activities of your schedule are now critical?

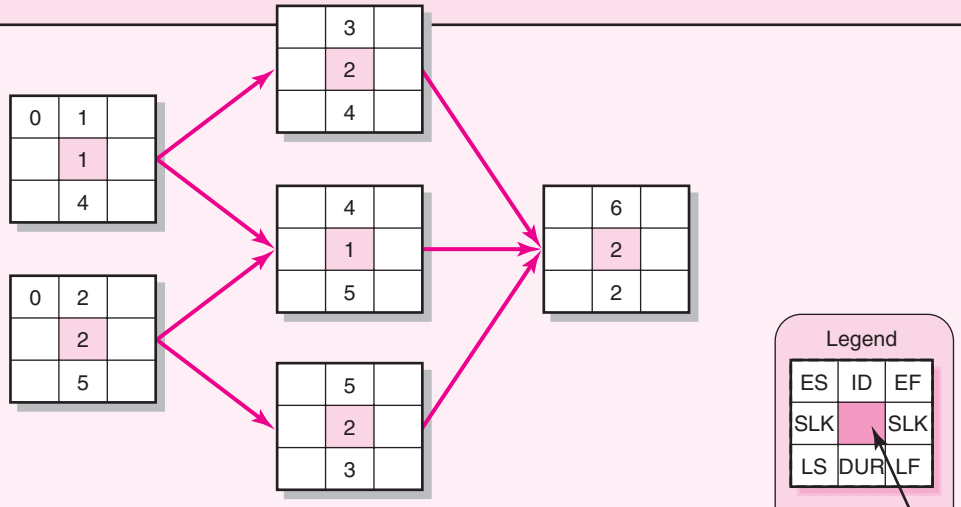
Recompute your slack for each activity given your new schedule. What is the slack for activity 1? 4? 5?

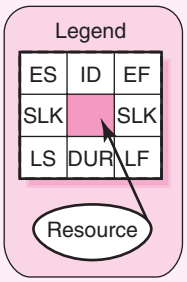


Scheduled resource load chart with ES and slack updates

ID	RES	DUR	ES	LF	SLK	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	3	0	4	1																
2	1	4	0	4	0																
3	1	5	0	6	1																
4	1	6	4	10	0																
5	2	4	5	10	1																
6	2	3	10	13	0																
Resources scheduled																					
Resources available						3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

4. Develop a resource schedule in the loading chart that follows. Use the parallel method and heuristics given. Be sure to update each period as the computer would do. Note: Activities 2, 3, 5, and 6 use two of the resource skills. Three of the resource skills are available.





Complete a time-constrained plan in the project network.

Use the following heuristics:
Minimum slack
Smallest duration
Lowest identification number

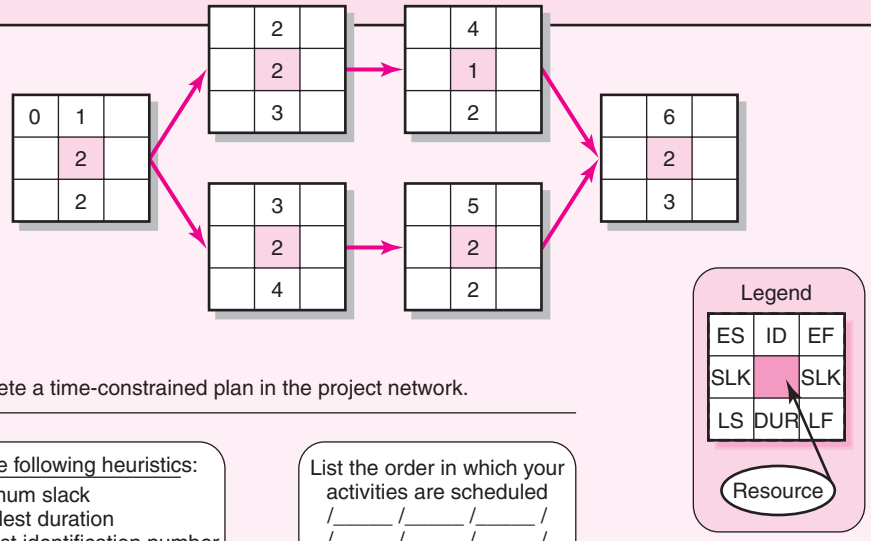
List the order in which your activities are scheduled
/_____ /_____ /_____ /
/_____ /_____ /_____ /

Develop a resource-constrained schedule in the loading chart below.

ID	RES	DUR	ES	LF	SLK	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	4	0	5	1																
2	2	5	0	5																	
3	2	4	4	10																	
4	1	5	5	10																	
5	2	3																			
6	2	2																			
Resources scheduled																					
Resources available						3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

What is the schedule slack for 1____, 2____, and 4_____?
Which activities are critical now? _____

5. Develop a resource schedule in the loading chart that follows. Use the parallel method and heuristics given. Be sure to update each period as the computer would do. Note: Activities 1, 2, 3, 5, and 6 use two of the resource skills. Three of the resource skills are available.



Complete a time-constrained plan in the project network.

Use the following heuristics:
Minimum slack
Smallest duration
Lowest identification number

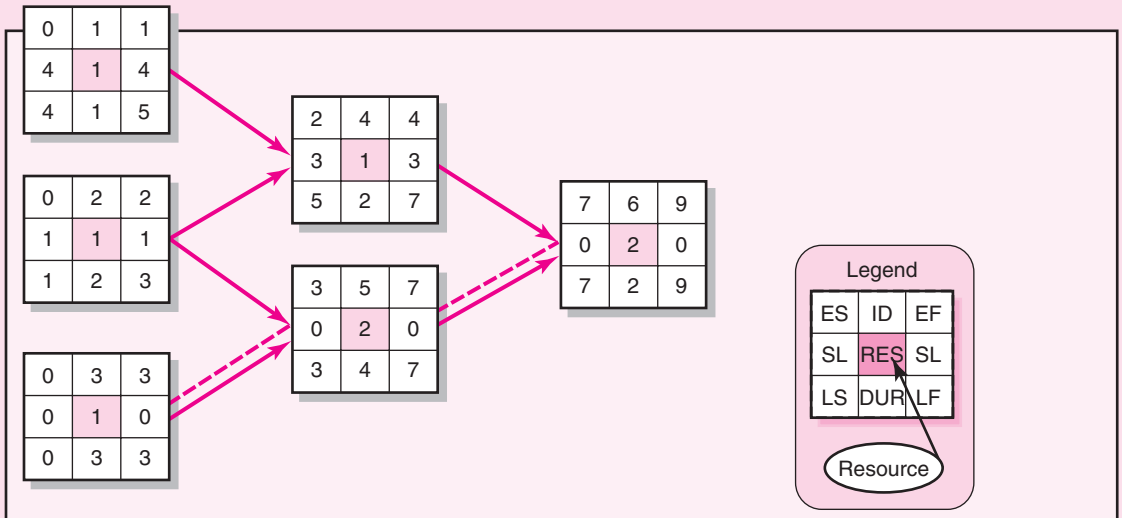
List the order in which your activities are scheduled
/_____ /_____ /_____ /
/_____ /_____ /_____ /

ID	RES	DUR	ES	LF	SLK	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	2	2	0	2	0	2	2															
2	2	3	2	6	1																	
3	2	4	2	6	0																	
4	1	2																				
5	2	2																				
6	2	3																				
Resources scheduled																						
Resources available						3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

What is the schedule slack for 1____, 2____3, and 4_____?
Which activities are critical now? _____

6. You have prepared the following schedule for a project in which the key resource is a backhoe. This schedule is contingent on having 3 backhoes. You receive a call from your partner, Brooker, who desperately needs 1 of your backhoes. You tell Brooker you would be willing to let him have the backhoe if you are still able to complete your project in 11 months.

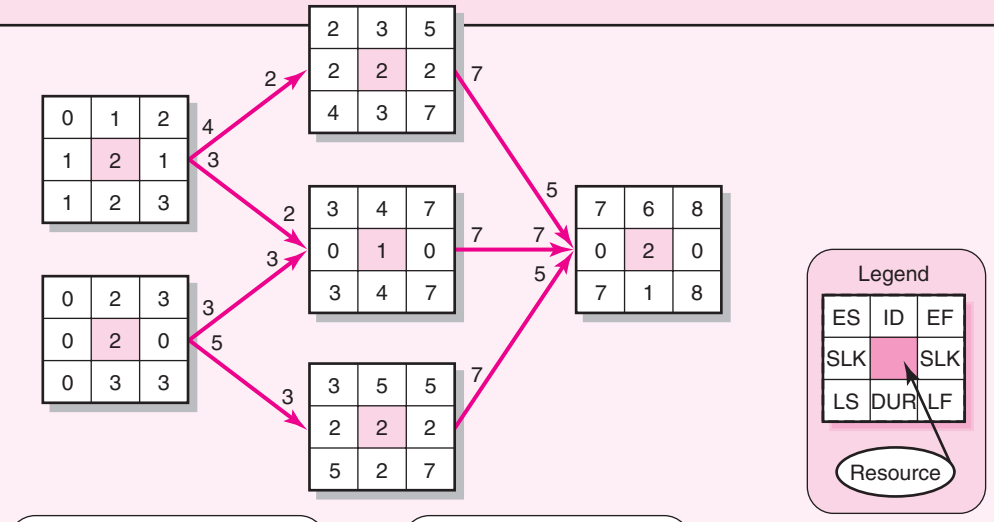
   Develop a resource schedule in the loading chart that follows to see if it is possible to complete the project in 11 months with only 2 backhoes. Be sure to record the order in which you schedule the activities using scheduling heuristics. Activities 5 and 6 require 2 backhoes, while activities 1, 2, 3, and 4 require 1 backhoe. No splitting of activities is possible. Can you say yes to Brooker's request?



Schedule the resource load chart with ES and Slack updates

ID	RES	DUR	ES	LF	SLK	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	1	0	5	4														
2	1	2	0	3	1														
3	1	3	0	3	0														
4	1	2	2	7	3														
5	2	4	3	7	0														
6	2	2	7	9	0														
Resources scheduled																			
Resources available						2	2	2	2	2	2	2	2	2	2	2	2	2	2

7. Develop a resource schedule in the loading chart that follows. Use the parallel method and heuristics given. Be sure to update each period as the computer would do. Note: Activities 1, 3, 5, and 6 use two of the resource skills. Three of the resource skills are available.



Use the following heuristics:
Minimum slack
Smallest duration
Lowest identification number

List the order in which your activities are scheduled
/_____ /_____ /_____ /
/_____ /_____ /_____ /

ID	RES	DUR	ES	LF	SLK	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2	2	0	3	1														
2	2	3	0	3	0														
3	2	3	2	7	2														
4	1	4	3	7	0														
5	2	2	3	7	2														
6	2	1	7	8	0														
Resources scheduled																			
Resources available						3	3	3	3	3	3	3	3	3	3	3	3	3	

What is the schedule slack for 1____, 2____, and 4_____ after loading?
Which activities are critical now? _____

References

Arrow, K. J., and L. Hurowicz, *Studies in Resource Allocation Process* (New York: Cambridge University Press, 1997).

Brucker, P., A. Drexl, R. Mohring, L. Newmann, and E. Pesch, "Resource-constrained Project Scheduling: Notation, Classification, Models and Methods," *European Journal of Operational Research,* 112 (1), 1999, pp. 3–42.

Burgess, A. R., and J. B. Kellebrew, "Variations in Activity Level on Cyclical Arrow Diagrams," *Journal of Industrial Engineering,* Vol. 13 March–April 1962, pp. 76–83.

Charnes, A., and W. W. Cooper, "A Network Interpretation and Direct Sub Dual Algorithm for Critical Path Scheduling," *Journal of Industrial Engineering,* July–August 1962.

Demeulemeester, E. L., and W. S. Herroelen, *Project Scheduling: A Research Handbook* (Norwell, Mass: Kluwer Academic Publishers, 2002).

Fendly, L. G., "Towards the Development of a Complete Multi Project Scheduling System," *Journal of Industrial Engineering,* Vol. 19, 1968, pp. 505–155.

Reinersten, D., "Is it Always a Bad Idea to Add Resources to a Late Project?" *Electric Design,* October 30, 2000, pp. 17–18.

Talbot, B. F., and J. H. Patterson, "Optimal Methods for Scheduling Under Resource Constraints," *Project Management Journal,* December 1979.

Wiest, J. D., "A Heuristic Model for Scheduling Large Projects with Unlimited Resources," *Management Science,* Vol. 18, February 1967, pp. 359–77.

Woodworth, B. M., and C. J. Willie, "A Heuristic Algorithm for Resource Leveling in Multiproject, Multiresource Scheduling," *Decision Sciences,* Vol. 6, July 1975, pp. 525–40.

Woodworth, B. M., and S. Shanahan, "Identifying the Critical Sequence in a Resource Constrained Project," *International Journal of Project Management,* 6 (2) 1988, pp. 89–96.

Case

Power Train, Ltd.

We have smashing systems for reporting, tracking, and controlling costs on design projects. Our planning of projects is better than any I have seen at other companies. Our scheduling seemed to serve us well when we were small and we had only a few projects. Now that we have many more projects and schedule using multiproject software, there are too many occasions when the right people are not assigned to the projects deemed important to our success. This situation is costing us big money, headaches, and stress!

Claude Jones, VP, Design and Operations

HISTORY

Power Train, Ltd. (PT), was founded in 1960 by Daniel Gage, a skilled mechanical engineer and machinist. Prior to founding PT he worked for three years as design engineer for a company that designed and built transmissions for military tanks and trucks. It was a natural transition for Dan to start a company designing and building power trains for farm tractor companies. Today, Dan is no longer active in the management of PT but is still revered as its founder. He and his family still own 25 percent of the company, which went public in 1988. PT has been growing at a 6 percent clip for the last five years but expects industry growth to level off as supply exceeds demand.

Today, PT continues its proud tradition of designing and building the best-quality power trains for manufacturers of farm tractors and equipment. The company employs 178 design engineers and has about 1,800 production and support staff. Contract

design projects for tractor manufacturers represent a major portion of PT's revenue. At any given time, about 45 to 60 design projects are going on concurrently. A small portion of their design work is for military vehicles. PT only accepts military contracts that involve very advanced, new technology and are cost plus.

A new phenomenon has attracted management of PT to look into a larger market. Last year a large Swedish truck manufacturer approached PT to consider designing power trains for its trucks. As the industry consolidates, the opportunities for PT should increase because these large firms are moving to more outsourcing to cut infrastructure costs and stay very flexible. Only last week a PT design engineer spoke to a German truck manufacturing manager at a conference. The German manager was already exploring outsourcing of drive trains to Porsche and was very pleased to be reminded of PT's expertise in the area. A meeting is set up for next month.

CLAUDE JONES

Claude Jones joined PT in 1989 as a new MBA from the University of Edinburgh. He worked as a mechanical engineer for U.K. Hydraulics for five years prior to returning to school for the MBA. "I just wanted to be part of the management team and where the action is." Jones moved quickly through the ranks. Today he is the vice president of design and operations. Sitting at his desk, Jones is pondering the conflicts and confusion that seem to be increasing in scheduling people to projects. He gets a real rush at the thought of designing power trains for large trucks; however, given their current project scheduling problems, a large increase in business would only compound their problems. Somehow these conflicts in scheduling have to be resolved before any serious thought can be given to expanding into design of power transmissions for truck manufacturers.

Jones is thinking of the problems PT had in the last year. The MF project is the first to come to mind. The project was not terribly complex and did not require their best design engineers. Unfortunately, the scheduling software assigned one of the most creative and expensive engineers to the MF project. A similar situation, but reversed, happened on the Deer project. This project involved a big customer and new hydrostatic technology for small tractors. In this project the scheduling software assigned engineers who were not familiar with small tractor transmissions. Somehow, thinks Jones, the right people need to be scheduled to the right projects. Upon reflection, this problem with scheduling has been increasing since PT went to multiproject scheduling. Maybe a project office is needed to keep on top of these problems.

A meeting with the information technology team and software vendors was positive but not very helpful because these people are not really into detailed scheduling problems. The vendors provided all sorts of evidence suggesting the heuristics used—least slack, shortest duration, and identification number—are absolutely efficient in scheduling people and minimizing project delays. One project software vendor, Lauren, kept saying their software would allow PT to customize the scheduling of projects and people to almost any variation selected. Lauren repeated over and over, "If the standard heuristics do not meet your requirements, create your own heuristics that do." Lauren even volunteered to assist in setting up the system. But she is not willing to spend time on the problem until PT can describe to her exactly what criteria will be used (and their sequence) to select and schedule people to projects.

WHAT NEXT?

Potential expansion into the truck power train business is not feasible until the confusion in project scheduling is solved or reduced significantly. Jones is ready to tackle this problem, but he is not sure where to start.

Appendix 8.1

The Critical-Chain Approach

In practice, project managers carefully manage slack on sensitive resource-limited projects. For example, some managers use an early start schedule and prohibit use of slack on any activity or work package to be used unless authorized by the project manager. Progress by percent complete and by remaining time are carefully monitored and reported to catch any activities that beat estimated completion times so that succeeding activities can start ahead of schedule—on critical and noncritical activities. Monitoring and encouraging of early completion of estimated times ensure that the time gained is used to start a succeeding activity earlier and time is not wasted. The intent is to save the slack as a time buffer to complete the project early or to cover delay problems that may creep up on critical activities or paths.

Eliyahu Goldratt, who championed the "theory of constraints" in his popular book *The Goal,* advocates an alternative approach to managing slack. He has coined the term "critical-chain" to recognize that the project network may be constrained by both resource and technical dependencies. Each type of constraint can create task dependencies, and in the case of resource constraints, new task dependencies can be created! Remember, the critical resource shifted the critical path. Visit Figure 8.4 again. The critical-chain (C-C) refers to the longest string of dependencies that exist on the project. Chain is used instead of path, since the latter tends to be associated with just technical dependencies not resource dependencies. Goldratt uses the C-C concept to develop strategies for accelerating the completion of projects. These strategies are based on his observations about time estimates of individual activities.

TIME ESTIMATES

Goldratt argues that there is a natural tendency for people to add safety (just-in-case) time to their estimations. It is believed that those who estimate activity times provide an estimate that has about an 80 to 90 percent chance of being completed on or before the estimated time. Hence, the median time (50/50 chance) is overestimated by approximately 30 to 40 percent. For example, a programmer may estimate that there is a 50/50 chance that he can complete an activity in five days. However, to ensure success and to protect against potential problems, he adds three days of safety time and reports that it will take eight days to complete the task. In this case the median (50/50) time is overestimated by approximately 60 percent. If this hidden contingency is pervasive across a project, then most activities in theory should be completed ahead of schedule. Remember, the programmer still has a 50/50 chance of completing the assignment within five days or less.

This situation raises an interesting paradox. Why, if there is a tendency to overestimate activity durations, do so many projects come in behind schedule? Goldratt offers several explanations:

- *Parkinson's law:* Work fills the time available. Why hustle to complete a task today when it isn't due until tomorrow?
- *Self-protection:* Participants fail to report early finishes out of fear that management will adjust their future standards and demand more next time.
- *Dropped baton:* Early finishes may not lead to the start of the next activity because people assigned to perform the next activity are not ready to start work early. The time gained is therefore lost.
- *Excessive multitasking:* This phenomenon adds time to the completion of tasks.

- *Resource bottlenecks:* Delays are caused by limited availability of critical resources.
- *Student syndrome:* There is a tendency to delay the start of tasks until you absolutely have to.

The dropped baton and student syndrome explanations deserve further discussion. Goldratt uses the metaphor of a relay race to describe the impact of poor coordination between resources. Just like a runner's time is lost if the next runner is not ready to receive the baton, so is the time gained from completing a task early if the next group of people are not ready to receive the project work.

Goldratt also asserts that just as students delay writing a term paper until the last minute, workers delay starting tasks when they perceive that they have more than enough time to complete the task. The problem with delaying the start of a task is that obstacles are often not detected until the task is underway. By postponing the start of the task, the opportunity to cope with these obstacles and complete the task on time is compromised.

CRITICAL-CHAIN IN ACTION

Goldratt's solution to reducing project time overruns is to insist on people using the "true 50/50" activity time estimates (rather than estimates which have an 80 to 90 percent chance of being completed before the estimated time); the 50/50 estimates result in a project duration about one-half the low risk of 80 to 90 percent estimates. Using 50/50 estimates will discourage Parkinson's law, the student syndrome, and self protection from coming into play and in turn should increase productivity on individual tasks. Similarly, the compressed time schedule reduces the likelihood of the dropped baton effect.

Goldratt recommends inserting time buffers into the schedule to act as "shock absorbers" to protect the project completion date against task durations taking longer than the 50/50 estimate. The rationale is that by using 50/50 estimates you are in essence taking out all of the "safety" in individual tasks. He also recommends using portions of this collective safety strategically by inserting time buffers where potential problems are likely to occur. First, since all activities along the critical chain have inherent *uncertainty* that is difficult to predict, project duration is uncertain. Therefore, a project time buffer is added to the expected *project duration*—say 50 percent of the aggregate of the hidden contingencies of activity durations. Second, *feeder* buffers are added to the network where noncritical paths merge with the critical chain. These buffers serve to protect the critical path from being delayed. Third, *resource* time buffers are inserted where scarce resources are needed for an activity. Resource time buffers come in at least two forms. One form is a time buffer attached to a critical resource to ensure that the resource is on call and available when needed. This preserves the relay race. The second form of time buffer is added to activities preceding the work of a scarce resource. This kind of buffer protects against resource bottlenecks by increasing the likelihood that the preceding activity will be completed when the resource is available. All buffers reduce the risk of the project duration being late and increase the chance of early project completion.

CRITICAL-CHAIN VERSUS TRADITIONAL SCHEDULING APPROACH

To illustrate how C-C affects scheduling lets compare it with the traditional approach to project scheduling. We will first resolve resource problems the way described in Chapter 8 and then the C-C method. Figure A8.1A shows the *planned* Air Control project network without any concern for resources. That is, activities are assumed to be independent and resources will be made available and/or are interchangeable. Figure A8.1B depicts the bar chart for the project. The green bars represent the durations of critical activities; the clear bars represent the durations of noncritical activities; the light gray

FIGURE A8.1A Air Control Project: Time Plan without Resources

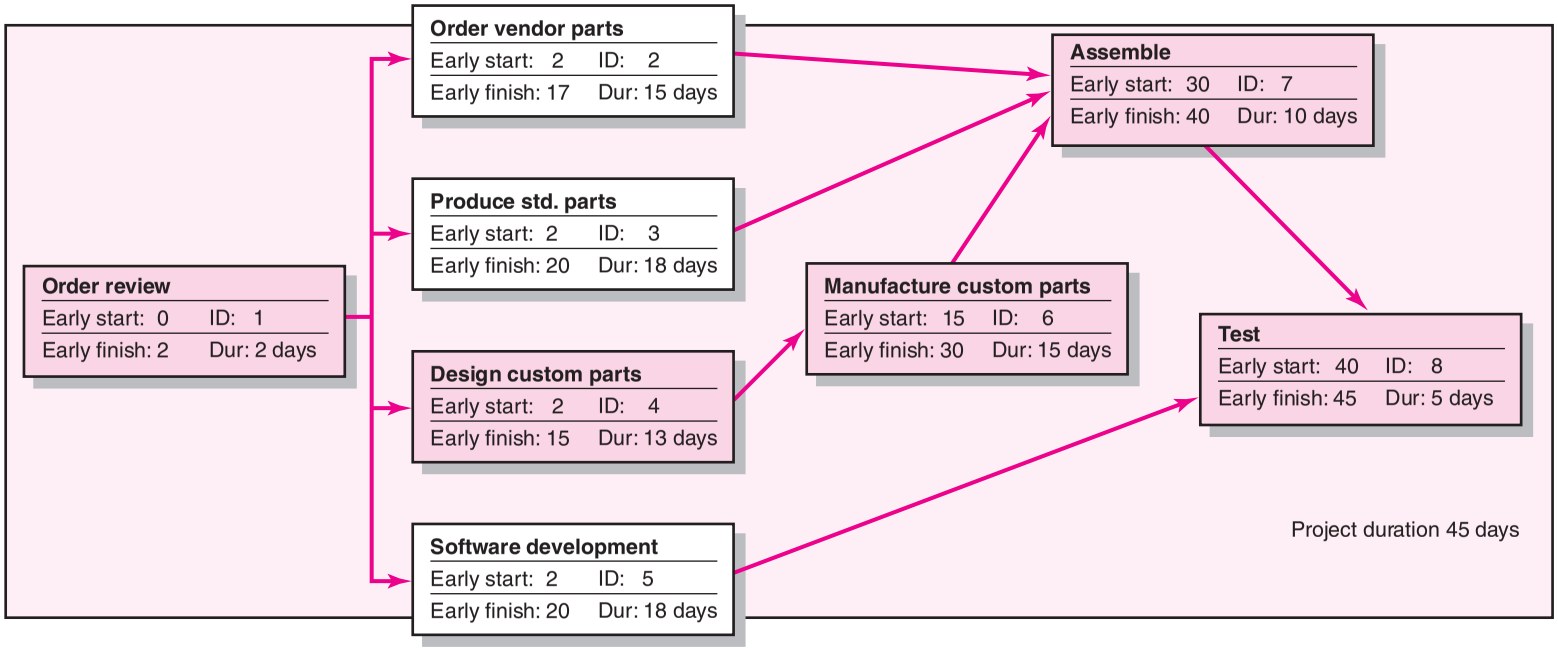
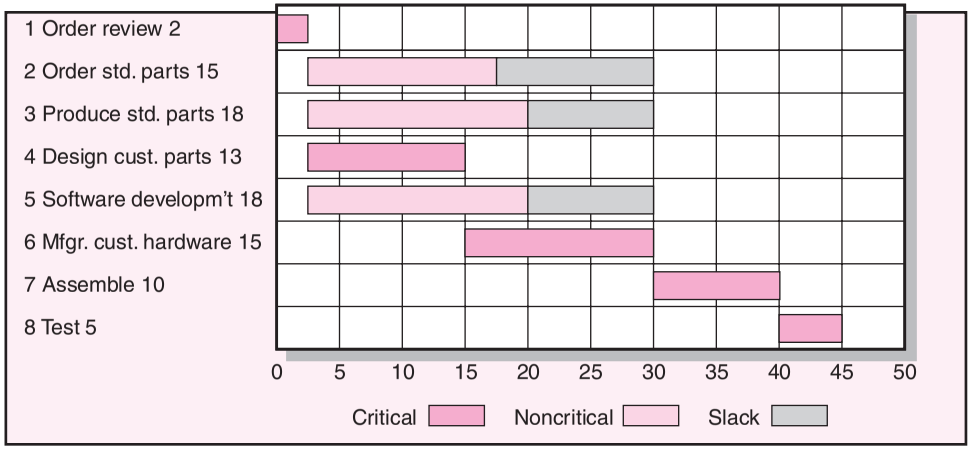


FIGURE A8.1B Air Control Project: Time Plan without Resources



bars represent slack. Note that the duration is 45 days and the critical path is represented by activities 1, 4, 6, 7, and 8.

Parallel activities hold potential for resource conflicts. This is the case in this project. Ryan is the resource for activities 3 and 6. If you insert Ryan in the bar chart in Figure A8.1B for activities 3 and 6, you can see activity 3 overlaps activity 6 by five days—an impossible situation. Because Ryan cannot work two activities simultaneously and no other person can take his place, a resource dependency exists. The result is that two activities (3 and 6) that were assumed to be independent now become dependent. Something has to give! Figure A8.2A shows the Air Control project network with the resources included. A pseudo-dashed arrow has been added to the network to indicate the resource dependency. The bar chart in Figure A8.2B reflects the revised schedule resolving the overallocation of Ryan. Given the new schedule, slack for some activities has changed. More importantly, the critical path has changed. It is now 1, 3, 6, 7, 8. The resource schedule shows the new project duration to be 50 days rather than 45 days.

Now let's apply the C-C approach to the Air Control project. Figure A8.3 details many of the changes. First, notice that task estimates now represent approximations of the 50/50 rule. Second, observe that not all of the activities on the critical-chain are technically linked. Manufacturing custom parts is included because of previously defined resource dependency. Third, a project time buffer is added at the end of schedule. Finally, feeder buffers are inserted at each point where a noncritical activity merges with the critical chain.

The impact the C-C approach has on the project schedule can best be seen in the Gantt chart presented in Figure A8.4. Notice first the late start times for each of the three noncritical activities. For example, under the critical path method, order vendor parts and software development would be scheduled to begin immediately after the order review. Instead they are scheduled later in the project. Three-day feeder buffers have been added to each of these activities to absorb any delays that might occur in these activities. Finally, instead of taking 50 days the project is now estimated to take only 27 days with a 10-day project buffer!

This example provides an opportunity for explaining the differences between buffers and slack. Slack is spare time inherent in the schedule of noncritical activities and can be determined by differences between the early start and late start of a specific activity. Buffers, on the other hand, are dedicated time blocks reserved to cover most likely contingencies and are monitored closely so, if they are not needed, subsequent activities can proceed on schedule. Buffers are needed in part because the estimates are based on 50/50 approximations, and therefore roughly half of the activities will take longer than planned. To protect against these extended activity durations, buffers are inserted to minimize the impact on the schedule. Buffers are not part of the project schedule and are used only when sound management dictates it.

While not depicted in the figures, an example of a resource buffer would be to add six days to Ryan's schedule (remember he is the critical resource that caused the schedule to be extended). This would ensure that he could continue to work on the project beyond the 18th day in case either produce standard parts and/or manufacture custom hardware takes longer than planned. Progress on these two tasks would be monitored closely, and his schedule would be adjusted accordingly.

C-C AND MULTITASKING

Buffers do not address the insidious effects of pervasive multitasking, especially in a multiproject environment where workers are juggling different project assignments. Goldratt has three recommendations that will help to reduce the impact of multitasking:

1. Reduce the number of projects so people are not assigned to as many projects concurrently.

FIGURE A8.2A Air Control Project: Schedule with Resources Limited

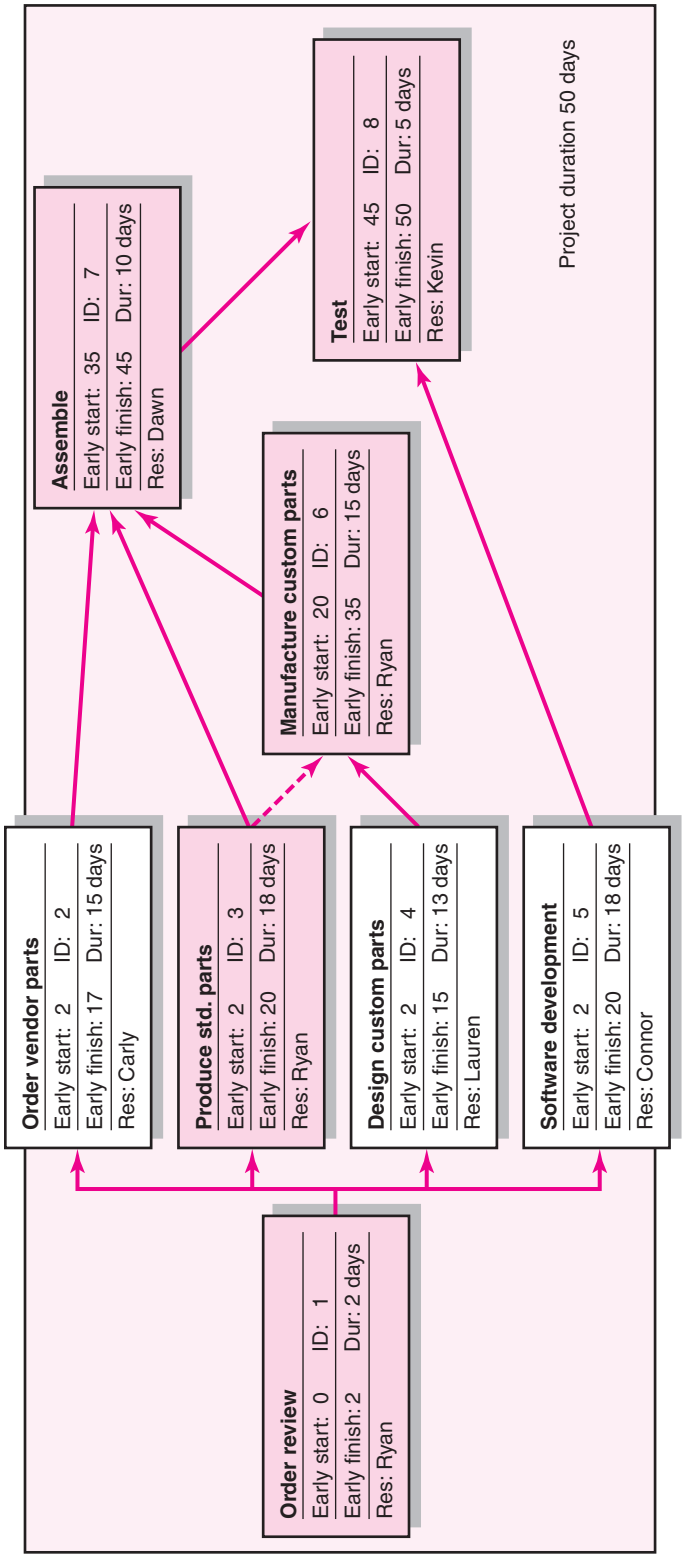


FIGURE A8.2B Air Control Project: Schedule with Resources Limited

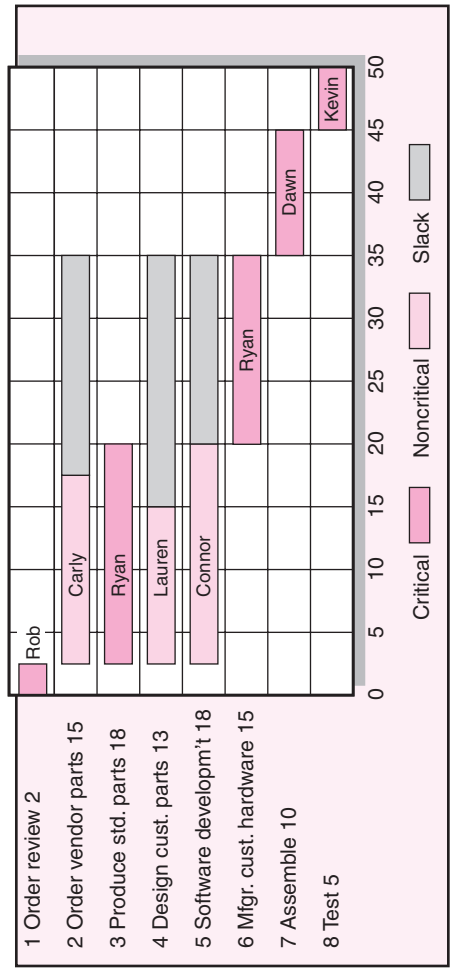


FIGURE A8.3 Air Control Project: Critical-Chain Network

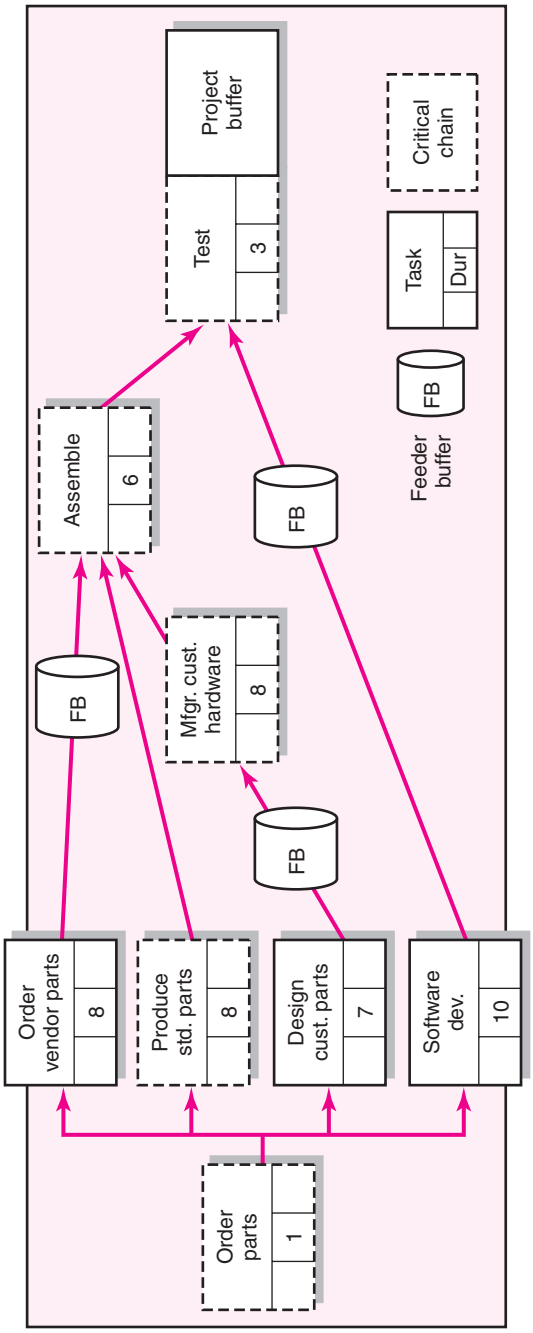
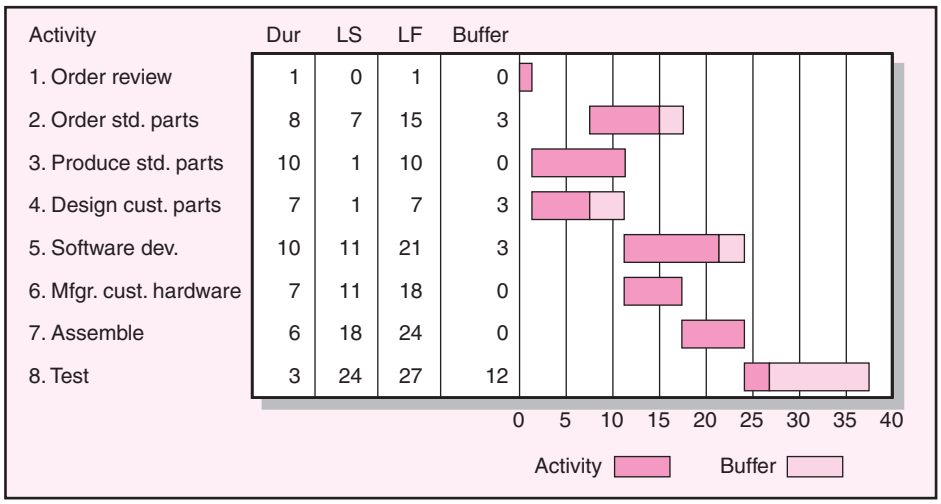


FIGURE A8.4
Air Control Project Gantt Chart: Critical-Chain Network



2. Control start dates of projects to accommodate resource shortages. Don't start projects until sufficient resources are available to work full time on the project.
3. Contract (lock in) for resources *before* the project begins.

MONITORING PROJECT PERFORMANCE

The C-C method uses buffers to monitor project time performance. Remember that as shown in Figure A8.3 a project buffer is used to insulate the project against delays along the critical-chain. For monitoring purposes, this buffer is typically divided into three zones—OK, Watch, and Act, respectively. As the buffer begins to decrease and moves into the second zone, alarms are set off to seek corrective action. (For additional sources see endnote 2.) To be truly effective, buffer management requires comparing buffer usage with actual progress on the project. For example, if the project is 75 percent complete and you have only used 50 percent of the project buffer, then the project is in pretty good shape. Conversely, if the project is only 25 percent complete and 50 percent of the buffer has already been used, you are in trouble and corrective action is needed. A method for estimating percentage complete is described in Chapter 13.

THE C-C METHOD TODAY

C-C has generated considerable debate within the project management community. While sound in theory, support at this time is limited but growing. For example, Harris Semiconductor was able to build a new automated wafer fabrication facility within 13 months using C-C methods when the industry standard for such a facility is 26–36 months. The Israeli aircraft industry has used C-C techniques to reduce average maintenance work on aircraft from two months to two weeks. The U.S. Air Force and Navy as well as Boeing, Lucent Technologies, Intel, GM, and 3M are applying critical-chain principles to their multi-project environments.

C-C is not without critics. First, C-C does not address the biggest cause of project delays, which is an ill-defined and unstable project scope. Second, some critics challenge Goldratt's assumptions about human behavior. They question the tendency of experts to pad estimates and that employees act deliberately against the organization for their own interest and benefit. They also object to the insinuation that trained professionals would exhibit the student syndrome habits. Right or wrong, for some there is an implied element of distrust and lack of faith in the employee's expertise in the C-C approach. The

harshest critics see C-C as a highly manipulative system in which people are set up for failure with the hope that this added pressure will increase productivity.

The key to implementing theory into practice is the culture of the organization. If the organization honors noble efforts that fail to meet estimates as it does efforts that do meet estimates, then greater acceptance will occur. Conversely, if management treats honest failure differently from success, then resistance will be high. Organizations adopting the C-C approach have to invest significant energy to obtaining "buy-in" on the part of all participants to its core principles and allaying the fears that this system may generate.

SUMMARY

Regardless of where one stands in the debate, the C-C approach deserves credit for bringing resource dependency to the forefront, highlighting the modern ills of multitasking, and forcing us to rethink conventional methods of project scheduling.

EXERCISE

1. Check out the Goldratt Institute's homepage at *http://www.goldratt.com* for current information on the application of critical-chain techniques to project management.
2. Apply critical-chain scheduling principles to the Print Software, Inc., project presented in Chapter 6 on page 183. Revise the estimated time durations by 50 percent except round up the odd time durations (i.e., 3 becomes 4). Draw a C-C network diagram similar to the one contained in Figure A8.3 for the Print Software project as well as a Gantt chart similar to Figure A8.4. How would these diagrams differ from the ones generated using the traditional scheduling technique?

APPENDIX REFERENCES

Goldratt, *Critical Chain* (Great Barrington, MA: North River Press, 1997).

Herroelen, W., R. Leus, and E. Demeulemeester, "Critical Chain Project Scheduling: Do Not Oversimplify," *Project Management Journal,* Vol. 33 (4), 2002, pp. 48–60.

Leach, L. P., "Critical Chain Project Management," *Proceedings of 29th Annual Project Management Institute, 1998, Seminars and Symposium,* (Newtown, PA: Project Management Institute, 1998), pp. 1239–44.

Levine, H. A., "Shared Contingency: Exploring the Critical Chain," *PM Network,* October, 1999, pp. 35–38.

Newbold, R. C., *Project Management in the Fast Lane: Applying the Theory of Constraints,* (Baton Roca, FL: St. Lucie Press, 1998).

Noreen, E., D. Smith, and J. Mackey, *The Theory of Constraints and it's Implication for Management Accounting* (Barrington, MA: North River Press, 1995).

Raz, T., R. Barnes, and D. Dvir, "A Critical Look at Critical Chain Project Management," *Project Management Journal,* December 2003, pp. 24–32.

Sood, S., "Taming Uncertainty: Critical-Chain Buffer Management Helps Minimize Risk in the Project Equation," *PM Network,* March 2003, pp. 57–59.

Zalmanson, E., "Readers Feedback," *PM Network,* Vol. 15 (1), 2001, p. 4.