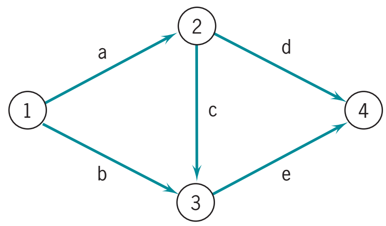
Solved Problem

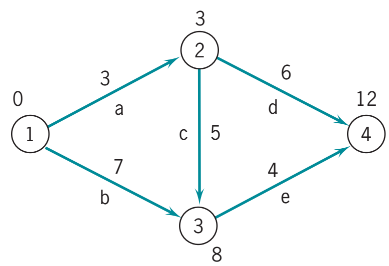
Given the following network (time in days):



Activity	*Crash Time, Cost*	*Normal Time, Cost*	*Partial Crashing?*
a	3, $60	3, $60	No
b	6, 80	7, 30	Yes
c	2, 90	5, 50	No
d	5, 50	6, 30	No
e	2, 100	4, 40	Yes

Find the lowest cost to complete the project in 10 days.

Answer:
Current time and cost: 12 days and $210



Since the critical path is **a-c-e**, we only initially need consider these three activities:

a: cannot be crashed

c: can cut three days at an extra cost of $40 but, due to **b**, only results in project completion by day 11. To reach 10 days, cut **b** by one day, total extra cost $90.

e: can cut **e** by two days for an extra cost of $60 and results in project completion by day 10.

Thus, cut **e** two days at a cost of $60.

construction projects, as well as other types of projects when the early "build" or "carry out" steps are fairly routine and well understood. It is a partial use of the basic concept in phase-gate project management and is dependent on effective feed-back and feed-forward communication.

9.2 THE RESOURCE ALLOCATION PROBLEM

A shortcoming of the scheduling procedures covered in the previous chapter is that they do not address the issues of resource utilization and availability. The focus is on time rather than physical resources. Also, in the discussion that follows it will not be sufficient to refer to resource usage simply as "costs." Instead, we must refer to individual types of labor, specific facilities, kinds of materials, individual pieces of equipment, and other discrete inputs that are relevant to an individual project but are limited in availability. Last, we must not forget that time itself is always a critical resource in project management, one that is unique because it can neither be inventoried nor renewed. **One cannot save time—one can only spend more or less of it.**

The relationship between progress, time, and resource availability/usage is the major focus of this chapter. Schedules should be evaluated not merely in terms of meeting project milestones, but also in terms of the timing and use of scarce resources. A fundamental measure of the PM's success in project management is the skill with which the trade-offs among performance, time, and cost are managed. It is a continuous process of cost-benefit analysis: "I can shorten this project by a day at a cost of $400. Should I do it?" "If I buy 300 more hours of engineering time, I may be able to improve performance by 2 or 3 percent. Should I do it?" Of course all such estimates are uncertain. What are the risks and how should I deal with them?

Occasionally it is possible that some additional (useful) resources can be added at little or no cost to a project during a crisis period. At other times, some resources in abundant supply may be traded for scarce ones. Most of the time, however, these trades entail additional costs to the organization, so a primary responsibility for the PM is to make do with what is available.

The extreme points of the relationship between time use and resource use are these:

- ***Time Limited:*** The project must be finished by a certain time, using as few resources as possible. But it is time, not resource usage, that is critical.
- ***Resource Limited:*** The project must be finished as soon as possible, but without exceeding some specific level of resource usage or some general resource constraint.

Project Management in Practice

Benefit/Cost Analysis Saves Chicago's Deep Tunnel Project

A creative approach to benefit/cost analysis comes from the Chicago Deep Tunnel Project, conceived in the 1970s and now scheduled for completion in 2015 at a cost of $3–4 billion. The project consists of 110 miles of 300-foot underground tunnels, plus reservoirs and pollution-control systems for holding both sewage and excess rainwater. However, escalating costs over the years have reduced the benefit-cost ratio to less than 3 and the federal government was forced to reconsider their spending on it. With the threat of a loss in funding, the project was redesigned to improve the benefits while reducing the costs by $100 million, thereby boosting the benefit-cost ratio to above four, thus meeting federal standards.

Source: Project Management Institute. "Digging Deep," *PM Network,* May 2006, p. 1.

The points between these two extremes represent time/resource-use trade-offs. As in Figure 9-2, they specify the times achievable at various resource levels. Equivalently, they specify the resources associated with various completion times. Clearly, the range of time or resource variability is limited.

Occasionally, both time and resources may be limited, but in this case the specifications cannot also be fixed. If all three variables—time, cost, specifications—are fixed, the system is "overdetermined." The PM has lost all flexibility to perform the trade-offs that are so necessary to the successful completion of projects. Of course, it is possible that all three variables might be fixed at levels that allowed the PM plenty of maneuvering room, but this is most unlikely. Far more likely, our project manager acquaintances tell us, is the case in which senior management assigns budgets, schedules, and specifications without regard for the uncertainties of reality. It is the PM's responsibility, possibly with help from the project's champion, to warn senior management of the impropriety of such restrictions in spite of the chance that a senior manager might respond with "I'll get someone who can . . . !" If our advice seems a bit strong, we refer you to Jim McCarthy's Rule #25 (McCarthy, 1995, pp. 88–89, or see Mantel et al., 2008, pp. 114–115): "Don't accept task dates, features, and resource dictates from managers unfamiliar with the task."

On occasion, it may be that one or more tasks in a project are *system-constrained*. A system-constrained task requires a fixed amount of time and known quantities of resources. Some industrial processes—heat treating, for instance—are system-constrained. The material must "cook" for a specified time to achieve the desired effect. More or less "cooking" will not help. When dealing with a system-constrained task or project, no trade-offs are possible. The only matter of interest in these cases is to make sure that the required resources are available when needed.

In the following sections, we discuss approaches for understanding and using these relationships in various project situations.

9.3 RESOURCE LOADING

Resource loading describes the amounts of individual resources an existing schedule requires during specific time periods. Therefore, it is irrelevant whether we are considering a single work unit or several projects; the loads (requirements) of each resource type are simply listed as a function of time period. Resource loading gives a general understanding of the demands a project or set of projects will make on a firm's resources. It is an excellent guide for early, rough project planning. Obviously, it is also a first step in attempting to reduce excessive demands on certain resources, regardless of the specific technique used to reduce the demands. Again, we caution the PM to recognize that the use of resources on a project is often nonlinear. Much of the project management software does not recognize this fact (Gilyutin, 1993).

If resources of a project are increased by X percent, the output of the project usually does not increase by X percent, and the time required for the project does not decrease by X percent. The output and time may not change at all, or may change by an amount seemingly not related to X. An increase of 20 percent in the number of notes played does not necessarily improve the quality of the music. Any time the resource base of a project is altered from standard practice, the risk that the project may not be successful is changed, often increased.

Given an action plan, deriving a resource-loading document is not difficult. Figure 6-3 (Chapter 6, Section 6-3) shows part of an action plan for a "Career Day" at a college. The part of the plan shown lists the personnel resources needed for each activity. (The hours required are included in the plan, but were not printed in Figure 6-3.) Utilizing data in the plan, MSP

Career Day Project Resource Usage Calendar																
				May				June				July				
ID	Resource Name	Work	25	2	9	16	23	30	6	13	20	27	4	11	18	25
1	**Secretary**	**1,020 hrs**	24h	40h	40h	40h	88h	120h	102h	40h	40h	40h	40h	40h	40h	40h
	Print forms	240 hrs														
	Gather college particulars	160 hrs	24h	40h	40h	40h	16h									
	Print programs	240 hrs					24h	40h	40h	40h	40h	40h	16h			
	Advertise in college paper	200 hrs					24h	40h	36h	0h	0h	0h	24h	40h	36h	
	Organize posters	180 hrs					24h	40h	26h	0h	0h	0h	0h	0h	4h	40h
2	**Program Manager**	**1,440 hrs**	40h	40h	40h	16h	24h	40h	40h	40h	16h					
	Contact organizations	600 hrs	16h													
	Select guest speaker	560 hrs														
	Organize food	120 hrs	24h	40h	40h	16h										
	Contact faculty	60 hrs					24h	36h								
	Arrange facility for event	100 hrs						4h	40h	40h	16h					
3	**Office Manager**	**180 hrs**	24h	40h	40h	40h	16h				20h					
	Collect display information	160 hrs	24h	40h	40h	40h	16h									
	Transport materials	20 hrs									20h					
4	**Graduate Assistant**	**1,140 hrs**	24h	40h	40h	40h	64h	80h	80h	56h	40h	40h	16h			
	Print participants' certificates	320 hrs														
	Organize refreshments	280 hrs	24h	40h	40h	40h	40h	40h	40h	16h						
	Send invitations	80 hrs														
	Organize gift certificates	220 hrs														
	Arrange banner	200 hrs					24h	40h	40h	40h	40h	16h				
	Class announcements	40 hrs										24h	16h			
5	**Director**	**400 hrs**	24h	40h	40h	40h	40h	40h	40h	40h	40h	40h	16h			
	Organize liquor	400 hrs	24h	40h	40h	40h	40h	40h	40h	40h	40h	40h	16h			

Figure 9-3 Resource usage calendar for the Career Day Project.

generated Figure 9-3, the resource usage calendar. Each of the human resources used in the project is listed, followed by the name of the activities in which the resource is used. The total hours of work for each resource called for by the action plan are shown together with the amount planned for each activity. The schedule for resource loading is derived and the loading is then shown for each resource for each week (or day or month) of the project. It should be clear that if the information in this calendar were entered into a Crystal Ball®/Excel® spreadsheet along with estimates of the variability of resource times, the resource loading for any or all of these resources could be estimated by simulation.

An examination of Figure 9-3 shows that the secretary is overloaded during late May and early June. Assuming that there is only one secretary, during the week of May 30 he or she must work 17+ hours per day of a 7-day week (or 24 hours per 5-day week). This is apt to try the patience of the most determined and loyal employee. Graduate assistants are certainly considered slaves by their faculty masters, but are usually indentured for only 20 hours per week of servitude. Unless there are four GAs, the project will have problems. It is the job of the PM to deal with these problems, either by adding people or by changing the schedule in such a way that the demand for resources does not exceed resource capacities.

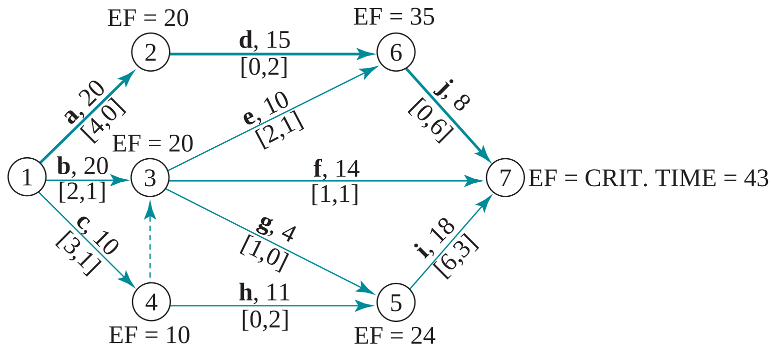


Figure 9-4 The AOA network of Table 8-2.

Because the project action plan is the source of information on activity precedences, durations, and resources requirements, it is the primary input for both the project schedule and its budget. The action plan links the schedule directly to specific demands for resources. Thus, the AOA network technique can be modified to generate time-phased resource requirements. A Gantt chart could be adapted, but the AOA diagram, particularly if modified to illustrate slacks as in Figure 9-1, will be helpful in the analysis used for resource leveling. Let us illustrate with the AON network used as an example in the previous chapter, but converted to an AOA diagram. The AOA network (from Table 8-2) is illustrated in Figure 9-4, and resource usage is illustrated for two hypothetical resources, A and B, on the arcs. The expected activity time is shown above the arc, and resource usage is shown in brackets just below the arc, with the use of A shown first and B second—e.g., [5, 3] would mean that five units of A and three units of B would be used on the activity represented by the arc. Figure 9-5 shows the "calendarized" AOA diagram, similar to the familiar Gantt chart. Resource demands can now be summed by time period across all activities.

The loading diagram for resource A is illustrated in Figure 9-6a, and that for resource B in Figure 9-6b. The loads are erratic and vary substantially over the duration of the project. Resource A, used in tasks **a**, **b**, and **c**, has a high initial demand that drops through the

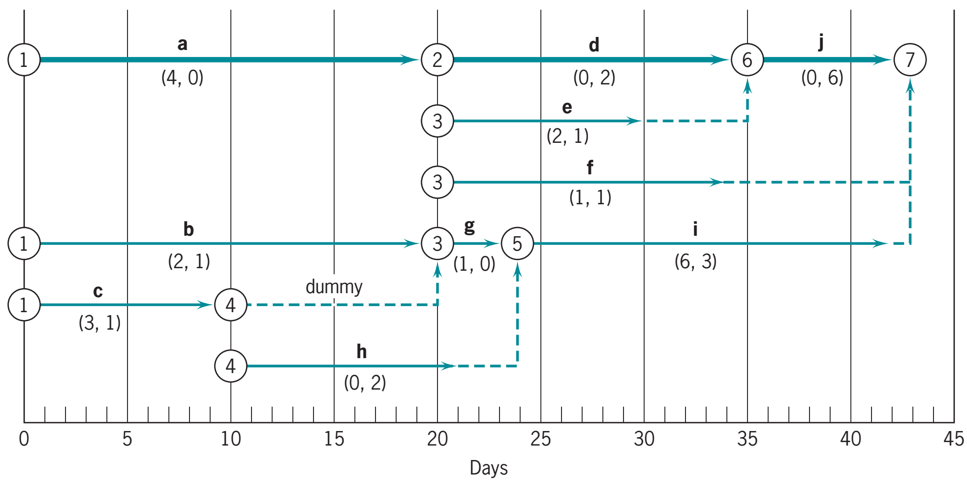


Figure 9-5 Modified AOA diagram showing activity slack and resource usage (from Figure 9-4).

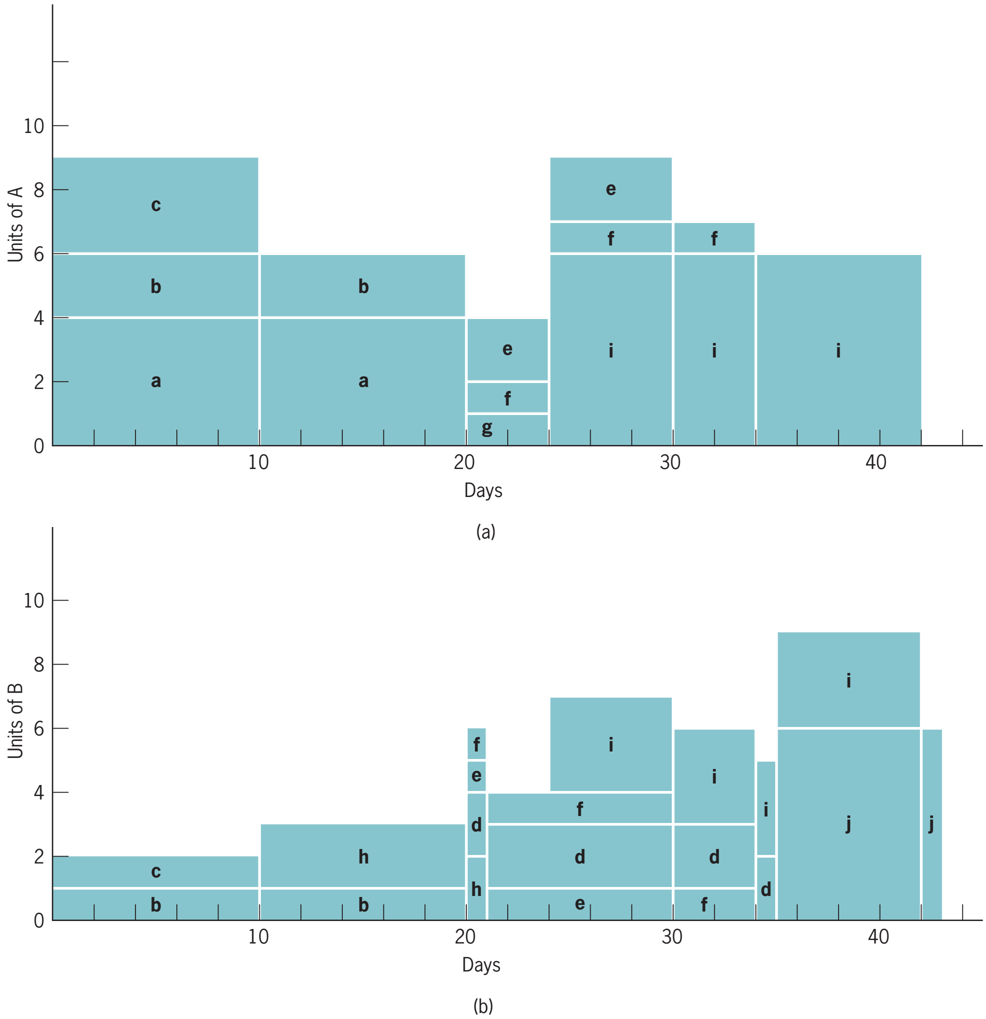


Figure 9-6 (a) Load diagram for resource A. (b) Load diagram for resource B.

middle of the project and then climbs again. Resource B, on the other hand, has low initial use but increases as the project develops. The PM must be aware of the ebbs and flows of usage for each input resource throughout the life of the project. It is the PM's responsibility to ensure that the required resources, in the required amounts, are available when and where they are needed. In the next three sections, we will discuss how to meet this responsibility.

9.4 RESOURCE LEVELING

In the preceding example, we noted that the project began with the heavy use of resource A, used smaller amounts during the middle of the project, and then continued with rising usage during the project's latter stages. Usage of B started low and rose throughout the project's life. Large fluctuations in the required loads for various resources are a normal occurrence—and are undesirable from the PM's point of view. Resource leveling aims to minimize the period-by-period variations in resource loading *by shifting tasks within their slack allowances*. The purpose is to create a smoother distribution of resource usage.

There are several advantages to smoother resource usage. First, much less hands-on management is required if the use of a given resource is nearly constant over its period of use. The PM can arrange to have the resource available when needed, can have the supplier furnish constant amounts, and can arrange for a backup supplier if advisable. Moreover, the PM can do this with little error. Second, if resource usage is level, the PM may be able to use a "just-in-time" inventory policy without much worry that the quantity delivered will be wrong. If the resource being leveled is people, leveling improves morale and results in fewer problems in the personnel and payroll offices because of increasing and decreasing labor levels.

Not only are there managerial implications to resource leveling, there are also important cost implications. When resources are leveled, the associated costs also tend to be leveled. If resource use increases as time goes by, and if resources are shifted closer to the present by leveling, costs will be shifted in the same way. The opposite is true, of course, if resource usage is shifted to the future. Perhaps most important from a cost perspective is leveling employment throughout a project or task. For most organizations, the costs of hiring and layoff are quite significant. It is often less expensive to level labor requirements in order to avoid hiring and layoff, even if it means some extra wages will be paid. In any case, the PM must be aware of the cash flows associated with the project and of the means of shifting them in ways that are useful to the parent firm.

The basic procedure for resource leveling is straightforward. For example, consider the simple AOA network shown in Figure 9-7a. The activity time is shown above the arc, and resource usage (one resource, workers) is in brackets below the arc. Activities **a, b**, and **c** follow event 1, and all must precede event 4. Activity **a** requires two workers and takes two days, **b** requires two workers and takes 3 days, and **c** needs 4 workers and 5 days. (We addressed the problem of trade-offs between labor and activity time in the first section of this chapter.) If all these tasks are begun on their early start dates, the resource loading diagram appears as shown in Figure 9-7b, steps of decreasing labor demand varying from eight workers to four workers. If, however, task **b** is delayed for 2 days, the full length of its slack in this particular case, the resource loading diagram is smoothed, as shown in Figure 9-7c. The same result would have occurred if **b** were started as early as possible and task **a** were delayed until day 3.

Resource leveling is a procedure that can be used for almost all projects, whether or not resources are constrained. If the network is not too large and there are only a few resources, the leveling process can be done manually. For larger networks and multiple resources, resource leveling becomes extremely complex, far beyond the power of manual solutions. Fortunately, a number of computer programs can handle most leveling problems efficiently (discussed in Chapter 10).

Reconsider the load diagrams of Figures 9-6a and b. Assume it is desired to smooth the loading of resource B, which is particularly jagged. Both activities **e** and **f** can be delayed (**e** has 5 days of slack and **f** has 9). If we delay both for one day, we remove the peak on day 20 without increasing any of the other peaks (see Figure 9-8b). If we do this, however, it also alters the use of resource A and deepens the "valley" on day 20 (see Figure 9-8a). If we further delay **f** another 7 days in order to level the use of A toward the end of the project, we would deepen the valley between days 20 and 24, and the resultant use of A would be as shown by the dotted lines on Figure 9-8a. Activity **f** would begin on day 28 (and would become critical). The effect on the usage of B is easy to see (Figure 9-8b). The change would lower usage by one unit beginning on day 21 (remember that we have already

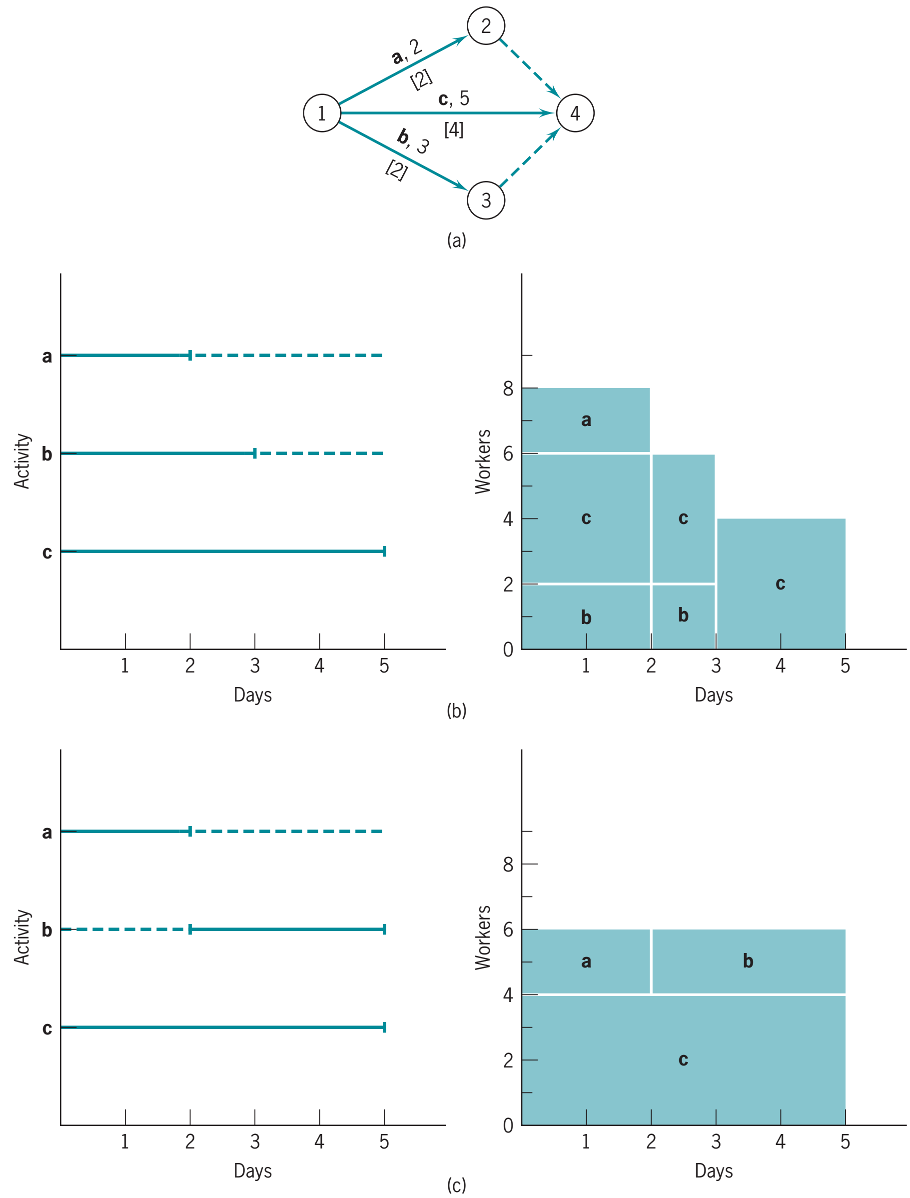


Figure 9-7 (a) The network. (b) Before resource leveling. (c) After resource leveling.

delayed **f** one day), and increase usage by one unit beginning on day 35, continuing to the end of the project. This action increases peak use of B from nine to ten units.

It is important to emphasize that if the network under consideration is more complex and the number of resources to be leveled is realistically large, a manual leveling process is out of the question. Computer-aided leveling is not only mandatory, it is also helpful because it allows the PM to experiment with various patterns of resource usage through simulation. In the next section, we raise the most general problem of minimizing resource usage while still achieving various completion dates—or the inverse problem, minimizing completion times while operating with specified limits on resources.

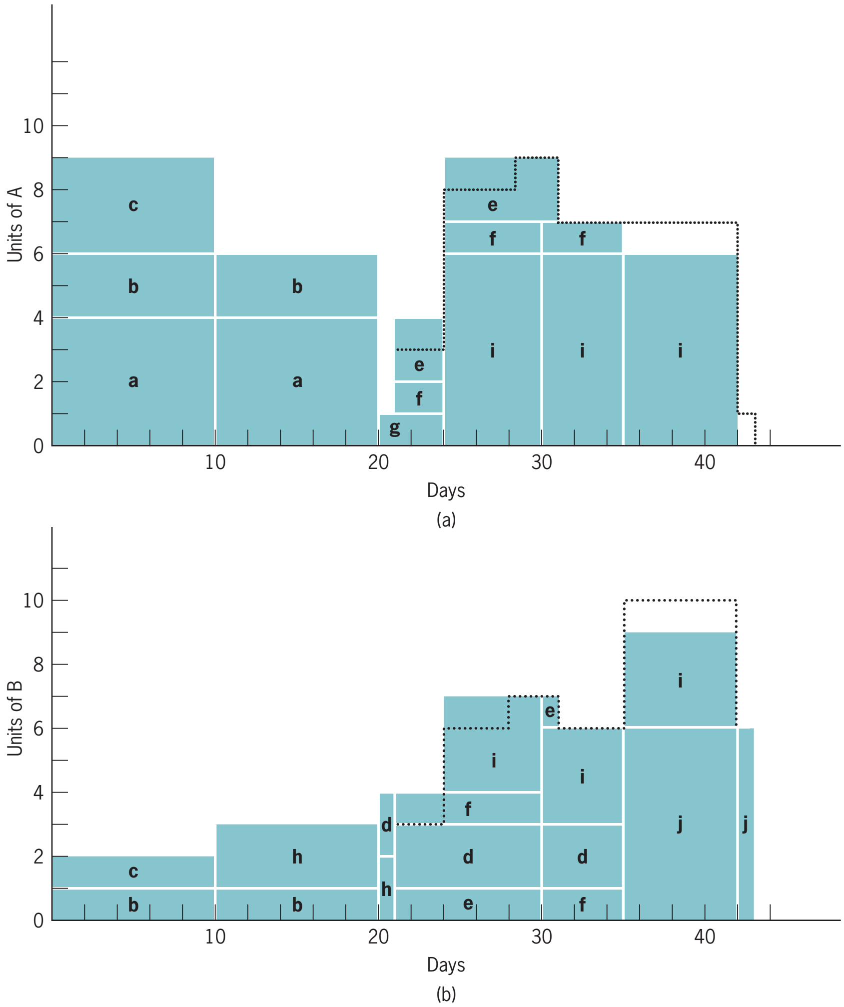


Figure 9-8 (a): Load diagram for resource A with activities **e** and **f** delayed by one day each. (b) Load diagram for resource B with activities **e** and **f** delayed by one day each.

Resource Loading/Leveling and Uncertainty

Figure 9-9 is a resource loading chart for a software engineering group in a large company, constructed by importing MSP resource loading information into an Excel® spreadsheet and then displaying it graphically. There are 21 engineers in the group, nominally scheduled to work 40 hours a week, resulting in a weekly capacity of

$$21 \times 40 = 840 \text{ labor-hours each week.}$$

The graph covers February through September, a period of 34 weeks. Thus, the total engineering capacity for the period shown is

$$34 \times 840 = 28{,}560 \text{ labor-hours.}$$

As shown, the total labor-hours required for the period is 28,282, so we see that there is excess capacity, a nice situation normally. However, there are two problems. As is clear

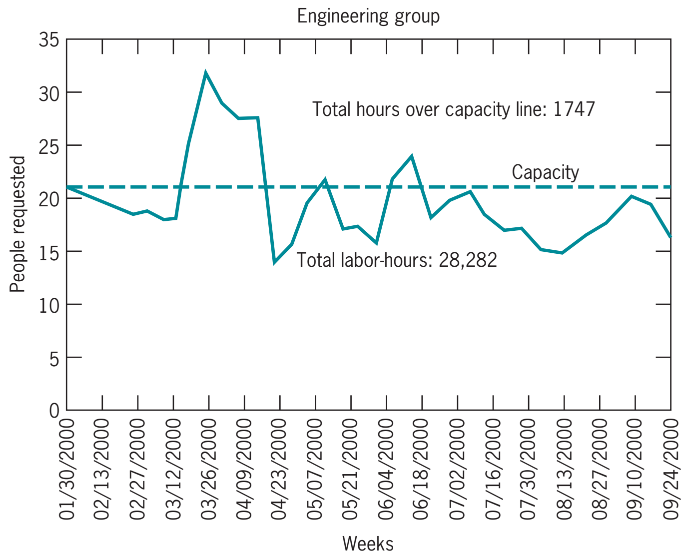


Figure 9-9 Thirty-four-week resource loading chart for a software engineering group.

in the loading chart, the demand for engineering labor is not evenly distributed throughout the period, hitting a major peak in the late March–early April time frame and then a few more times later in the period. This is counterbalanced by weeks throughout the time period where less-than-full capacity is required; however, this is not helpful since the engineers are employed for 40-hour weeks so the undercapacity times are wasted.

There are some alternatives used to address these kinds of situations. First, we can try to level the demand, moving some of it forward and some back, depending on our flexibility in this production environment. Second, we can try to alter the supply of engineering hours, asking the engineers to trade off time between periods of overcapacity and periods of undercapacity. We might expend additional resources, bringing in contract engineers to handle the overload period, or subcontract the workload, but such suggestions would almost certainly be rejected by senior management because of security worries by our clients. Perhaps it would be cheaper to let the work be delayed a few weeks and try to catch up later. We will identify other ways of resolving this element of uneven demand a bit later.

But there is another problem with this situation, which is that we try not to schedule a scarce resource for more than 85–90 percent capacity. The reason is due to disruptions, emergencies, maintenance requirements, personnel issues, and simple inefficiency of any resource that is scheduled for full production over an extended time period. Let us consider the case of the engineers, in particular. Over this 34-week period that includes the summer, there will probably be most of the scheduled 2-week vacations (if not longer). If 15 engineers are scheduled for vacations during this period, that will remove 15×2 weeks $\times 40 = 1200$ labor-hours from the capacity. In addition, there are three national holidays during this period: Memorial Day, the Fourth of July, and Labor Day, resulting in a further loss of $21 \times 3 \times 8$ hours $= 504$ hours. These two scheduled sets of events have now reduced our capacity to $28{,}560 - 1200 - 504 = 26{,}856$ labor-hours, 5 percent less than the demand over the period.

What about unscheduled events and disruptions? Illnesses will surely occur in this long time frame. Furthermore, will the facilities, equipment, materials, and the work itself be ready for the engineers when they move to the next task? Will everything show up precisely when it is needed? Will there be no delays in the work preceding what the engineers are expected to do? Will there be no scope changes in the preceding work, thus delaying the succeeding tasks scheduled for the engineers? As you can see, we expect there to be "unexpected" delays for multiple reasons, hence the admonition to never schedule a resource for more than 85–90 percent of its capacity.

But what about manufacturing situations in which machines and processes are commonly run near capacity for extended durations? These situations are not projects but rather routine production environments, assembly lines, for example. New work is scheduled to arrive precisely when the previous work has been completed. Planning is extensive, maintenance is carefully scheduled, experience in what can go wrong is abundant, resources are carefully controlled and monitored, and so on. That is not the situation of projects, which by definition are nonroutine. Depending on experience, when planning routine types of manufacturing processes, we try to have line capacity just slightly in excess of our average demand for the line's output. This policy is a sure course to disaster when applied to project management.

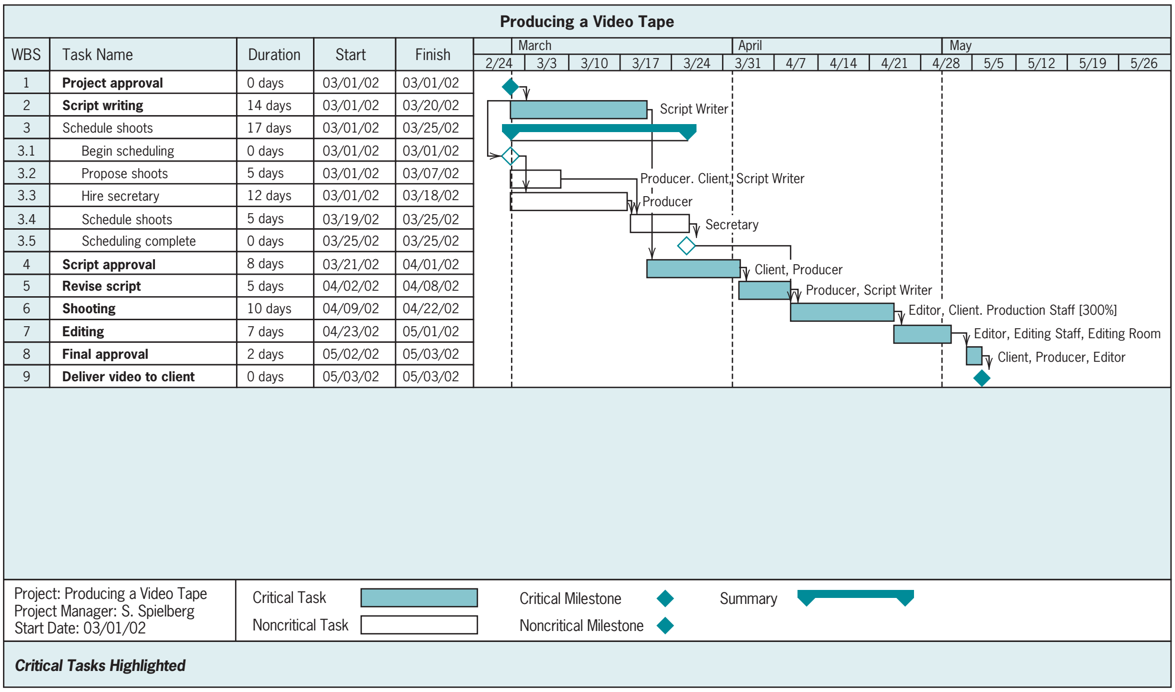
Now, what do we do about our software engineers? As it happens, some groups of professionals, such as engineers, are employed with the understanding that there will be periods of overtime required (for which they are generally not paid) and periods when things will be slack and they are relatively free to come and go as they please. In reality, engineers often work 50 to 60 hours per week for extended periods, and if a prolonged period of insufficient work is available at the company, management may lay off some engineers. As can be seen, a workweek of, say, 55 hours $\times$ 21 engineers $\times$ 34 weeks $\times$ 85 percent capacity $=$ 33,379 labor-hours, more than sufficient for the 28,282 labor-hours required—but not much more than sufficient.

9.5 CONSTRAINED RESOURCE SCHEDULING

Far too often, PMs are surprised by resource constraints. The cause of this condition is usually the direct result of a failure to include resource availability in risk identification activities. The lack of a resource where and when it is needed can have many causes, but the most common causes are not difficult to identify and mitigate: failure of a supplier to produce and/or deliver, the assignment of the resource to another activity, and loss or theft of a resource. PMs often apply risk management techniques to resources known to be scarce, but neglect to consider the more common resources that usually cause the problems.

There are two fundamental approaches to constrained resource allocation problems: heuristics and optimization models. Heuristic approaches employ rules of thumb that have been found to work reasonably well in similar situations. They seek better solutions. Optimization approaches seek the best solutions but are far more limited in their ability to handle complex situations and large problems. We will discuss each separately.

Most PC software designed for project management will level resources and solve the problems of overscheduling resources. They require priority rules to establish which activities take precedence. The priority rules the programs use vary somewhat, but most packages offer a choice. For example, reconsider the video tape project used to demonstrate MSP output forms in Chapter 8. We can include the resource requirement for each activity directly on the Gantt chart, as in Figure 9-10, but we can also show separate diagrams that illustrate the demand or "load" for each of the resources as in Figure 9-11. Figures 9-11 and 9-12 show a resource conflict for the producer and the resource leveling solution. Note the changes in the scheduled finish dates for the leveled solution.



WBS	Task Name	Duration	Start	Finish
1	**Project approval**	0 days	03/01/02	03/01/02
2	**Script writing**	14 days	03/01/02	03/20/02
3	Schedule shoots	17 days	03/01/02	03/25/02
3.1	Begin scheduling	0 days	03/01/02	03/01/02
3.2	Propose shoots	5 days	03/01/02	03/07/02
3.3	Hire secretary	12 days	03/01/02	03/18/02
3.4	Schedule shoots	5 days	03/19/02	03/25/02
3.5	Scheduling complete	0 days	03/25/02	03/25/02
4	**Script approval**	8 days	03/21/02	04/01/02
5	**Revise script**	5 days	04/02/02	04/08/02
6	**Shooting**	10 days	04/09/02	04/22/02
7	**Editing**	7 days	04/23/02	05/01/02
8	**Final approval**	2 days	05/02/02	05/03/02
9	**Deliver video to client**	0 days	05/03/02	05/03/02

Figure 9-10 MSP Gantt chart of video tape project showing resource needs.

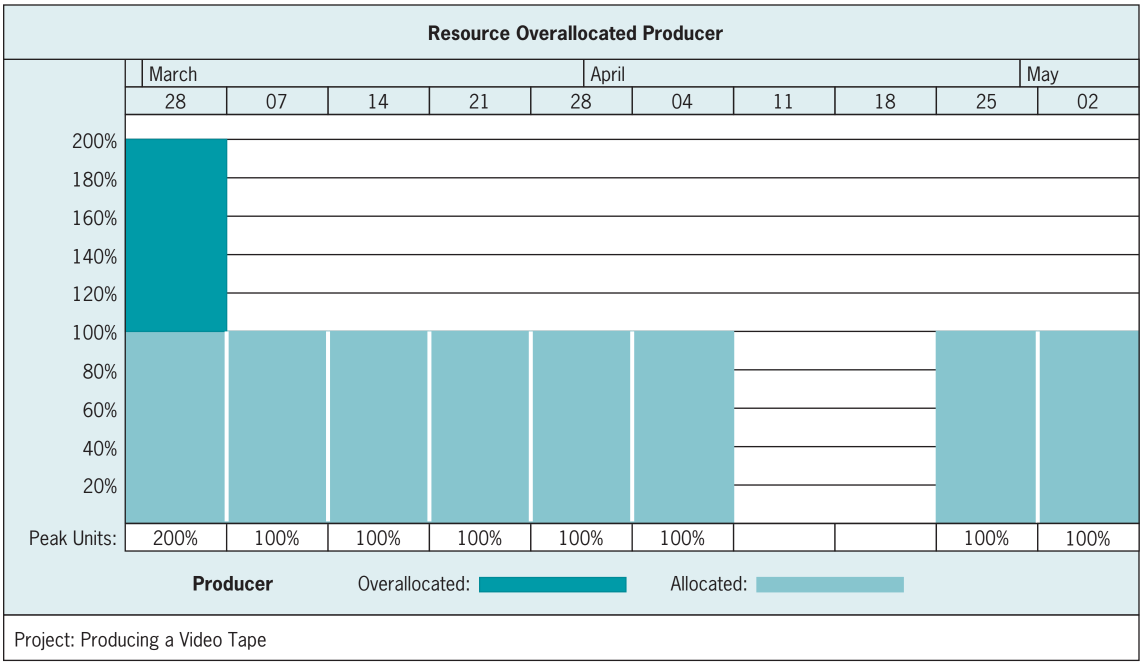
Heuristic Methods

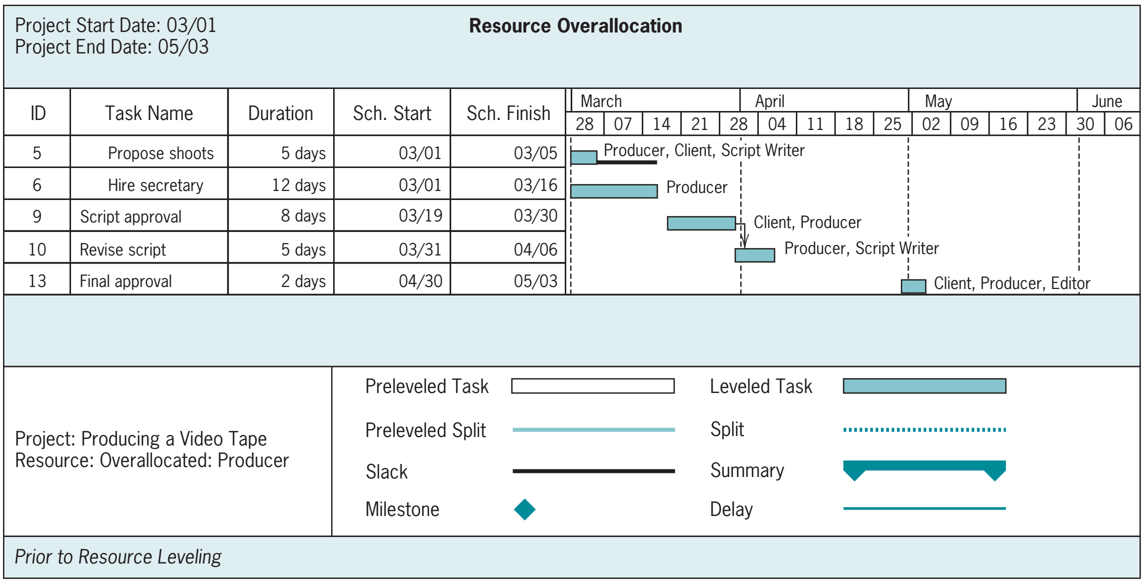
Heuristic approaches to constrained resource scheduling problems are in wide, general use for a number of reasons. First, they are the only feasible methods of attacking the large, nonlinear, complex problems that tend to occur in the real world of project management. Second, while the schedules that heuristics generate may not be optimal, they are usually quite good—certainly good enough for most purposes. Commercially available computer programs handle large problems and have had considerable use in industry. Further, modern simulation techniques allow the PM to develop many different schedules quickly and to determine which, if any, are significantly better than current practice. If a reasonable number of simulation runs fail to produce significant improvement, the PM can feel fairly confident that the existing solution is a good one.

Most heuristic solution methods start with the PERT/CPM schedule and analyze resource usage period by period, resource by resource. In a period when the available supply of a resource is exceeded, the heuristic examines the tasks in that period and allocates the scarce resource to them sequentially, according to some priority rule. The major difference among the heuristics is in the priority rules they use. Remember that the *technological necessities always take precedence*. Some of the most common priority rules are:

As Soon as Possible The default rule for scheduling. This provides the general solution for critical path and time.

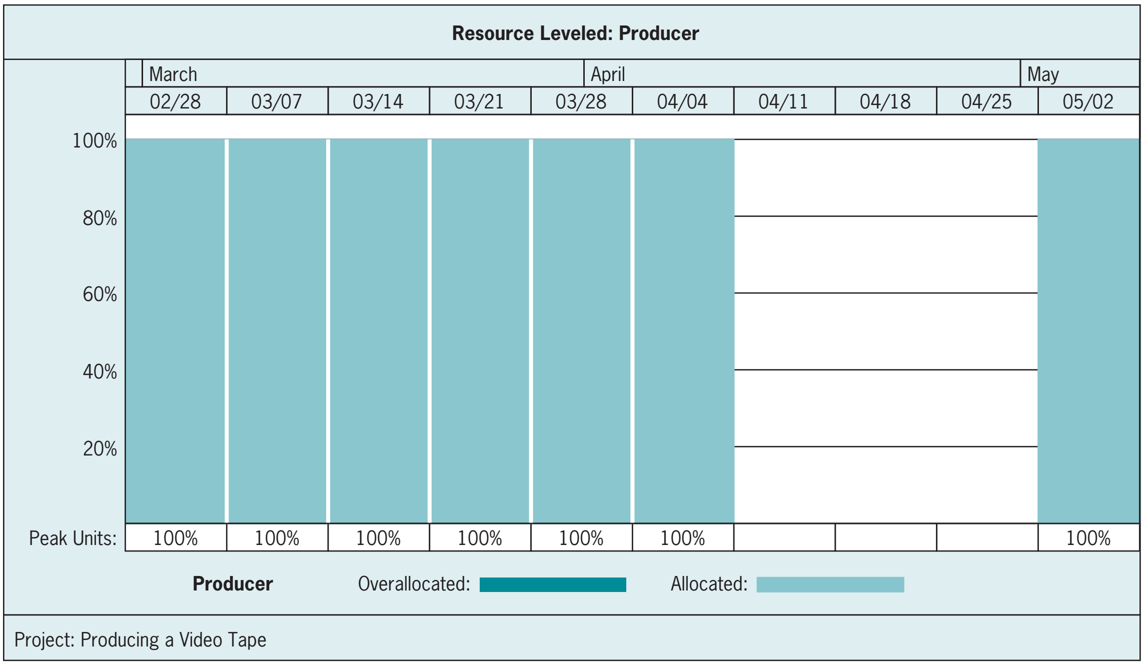
As Late as Possible All activities are scheduled as late as possible without delaying the project. The usual purpose of this heuristic is to defer cash outflows as long as possible.





ID	Task Name	Duration	Sch. Start	Sch. Finish
5	Propose shoots	5 days	03/01	03/05
6	Hire secretary	12 days	03/01	03/16
9	Script approval	8 days	03/19	03/30
10	Revise script	5 days	03/31	04/06
13	Final approval	2 days	04/30	05/03

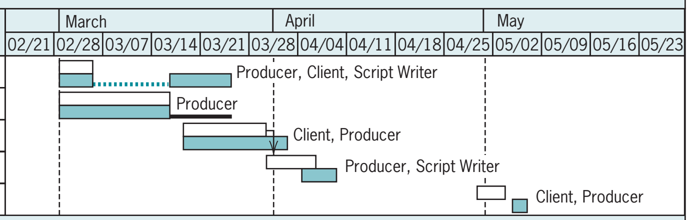
Figure 9-11 MSP load diagram showing resource conflict (producer used beyond capacity).



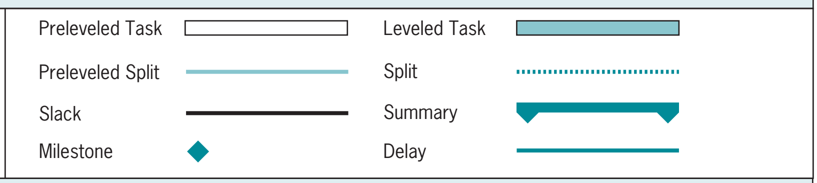
Project Start Date: 03/01
Project End Date: 05/06

Resource Leveled

ID	Task Name	Duration	Sch. Start
5	Propose shoots	12 days	03/01
6	Hire secretary	12 days	03/01
9	Script approval	11 days	03/19
10	Revise script	5 days	04/05
13	Final approval	2 days	05/05



Project: Producing a Video Tape
Resource: Producer



Resource Leveled

Figure 9-12 MSP rescheduling to level producer resource usage without exceeding capacity.

Shortest Task First Tasks are ordered in terms of duration, with the shortest first. In general, this rule will maximize the number of tasks that can be completed by a system during some time period.

Most Resources First Activities are ordered by use of a specific resource, with the largest user heading the list. The assumption behind this rule is that more important tasks usually place a higher demand on scarce resources.

Minimum Slack First This heuristic orders activities by the amount of slack, least slack going first. (It is common, when using this rule, to break ties by using the shortest-task-first rule.)

Most Critical Followers Tasks are arranged by number of critical activities following them. The ones with the greatest number of critical followers go first.

Most Successors This is the same as the previous rule, except that all followers, not merely critical ones, are counted.

Arbitrary Priorities are assigned to activities according to some rule not associated with task length, slack, or resource requirements. Such rules might be that tasks on projects of higher value to the parent organization (or for the project of a favored customer) are taken before those of lower value.

There are many such priority rules employed in scheduling heuristics. From time to time researchers subject several of the more popular of the project management software programs to tests of their ability to handle such tasks as allocating constrained resources and resource leveling. Although their findings vary somewhat because of slightly different assumptions, the minimum-slack-first rule was found to be best or near-best quite often and rarely caused poor performance. It usually resulted in the minimum amount of project schedule slippage, the best utilization of facilities, and the minimum total system occupancy time.

As the scheduling heuristic operates, one of two events will result. The routine runs out of activities (for the current period) before it runs out of the resources, or it runs out of resources before all activities have been scheduled. (Rarely is the supply of resources precisely equal to the demand.) If the former occurs, the excess resources are left idle, assigned elsewhere in the organization as needed during the current period, or applied to future tasks required by the project—always within the constraints imposed by the proper precedence relationships. If one or more resources are exhausted, however, activities requiring those resources are slowed or delayed until the next period when resources can be reallocated. For example, if the minimum-slack-first rule is used, resources would be devoted to critical or nearly critical activities, delaying those with greater slack. Delay of an activity uses some of its slack, so the activity will have a better chance of receiving resources in the next allocation. Repeated delays move the activity higher and higher on the priority list.

Optimizing Methods

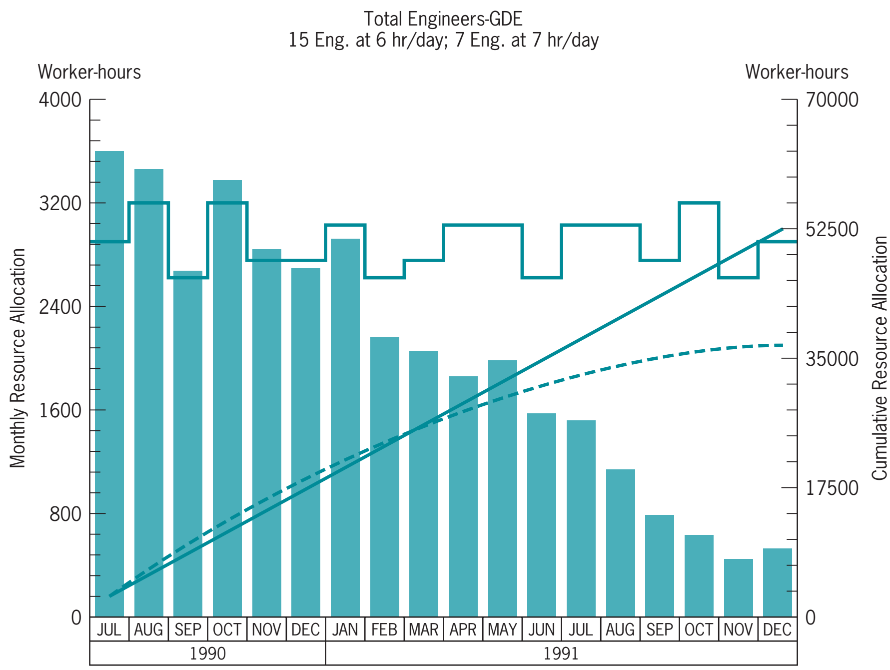
In the past several years, a wide range of attacks have been made on the problems of resource allocation and scheduling when resources are constrained. Some of these depend on sophisticated mathematical and/or graphical tools and may be quite powerful in what they can do. The methods to find an optimal solution to the constrained resource scheduling problem fall primarily into two categories: mathematical programming (linear programming, LP, for the most part) and enumeration. In the late 1960s and early 1970s, limited enumeration techniques were applied to the constrained resource problem with some success. Advances in linear programming (LP) techniques now allow LP to be used on large constrained resource scheduling problems. Other approaches have combined programming and enumeration methods.

Project Management in Practice

Benefits of Resource Constraining at Pennsylvania Electric

Pennsylvania Electric Company, headquartered in Johnstown, PA, operates generating facilities with a capacity of 6950 megawatts to serve 547,000 customers over an area of 17,600 square miles. The Generation Division Planning Group is responsible for planning all maintenance and capital projects. In the early 1980s, the group used a manual method of planning with hand-drawn charts. The planning process has now been computerized, which is faster, allows "what-if" analyses, and controls more than just the previously monitored critical path. In bringing the planning process in-house, the group also saved $100,000 a year in service fees from an outside engineering firm who was planning their construction activities.

A special feature of the computerized system is its resource constraining module which establishes labor requirements across all jobs. In the pilot program to test the new software, $300,000 was saved when it was discovered that a job could be done with 40 percent fewer mechanics than normally used and still complete the job on time. In another application, it was found that a turbine disassembly and inspection could be added to the task list without delaying the project or exceeding the project budget.



	JUL	AUG	SEP	OCT	NOV	DEC	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
AVAILABILITY	2919	3197	2641	3197	2780	2780	3058	2641	2780	3058	3058	2641	3058	3058	2780	3197	2641	2919
SCHEDULED	3599	3488	2685	3404	2865	2727	2949	2183	2083	1892	2010	1567	1541	1144	777	648	453	537

MONTHLY RESOURCE ALLOCATION WORKER-HOURS

AVAILABILITY
RES AVA TOTAL-GDE WORKER-HOURS
CURR SCH ES

SCHEDULED
RESOURCE TOTAL-GDE WORKER-HOURS
TARG SCH 45 FS

CUMULATIVE RESOURCE ALLOCATION WORKER-HOURS

AVAILABILITY
RES AVA TOTAL-GDE WORKER-HOURS
CURR SCH ES CUM

SCHEDULED
RESOURCE TOTAL-GDE WORKER-HOURS
TARG SCH 45 FS CUM

Total Engineering Resource Profile

After worker-hours are input to the program by activity, actual progress is monitored (see figure) and schedule and cost deviations are highlighted for management attention. This allows management to make adjustments to recover the schedule, slow the project down, or acquire more funds to get the project back on schedule. Obviously, there are always some emergencies outside the plan that must be handled on an exception basis. But with this software, management knows what effect different actions will have on the basic plan and can thereby make the best use of available resources to handle the emergency with minimal impact on the plan.

Source: A. J. Cantanese, "At Penelec, Project Management Is a Way of Life," *Project Management Journal*, December 1990.

9.6 MULTIPROJECT SCHEDULING AND RESOURCE ALLOCATION

In Chapter 2, we described a method for strategically selecting the best set of projects that would help achieve the organization's strategic goals. This method, called the Project Portfolio Process, ended with a set of high-priority projects that could be adequately funded by the organization. However, the process did not consider the scheduling and allocation of limited, individual critical resources among the projects. In some cases, a particular machine, or skilled employee may be needed on two or more projects at the same time. This involves multiproject scheduling and resource allocation.

Scheduling and allocating resources to multiple projects are much more complicated than for the single-project case. The most common approach is to treat the several projects as if they were each elements of a single large project. (A more detailed explanation is given below when we consider a specific multiproject scheduling heuristic.) Another way of attacking the problem is to consider all projects as completely independent; see the works by Kurtulus et al. (1982, 1985) for example. As they show, these two approaches lead to different scheduling and allocation outcomes. For either approach, the conceptual basis for scheduling and allocating resources is essentially the same.

There are several projects, each with its own set of activities, due dates, and resource requirements. In addition, the penalties for not meeting time, cost, and performance goals for the several projects may differ. Usually, the multiproject problem involves determining how to allocate resources to, and set a completion time for, a new project that is added to an existing set of ongoing projects. This requires the development of an efficient, dynamic multiproject scheduling system.

To describe such a system properly, standards are needed by which to measure scheduling effectiveness. Three important parameters affected by project scheduling are: (1) schedule slippage, (2) resource utilization, and (3) in-process inventory. The organization (or the PM) must select the criterion most appropriate for its situation.

Schedule slippage, often considered the most important of the criteria, is the time past a project's due date or delivery date when the project is completed. Slippage may well result in penalty costs that reduce profits. Further, slippage of one project may have a ripple effect, causing other projects to slip. Indeed, expediting a project in order to prevent slippage may, and usually does, disturb the overall organization to the point where slippage due to resource shortages may then be caused in other projects. The loss of goodwill when a project slips and deliveries are late is important to all producers. As is the case with many firms, Grumman Aircraft, purchased by the Northrup Corporation in 1994, jealously guards its reputation for on-time delivery. During a project to install a new machine control system on a production line, Grumman insisted that the project be designed to minimize disturbance to operations in the affected plant and avoid late shipments. This increased the cost of the project, but the firm maintained delivery schedules.

A second measure of effectiveness, *resource utilization*, is of particular concern to industrial firms because of the high cost of making resources available. A resource allocation system that smooths out the peaks and valleys of resource usage is ideal, but it is extremely difficult to attain while maintaining scheduled performance because all the projects in a multiproject organization are competing for the same scarce resources. In particular, it is expensive to change the size of the human resource pool on which the firm draws.

While it is relatively easy to measure the costs of excess resource usage required by less than optimal scheduling in an industrial firm, the costs of uncoordinated multiproject scheduling can be high in service-producing firms, too. In the real estate syndication firm example at the end of Chapter 8, the scarce resource is executive judgment time. If two deals arrive at the same time, one must wait. This is undesirable because other potential buyers are seeking properties, and the process must move along without delay.

The third standard of effectiveness, the amount of *in-process inventory*, concerns the amount of work waiting to be processed because there is a shortage of some resource(s). Most industrial organizations have a large investment in in-process inventory, which may indicate a lack of efficiency and often represents a major source of expense for the firm. The remedy involves a trade-off between the cost of in-process inventory and the cost of the resources, usually capital equipment, needed to reduce the in-process inventory levels. It is almost axiomatic that the most time-consuming operation in any production system involving much machining of metals is an operation called "wait." If evidence is required, simply observe parts sitting on the plant floor or on pallets waiting for a machine, or for jigs, fixtures, and tools.

These criteria cannot be optimized at the same time. As usual, trade-offs are involved. A firm must decide which criterion is most applicable in any given situation, and then use that criterion to evaluate its various scheduling and resource allocation options.

At times, the demands of the marketplace and the design of a production/distribution system may require long production runs and sizable levels of in-process inventory. This happens often when production is organized as a continuous system, but sales are organized as projects, each customized to a client order. Items may be produced continuously but held in a semifinished state and customized in batches.

A mattress manufacturing company organized to produce part of its output by the usual continuous process; but the rest of its production was sold in large batches to a few customers. Each large order was thought of as a project and was organized as one. The customization process began after the metal frames and springs were assembled. This required extensive in-process inventories of semifinished mattresses.

As noted earlier, the minimum-slack-first rule is the best overall priority rule and generally results in minimum project slippage, minimum resource idle time, and minimum system occupancy time (i.e., minimum in-process inventory). But the most commonly used priority rule is first-come, first-served—which has little to be said for it except that it fits the client's idea of what is "fair," if the client is at the head of the line. In any case, individual firms may find a different rule more effective in their particular circumstances and should evaluate alternative rules by their own performance measures and system objectives.

Manufacturing Process Models

The general approach implied by the previous discussion is adopted from the model of a job-shop manufacturing system. The criteria for measuring schedule effectiveness are those applied to job-shops, and several approaches to resource-constrained multiple project scheduling use the job-shop model. A scarce resource required for several projects or by several different activities in one project acts like a bottleneck in a manufacturing system. The bottleneck acquires a waiting line or queue. A short digression into queuing theory is in order.

Assume a random (Poisson) arrival rate, λ, of jobs to be processed by a facility, and the facility's rate of servicing jobs is also random (exponential), denoted μ. The average number of jobs, J, waiting in line for service is $J = (\lambda/\mu)/(1 - \lambda/\mu)$. Clearly, as the arrival rate approaches the service rate, the queue length heads toward infinity. The result is that to avoid long waiting lines for projects or for activities within a project, the capacity of the scarce resource must be significantly greater than the demand for it, or else the arrival rate of jobs must be tightly controlled so that it is no longer random. Even if the arrival rate is controlled, it must still be at a level that is less than the service rate (which is still random).

Adler et al. (1996) found that some highly successful firms had been applying work-process management to product development projects. Specifically, they found that projects are completed faster when the firm does fewer of them, that increasing bottleneck capacity pays large dividends to the investment, and that eliminating unnecessary workload and processes decreases the variation in service times (p. 134). Levy et al. (1997) also adopt queuing theory to deal with multiproduct management. They illustrate calculations for the "cost of waiting in line," the "cost of underutilizing" a facility with capacity greater than the demand for it, and "the cost of delayed projects." They also discuss ways of reducing these costs.

Given these observations, let us examine some examples of the various types of multiproject scheduling and resource allocation techniques. We begin with a short description of several heuristics, and then discuss one heuristic in greater detail.

Heuristic Techniques

Because of the difficulties with the analytical formulation of realistic problems, major efforts in attacking the resource-constrained multiproject scheduling problem have focused on heuristics. We touched earlier on some of the common general criteria used for scheduling heuristics. Let us now return to that subject.

There are scores of different heuristic-based procedures in existence. A great many of the procedures have been published (see Davis et al., 1975, for example), and descriptions of some are generally available in commercial computer programs.

The most commonly applied rules were discussed in Section 9.5. The logical basis for these rules predate AOA and AON. They represent rather simple extensions of well-known approaches to job-shop scheduling. Some additional heuristics for resource allocation have been developed that draw directly on AOA and AON. All these are commercially available for computers, and most are available from several different software vendors in slightly different versions.

Resource Scheduling Method In calculating activity priority, give precedence to that activity with the minimum value of d_{ij}, where

$$\begin{aligned} d_{ij} &= \text{increase in project duration resulting when activity } j \text{ follows activity } i. \\ &= \text{Max}\left[0; \left(\text{EF}_i - \text{LS}_j\right)\right] \end{aligned}$$

where

EF_i = early finish of activity $\boldsymbol{i}$

LS_j = latest start of activity $\boldsymbol{j}$

The comparison is made on a pairwise basis among all activities in the conflict set.

Minimum Late Finish Time This rule assigns priorities to activities on the basis of activity finish times as determined by AOA or AON. The earliest late finishers are scheduled first.

Greatest Resource Demand This method assigns priorities on the basis of total resource requirements, with higher priorities given for greater demands on resources. Project or task priority is calculated as

$$\text{Priority} = d_j \sum_{i=1}^{m} r_{ij}$$

where

d_j = duration of activity $\boldsymbol{j}$

r_{ij} = per period requirement of resource $\boldsymbol{i}$ by activity $\boldsymbol{j}$

m = number of resource types

Resource requirements must be stated in common terms, usually dollars. This heuristic is based on an attempt to give priority to potential resource bottleneck activities.

Greatest Resource Utilization This rule gives priority to that combination of activities that results in maximum resource utilization (or minimum idle resources) during each scheduling period. The rule is implemented by solving a 0-1 integer programming problem, as described earlier. This rule was found to be approximately as effective as the minimum slack rule for multiple project scheduling, where the criterion used was project slippage.

Most Possible Jobs Here, priority is given to the set of activities that results in the greatest number of activities being scheduled in any period. This rule also requires the solution of a 0-1 integer program. It differs from the greatest-resource-utilization heuristic in that the determination of the greatest number of possible jobs is made purely with regard to resource feasibility (and not with regard to any measure of resource utilization).

Heuristic procedures for resource-constrained multiproject scheduling represent the only practical means for finding workable solutions to the large, complex multiproject problems normally found in the real world. Let us examine a multiproject heuristic in somewhat more detail.

A Multiproject Scheduling Heuristic

To attack this problem, recall the hierarchical approach to project planning we adopted in Chapter 6. A project plan is a nested set of plans, composed of a set of generalized tasks, each of which is decomposed into a more detailed set of work packages that are, in turn, decomposed further. The decomposition is continued until the work packages are simple enough to be considered "elemental." A network diagram of a project might be drawn for any level of task aggregation. A single activity (arrow) at a high level of aggregation would represent an entire network of activities at a lower level (see Figure 9-13). Another level in the planning hierarchy is shown as a Gantt chart in Figure 9-14.

If an entire network is decomposed into subnetworks, we have the equivalent of the multiproject problem where each of the projects (subnetworks) is linked to predecessor and successor projects (other subnetworks). In this case, the predecessor/successor relationships depend on the technology of the parent project. In the true multiproject case, these relationships may still depend on technological relationships; for example, a real estate development project being dependent on the outcome of a land procurement project. The relationships may, however, be determined more or less arbitrarily, as when projects are sequenced on a first-come, first-served basis, or by any other priority-setting rule, or

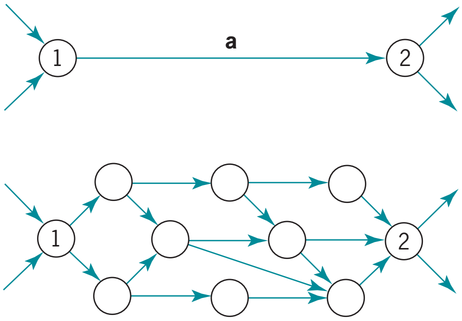


Figure 9-13 Task **a** decomposed into a network of subtasks.

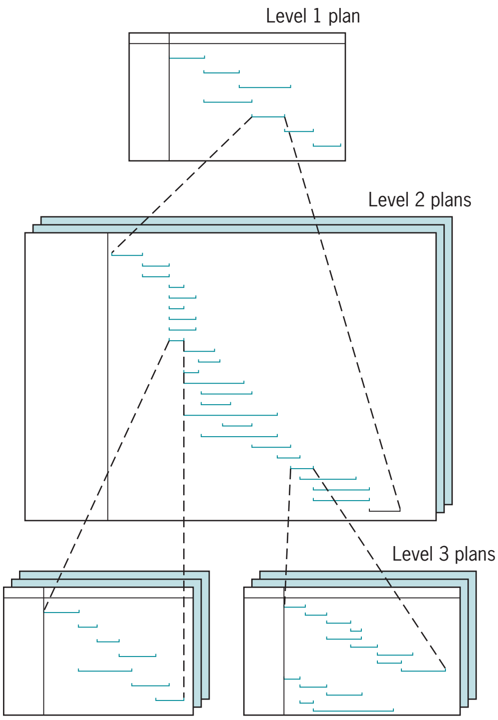


Figure 9-14 Hierarchy of Gantt charts. *Source:* F.L. Harrison (1983). *Advanced Project Management*. Hants, U.K.: Gower.

undertaken simultaneously in the hope that some synergistic side effects might occur. Or the relationship among the projects may simply be that they share a common pool of resources.

With this conceptual model, assume we have a set of projects. Each individual project is represented by a network of tasks. We can form a single network of these projects by connecting them with dummy activities (no resources, no duration) and/or pseudoactivities (no resources, some duration). Both dummy activities and pseudoactivities represent dependency relationships, but these dependencies, as noted above, may be technological or quite arbitrary.*

*This exposition is based on Weist's (1967) work, and on Corwin's (1968) application of Weist's papers to resource allocation among multiple R&D projects.

As usual, and excepting dummy and pseudoactivities, each task in each network requires time and resources. The amount of time required may or may not vary with the level of resources applied to it. The total amount of resources and/or amounts of individual resources are limited in successive scheduling periods. Our problem is to find a schedule that best satisfies the sequence and resource constraints and minimizes the overall duration of the entire network. The resulting schedule should indicate when to start any activity and at what level of resources it should be maintained while it is active.

Before undertaking the allocation of resources, it is proper to consider the quantity of resources available for allocation. (For the moment, we consider "resources" as an undifferentiated pool of assets that can be used for any purpose.) At the beginning of any period (hour, day, week, month, etc.) we have available any resources in inventory, R_I, which is to say, left over as excess from the previous allocation process. Changes in the inventory can be made from within the system of projects or by importing or exporting inventory from the outside. Excluding activities that have been completed in previous periods, every activity planned by the project is in one of four states; ongoing, stopping, waiting and technologically able to start, or waiting and technologically unable to start.

Figure 9-15 illustrates these conditions. We label ongoing activities as "resource users." Those stopping are "resource contributors." Those waiting and able to start are "resource demanders." Those waiting and unable to start can be ignored for the present. The amount of resources available for allocation is, therefore, the amount in inventory plus the amount contributed, $R_I + R_C$. If the amount demanded is less than this sum, there will be a positive inventory to start the next period. If not, some demanders will go unfunded.

Weist's heuristic (SPAR-1, Scheduling Program for Allocation of Resources) allocates resources to activities in order of their early start times. In the first period, we would list all available tasks and order them by their slack, from least to most. (Calculation of slack is based on the assumption that activities will be supported at *normal* resource levels.) Activities are selected for support and scheduling one by one, in order. As activities at the top of the list are supported, the relevant resource stocks are debited. Tasks are scheduled sequentially until the list of available jobs is completed, or until the stock of one or more necessary resources is depleted. If we deplete resources before completing the task list, remaining tasks are delayed until the next period. Postponed activities lose slack and rise toward the top of the priority list.

Thus, resources are devoted to activities until the supply of available resources or activities is exhausted. If we use up the resources before all critical activities are scheduled, we can adopt one of two subheuristics. First, we may be able to borrow resources from

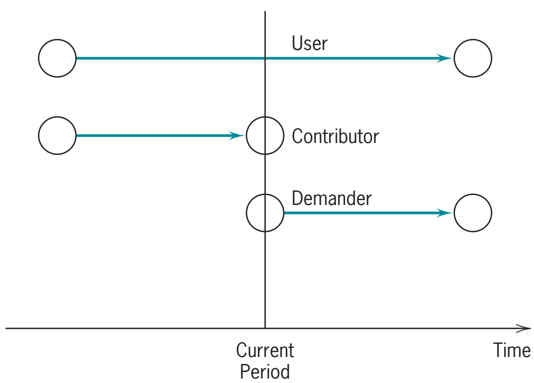


Figure 9-15 Sources and uses of resources.

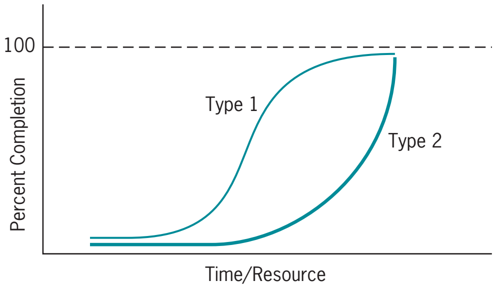


Figure 9-16 Project or task life cycles.

currently active, but noncritical, tasks. Second, we may “deschedule” a currently active, noncritical task. The former presumably slows the progress of the work, and the latter stops it. In both cases, some resources will be released for use on critical tasks. Obviously, if a critical task is slowed, descheduled, or not supported, the duration of the associated project will be extended.

The decision about which of these courses of action to take, borrowing or descheduling, can be made by adopting the same logic used in Chapter 7 when we discussed the budget negotiations between subordinate and superior. The decision to borrow or deschedule depends on our estimate of the impact either action would have on the task under consideration, given its current state of completion. Figure 9-16 shows two different versions of the project or task life cycle discussed in Chapter 7. If the task is a Type 1, borrowing would minimize the damage to the task unless it is quite near completion and we are willing to accept the outcome in its current state, in which case we can deschedule. If the task is Type 2, borrowing is apt to have a catastrophic effect on the task and we should either deschedule it (and start it again later) or reject it as a source of resources. If there is more than one scarce resource, a separate activity can be created for each type of scarce resource. These “created” activities must be constrained to start in the same period as the parent activity, and to have the same level of resource assignment (normal, crash, or minimal).

As we have noted, many commercially available software packages have the ability to schedule constrained resources and deal with resource conflicts. Many of the packages will allow the user to solve the problem either automatically, using the program’s heuristics, or by hand in which case the user can adopt any method desired. If a set of projects is linked together by dummy activities so that it can be treated like a single project, the software will report resource usage conflicts; that is, cases in which the scheduled utilization of a resource is greater than the supply of that resource.

In one sense this chapter’s emphasis on resource shortages is misleading. The common case of shortage applies not to resources in general, but to one or two highly specific resources. For example, an insurance firm specializing in casualty insurance has a typical kind of scarce resource, a “Walt.” Walter A. is a specialist in certain types of casualty losses in the firm’s commercial lines business. He is the only such specialist in the firm, and his personal knowledge is required when designing new policies in the field. His knowledge is based on years of experience and an excellent, analytical mind. It is common for projects involving the modification or creation of policies in the commercial lines area to have problems associated with the fact that the firm has one, and only one Walt. Walt-capacity cannot be hired, trained, or subcontracted within an appropriate time frame. The firm’s ability to extend its current Walt-capacity is not sufficient to satisfy its current Walt-demand. Left with no alternative, some projects must be delayed so that others can proceed.