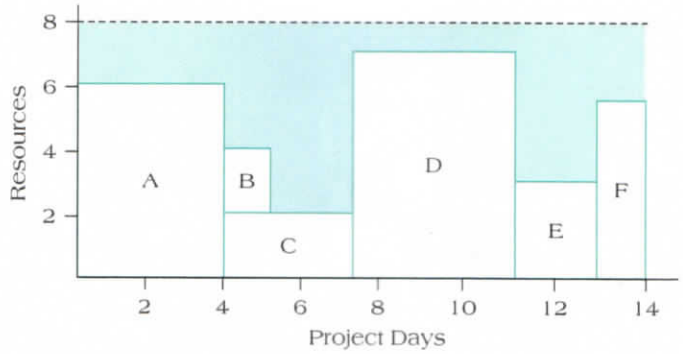

FIGURE 12.18 **Resource-Loading Chart for Sample Project**

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

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

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

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

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

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