Critical Chain Project Scheduling: Do Not Oversimplify

Willy Herroelen, Katholieke Universiteit Leuven, Department of Applied Economics, Naamsestraat 69, Leuven B-3000 Belgium

Roel Leus, Katholieke Universiteit Leuven, Department of Applied Economics, Naamsestraat 69, Leuven B-3000 Belgium

Erik Demeulemeester, Katholieke Universiteit Leuven, Department of Applied Economics, Naamsestraat 69, Leuven B-3000 Belgium

Abstract

Critical chain scheduling/Buffer management (CCS/BM)—the direct application of the theory of constraints (TOC) to project management—has received much attention in project management literature. There still is controversy over the merits and pitfalls of the CCS/BM methodology. This paper focuses on the fundamental elements of CCS/BM logic and pinpoints some intricacies that are not commonly referred to in the available literature. The authors' analysis is based on a critical review of the relevant sources and experimentation with both commercial CCS/BM software and an internally developed CCS/BM-based tool.

Keywords: critical chain; buffer management; resource-constrained project scheduling

2002, Vol. 33, No. 4, 48–60
8756–9728/02/$10.00 per article + $0.50 per page

Critical chain scheduling/Buffer management (CCS/BM)—the direct application of the theory of constraints (TOC) to project management—has received much attention in the project management literature. Subsequent to the publication of Goldratt's book *Critical Chain* in 1997, numerous books (Newbold, 1998; Leach, 2000) and papers (Cabanis-Brewin, 1999; Globerson, 2000; Herroelen & Leus, 2000, 2001; Leach, 1999; Maylor, 2000; Patrick, 1999; Pinto, 1999; Rand, 2000; Umble & Umble, 2000) have been written to clarify the CCS/BM philosophy. Internet discussion groups also have been set up to discuss critical chain scheduling issues. These groups include the critical chain Yahoo group (http://groups.yahoo.com/group/criticalchain) and more general TOC-oriented groups such as the Yahoo TOC Experts (http://groups.yahoo.com/group/tocexperts) and the APICS Constraints Management special interest group (accessible via http://www.apics.org/lists/default.htm).

While real-world applications by companies such as Lucent Technologies and Harris Semiconductor have been described to demonstrate the effectiveness of the CCS/BM approach (Leach, 1999; Umble & Umble, 2000), other sources (Zalmenson & Zinman, 2000, 2001) alert the reader to serious drawbacks and implementation failures or at least claim CCS/BM is not at all innovative (Wilkens, 2000). There still is a lot of controversy over the merits and pitfalls of the CCS/BM methodology. Being confronted with many views and comments on CCS/BM written by either true believers or committed naysayers who fail to offer a fair appraisal of the CCS/BM project technique, *PM Network*, the professional magazine of the Project Management Institute, put an editor's note in its January 2001 issue stating "*PM Network* is neutral on Theory of Constraints; the publishing division holds no allegiance either for or against it." The objective of this paper is not to defend or reject CCS/BM. The CCS/BM methodology has much to offer if applied wisely and if the practical implications and limitations are well understood. The authors' analysis is based on a critical review of the relevant literature and experimentation with both commercial CCS/BM software and an internally developed CCS/BM-based tool (Herroelen & Leus, 2000).

The CCS/BM Methodology

The fundamentals of CCS/BM are summarized in Table 1. CCS/BM builds a baseline schedule using activity duration estimates based on a 50% confidence level. Activity due dates and project milestones are eliminated, and multitasking is to be avoided.

50% probability activity duration estimates
No activity due dates
No project milestones
No multitasking
Scheduling objectives = minimize makespan; minimize work in progress
Determine a precedence and resource-feasible *baseline schedule*
Identify the critical chain
Aggregate uncertainty allowances into buffers
Keep the baseline schedule and the critical chain fixed during project execution
Determine an early start-based unbuffered *projected schedule* and report early completions (apply the roadrunner mentality)
Use the buffers as a proactive warning mechanism during schedule execution

Table 1. Fundamentals of CCS/BM

Prioritize the projects
Plan the individual projects according to CCS/BM fundamentals
Stagger the projects by inserting capacity buffers
Insert drum buffers
Measure and report the buffers
Manage the buffers

Table 2. CCS/BM Multiproject Method

To minimize work in progress, a precedence-feasible schedule is constructed by timing activities at their latest start dates based on critical path calculations. If resource conflicts occur, they are resolved by moving activities earlier. The critical chain then is defined as that chain of precedence and resource dependent activities that determines the overall duration of a project. If there is more than one critical chain, just select one.

The safety time that is eliminated from the critical chain activity durations by selecting aggressive duration estimates is shifted to the end of the critical chain in the form of a project buffer (PB). This PB should protect the project due date promised to the customer from variability in the critical chain activities.

Feeding buffers (FB) are inserted whenever a noncritical chain activity joins the critical chain. Their aim is to protect the critical chain from disruptions on the activities feeding it and to allow critical chain activities to start early in case things go well. Although more detailed methods can be used for sizing the buffers (Newbold, 1998; Product Development Institute, 1999), the default procedure is to use the 50% buffer sizing rule, i.e., to use a PB of half the project duration and to set the size of a FB to half the duration of the longest noncritical chain path leading into it. Resource buffers (RB), usually in the form of an advance warning, are placed whenever a resource has to perform an activity on the critical chain, and the previous critical chain activity is done by a different resource.

During project execution, both the critical chain and the baseline schedule should be fixed. Project activities are executed according to the roadrunner mentality using an unbuffered projected schedule. This schedule is early start-based, except for the gating tasks (activities without predecessors), which are to be started at their baseline scheduled start times. Early completion of activities must be reported, and activities should be started as soon as work becomes available. The execution of the project is managed by using the buffers as a proactive warning mechanism.

As activities are completed, project management should keep track of buffer consumption. As long as a predetermined portion of the buffer remains [buffer consumption is only in the green (O.K.) zone], everything is assumed to go well. If buffer consumption moves beyond a certain point, a warning is raised (the yellow watch-and-plan zone). If it deteriorates past a critical point (the red act zone), corrective action must be taken.

In a multiproject environment, CCS/BM relies on the six common steps summarized in Table 2:

1. Prioritize the organization's projects;
2. Plan the individual projects according to the CCS/BM fundamentals;
3. Stagger the projects;
4. Insert drum buffers;
5. Measure and report the buffers;
6. Manage the buffers.

Step 1 aims to avoid multitasking among projects. Step 2 calls for the identification of the most constraining company resource as the "bottleneck" (strategic or drum resource). Projects are staggered (Step 3) on the basis of the schedule for the strategic resource (drum plan). This is done by placing a capacity buffer in front of a strategic resource activity that is on the critical chain in the strategic resource schedule. A drum buffer (Step 4) is placed before activities on the strategic resource to protect the strategic resource from disruptions on nonstrategic resources. The buffers then can be managed more or less the same way as done in the single-project case (Steps 5 and 6).

CCS/BM Scheduling Approach

For discussions on a more managerial level, the authors refer to Pinto (1999), Elton and Roe (1998), and McKay and Morton (1998). The authors focused on the procedures for identifying the critical chain in the baseline schedule, buffering the baseline schedule, using the buffers as a proactive mechanism, and the issue of stability and rescheduling.

Throughout the discussion, the authors refer to major findings of a factorial experiment conducted by Herroelen and Leus (2001) on the well-known Patterson (1984) test problem set, which contains test instances with a number of activities ranging from seven to 50 and requiring up to three renewable resource types each. The authors confronted the results obtained using an exact solution procedure for the resource-constrained project scheduling problem (Demeulemeester & Herroelen, 1992, 1997) with the results generated using their own computerized CCS/BM software and tested the impact of the project scheduling mechanism, the buffer size computation method, the composition of the critical chain, and the recomputation mechanism of the baseline schedule.

Identifying the Critical Chain

One of the fundamental merits of CCS/BM is that it explicitly recognizes it is the interaction between activity durations, precedence relations, resource requirements, and resource availabilities that determines the project duration. This interaction results in one or more sequences of activities—consisting of precedence-related and resource-dependent segments—that determine the length of the baseline schedule. Goldratt (1997) identifies such a sequence as the critical chain. This concept is not new; Wiest (1964) introduced the concept of a critical sequence more than 30 years ago. It has been used by many researchers in project scheduling to compute a lower bound on the project duration (Herroelen, De Reyck, & Demeulemeester, 1998).

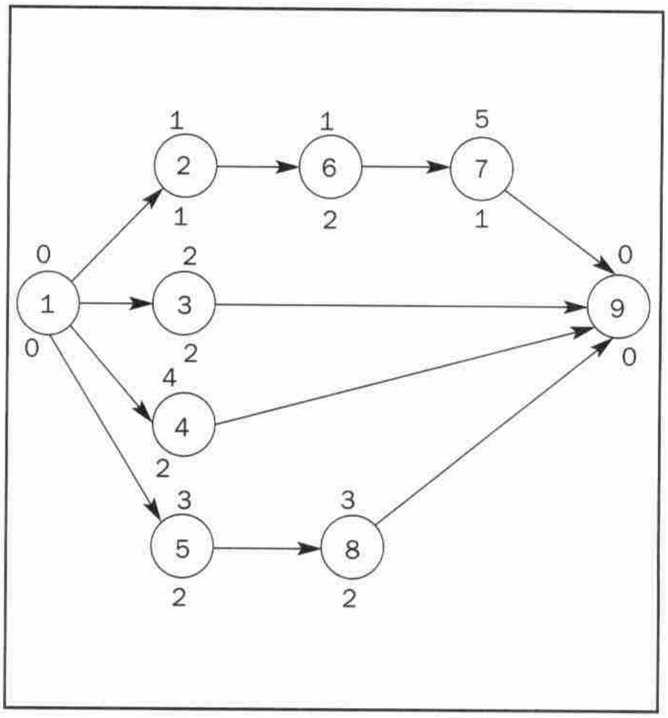


Figure 1. Project Network Example

CCS/BM has revealed the importance of this critical chain to the practitioner and plays a crucial role in the awakening process to the disappearance of the traditional critical path concept in the presence of resource constraints. Moreover, CCS/BM's argument that time protection against uncertainty should not be allocated to the individual activities but should be aggregated and concentrated at particular points in the baseline schedule is well-taken and commonly accepted in TOC-based scheduling. In short, the idea of generating a deterministic baseline schedule and protecting it against uncertainty is sound and appeals to management. CCS/BM paved the way for the recent introduction of protective buffering mechanisms into commercial project planning software. Examples include ProChain as an add-in to Microsoft Project® and the critical chain-planning functions of Scitor's PS8 package.

A baseline schedule may contain more than one critical chain, and the composition of the critical chain(s) is entirely dependent on the procedure used for generating the baseline schedule. Creating a precedence and resource-feasible baseline schedule that minimizes the project duration—the number one objective used by CCS/BM's baseline scheduling logic—is not easy because most resource-constrained project scheduling problems are NP-hard (they cannot always be solved to optimality in polynomial time). Goldratt (1997, p. 217) seems to minimize the issue and claims that "in each case the impact on the lead time of the project is less than even half the project buffer" and suggests (p. 220) to cut for each step a piece of paper so that "the length represents time. This way we can move them around until there is no contention."

Also, there is a lack of consensus in the CCS/BM literature about the activity duration estimates to be used for generating

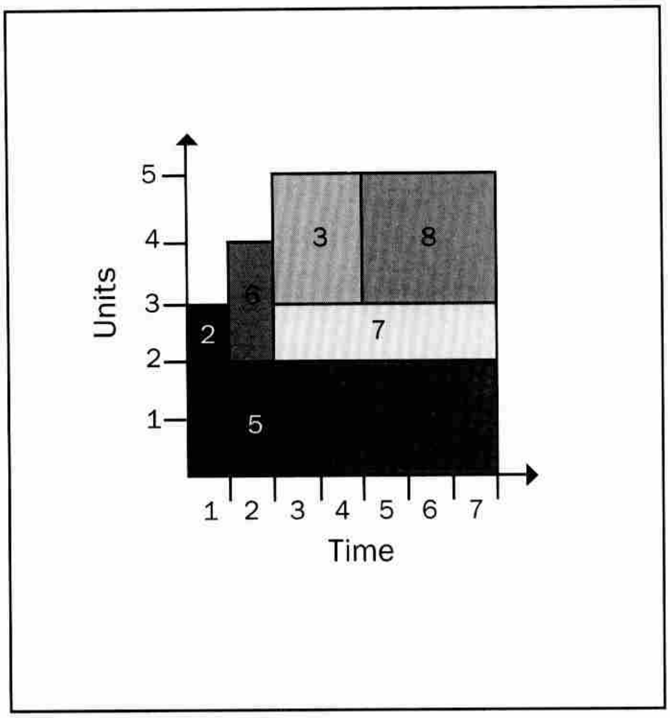


Figure 2. Minimum Makespan Schedule for the Network of Figure 1

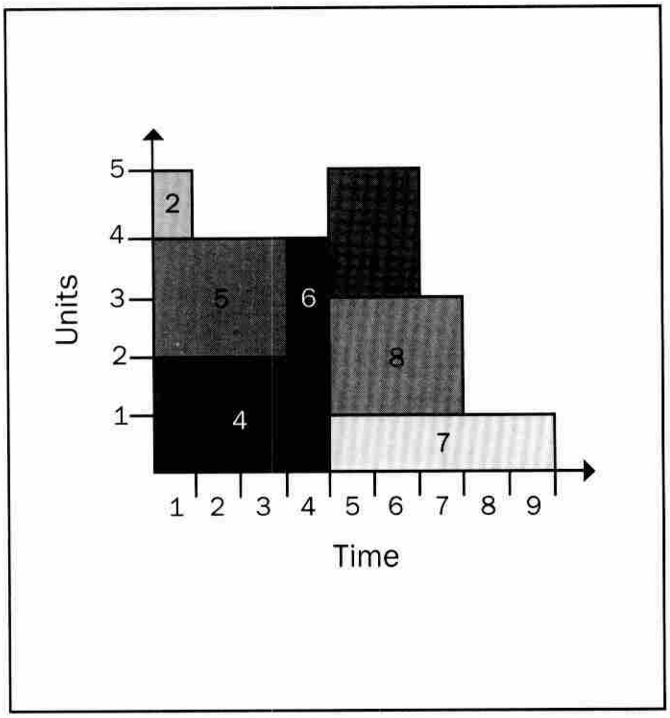


Figure 3. Schedule Obtained Using Microsoft Project

the initial baseline schedule. Goldratt (1997) suggests using the median, while the Product Development Institute (1999) argues in favor of the mean. Herroelen and Leus (2001) conclude from their full factorial experiment that the use of the mean activity duration provides the safest estimates of the project duration.

To illustrate that the length of the baseline schedule and the collection of possible critical chains is entirely dependent on the scheduling procedure used, consider the example network given in Figure 1 (Demeulemeester, Herroelen, Simpson, Baroum, Patterson, & Yang, 1994). The numbers above each node denote the activity duration of the corresponding activity (node 1 and node 9 are dummy nodes with zero duration). It is assumed that these activity duration estimates do not include individual safety provisions. The numbers below each node of the network in Figure 1 represent the number of units of a renewable resource, e.g., workers, required during every period of the corresponding activity's duration interval. It is assumed that the renewable resource has a constant availability of five units per period.

Figure 2 gives a minimum duration schedule. This can be obtained, for example, by using the branch-and-bound procedure of Demeulemeester and Herroelen (1992, 1997). As can be verified easily, the project duration is seven time periods (by poor luck this corresponds to the length of the longest critical path 1–2–6–7–9). The baseline schedule reveals three critical chains: the chain 2–6–7, the chain 2–6–3–8, and the chain 5–4.

Figure 3 shows the baseline schedule generated by Microsoft Project. This baseline schedule has a length of nine time periods and reveals two possible critical chains, different from the ones present in the optimal duration schedule of Figure 2: the chain 4–7 and the chain 5–6–7.

If the project planner would rely on the ProChain software, the schedule in Figure 4 would be obtained. The schedule now has a length of eight time periods and has two critical chains: the chain 3–5–8 and the chain 3–6–7. Both activity 2 and activity 4 have been right-shifted, i.e., scheduled as late as possible within the scheduling horizon determined by the critical chains. The software selects the chain 3–6–7 as the critical chain.

If the user relies on PS8 for generating the baseline schedule, the optimal seven-period schedule of Figure 2 results, with the critical chains 2–6–7, the chain 2–6–8–3 (8 and 3 have switched position), and the chain 5–4. The software selects the critical chain 2–6–7.

In short, four different baseline schedules have been obtained for the same project, ranging in duration from seven to nine periods, and each providing the planner with a different population of critical chains. Goldratt (1997) contends that it does not really matter which critical chain is chosen. This simple example proves the fact that generating a good baseline schedule does matter, and that statements such as "the critical chain is never ambiguous" (Leach, 2001b) must be interpreted with sufficient care. The length of the baseline schedule, the set of candidate critical chains, and the activities in the chosen critical chain all depend on the procedure used for generating the baseline schedule.

Adding a 50% integer length PB (according to CCS/BM logic) to the baseline schedules of Figures 2, 3 and 4 would inspire management to make the customer promise to deliver the project in at least 11, 14 or 12 periods, respectively. These should be interpreted as "lower bounds:" The

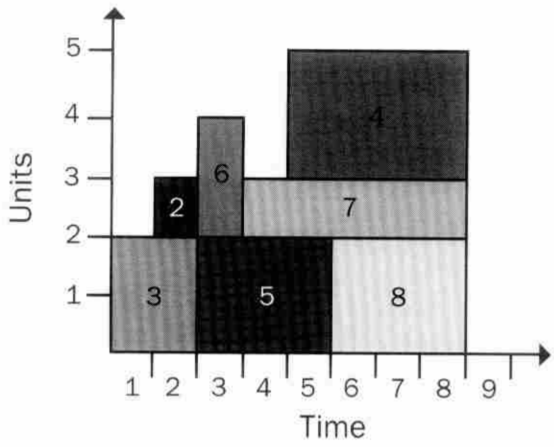


Figure 4. Schedule Obtained Using ProChain

insertion of FBs may well lead to larger committed project durations. If one fails to use the minimum duration baseline schedule as the starting point for buffering and negotiating the project due date with the customer, one may well lose the contract.

Buffer Insertion and Management

CCS/BM argues that safety time should best be eliminated from individual activity duration estimates and aggregated in the form of buffers at crucial locations in the baseline schedule. The buffers should offer protection against statistical variation and should act as safeguards that provide a proactive protection mechanism. The experiment of Herroelen and Leus (2000) revealed that the 50% rule for sizing the buffers may lead to a serious overestimation of the required PB size and, consequently, of the project duration committed to the customer.

By buffering the baseline schedule so that the completion time of the last positioned critical chain activity is scheduled to occur much later than the completion of any other noncritical chain of activities, CCS/BM avoids the cumbersome task of performing generalized PERT calculations (Elmaghraby, 1977) to make reliable predictions about project completion in the presence of resource constraints. The FBs decouple the critical chain from the remaining activities in the network, and as such they help management focus and set realistic due dates. As a result, there is a subtle difference between the interpretation of a FB and the critical path notion of activity float.

On the other hand, the management of the buffers during project execution does not notably differ from float management in the absence of resource constraints. As was already noted by Wiest (1964), however, we must recognize the conditional nature of slack when resources are limited: Slack values are associated with a particular schedule. Moreover, the rescheduling policy adopted also will influence activity slack [refer to Bowers (1995), who works in an environment where resource allocation remains unchanged].

When the FBs are inserted by pushing back their feeding chains, the critical chain may no longer be the longest chain in the network. This has serious implications for the projected schedule, which is unbuffered and in which every gating task (task without real predecessors) is set to start at its scheduled time in the baseline schedule and the roadrunner mentality dictates the start times of the other activities. It may well be that the critical chain, the most important chain of activities that is supposed to determine the duration of the project, is started later than noncritical chain activities. This is a rather counterintuitive way of work.

Also, gaps may be created in the critical chain of the buffered baseline schedule. CCS/BM offers a pragmatic response to these problems in that the critical chain simply is defined as the longest chain before the buffers are inserted (Product Development Institute, 1999). Last, inserting the FBs may lead to immediate resource conflicts. These conflicts must be resolved with sufficient care to avoid an unnecessary increase of the length of the baseline schedule.

While the buffer protection idea certainly is defendable and widespread within scheduling practice, the practical implications should be understood well. First, project management should be sufficiently alert for the idiosyncrasies of the project planning software package in use and should interpret the buffered schedules generated by commercial software packages with extreme care. Figure 5 shows the buffered baseline schedule generated by a commercial software package with integrated CCS/BM scheduling logic for the project of Figure 1. The planned project duration is 13 periods. The critical chain activities 2–6–7 are kept in series, but the chain exhibits a one-period gap.

Figure 6 shows the buffered baseline schedule obtained for the same project with an add-in commercial software package. The planned project duration also is 13 periods, but now management is invited to focus on a different critical chain 3–6–7, which is no longer a chain. The critical chain activities are not kept in series. Critical chain activity 3 is shifted forward and is scheduled in parallel with critical chain activity 7.

FBs should be inserted with sufficient care or an unwanted serious increase in the length of the baseline schedule may well be the result. Rather than using the CCS/BM textbook approach to simply shift activities back and forth until resource contention is resolved, the authors recommend treating the buffers as artificial activities and rescheduling the network. Also, the buffer protection mechanism may fail to be proactive. To illustrate the first point, consider the project shown in Figure 7. The numbers above each node again denote the activity duration, and the numbers below each node represent the renewable resource requirement per period for a single resource type with a constant availability of three units per period. As such, activity 5 has a duration of

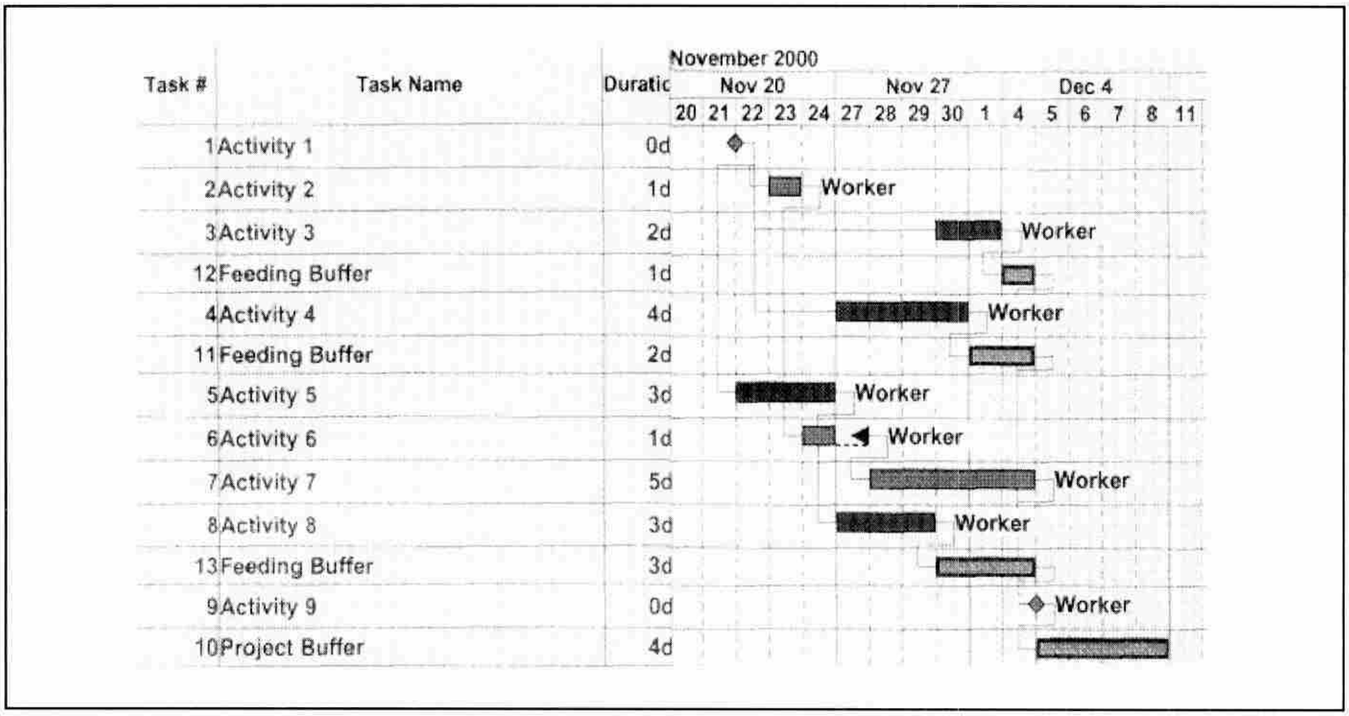


Figure 5. Buffered Baseline Schedule Generated by an Integrated Commercial Software Package

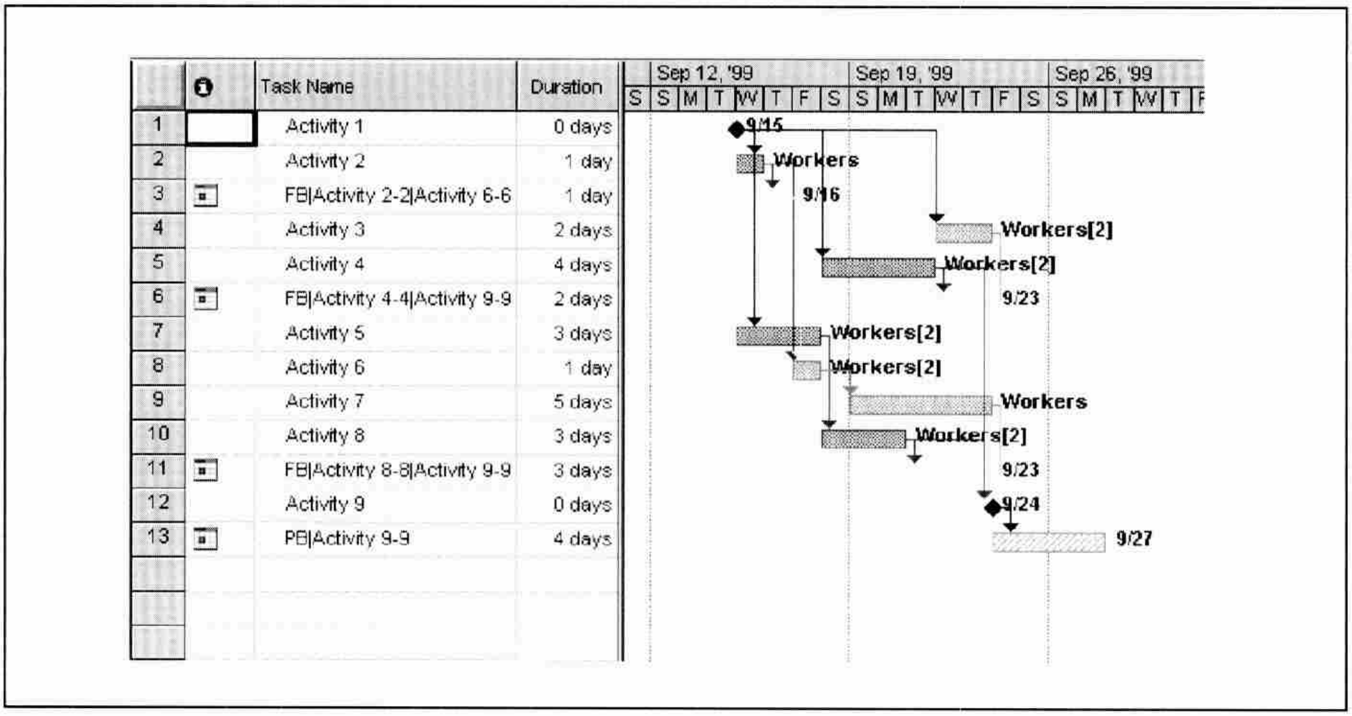


Figure 6. Buffered Baseline Schedule Generated by a Commercial Add-in Software Package

five periods and requires two units of the renewable resource type during each of its execution periods.

Figure 8(a) shows a minimum duration schedule for the project. The critical chain is identified as chain 1–2–3–8 with a 17-period duration. As indicated in Figure 8(b), three FBs must be inserted: The buffer FB5 should be inserted after activity 5, the buffer FB6 should be inserted after activity 6, and the buffer FB7 should be placed after activity 7. If a person applies the 50% rule for buffer sizing and assumes integer length buffers, FB5 should have a length of four periods, FB6 should have a length of three periods, and FB7 should be two periods long.

If the buffers are inserted by pushing the chain of activities feeding the FB backwards in time, immediate resource conflicts may occur. If activity 5 is pushed backwards, an immediate resource conflict is created with activities 6 and 7. Shifting activity 7 backwards does not create sufficient room for activity 5. Activity 4 is a predecessor of both activities 5 and 6. Shifting the chain of activities 4–6 backwards in time does not release sufficient resources for the placement of activity 5, which may be allowed to jump over activity 6, but cannot jump over its predecessor activity 4. Shifting backwards the chain of activities 4–5, which feeds FB5 does not help either. The manager is forced by the resource constraint to increase the project duration by almost 50%, as shown in Figure 8(b).

If, however, the buffers are treated as dummy activities with a positive duration and a minimal duration schedule is computed, keeping the critical chain unchanged, the schedule of Figure 8(c) results. It has exactly the same length as the original baseline schedule of 8(a) and only differs in the planning of the noncritical chain activities. Sufficient room is available for the proper insertion of the three feeding buffers.

The fact that the penetration of FBs may instantaneously lead to resource conflicts that prohibit the FBs from acting as a proactive warning mechanism can be illustrated on the example network shown in Figure 9(a). The numbers above each node again denote the corresponding activity durations. Six resource types are used in this project. Resource types A, B, C, D, and F have a constant availability of one unit per time period, while resource type E has a constant availability of two units per time period. The activity resource requirements are indicated below each network node.

The buffered baseline schedule is shown in Figure 9(b). The critical chain consists of the activities 1–4–8. Assume now that during project execution, activity 2 is the subject of a very small delay. This delay will cause a penetration of FB3–4. Given the small delay, this penetration will not exceed the first third of the buffer, so that no management action is required. However, the delay also will lead to an immediate resource conflict with activity 6, which requires the same resource B. Delaying activity 6 will not just lead to a penetration of FB 7–8. While this penetration will be too small to call for management action, it also will lead to a resource conflict with critical chain activity 4, which requires the same resource B. The FBs do not protect the critical chain from the merge bias. The result is an immediate delay in the critical chain. This will be the case even if a resource buffer (wake-up call) for resource B would be present in front of activity 4. Clearly, FB 3–4 and FB 7–8 fail to act as a warning mechanism in anticipation of future difficulties.

The FBs used by CCS/BM are time buffers. Time buffers may fail to cushion resource conflicts. Such immediate resource conflicts must be resolved, possibly requiring actions to repair the schedule. In this simple example, the required schedule repair actions are rather trivial. This no longer will be the case for real-life project network structures in which multiple activities and multiple resource types may be involved.

Buffer sizes may very well be updated as the project progresses. The PB provides an estimate of the project completion time that can be realized with an appropriate probability and, hence, an estimate of the project due date. As shown in the experiments conducted by Herroelen and Leus (2001), it may be beneficial to update the size of the PB during project execution, as a reaction to the dynamic process of reevaluating the chances of meeting the due date. A related idea would be to shift the alert thresholds of buffer consumption backward in time, assessing the consumption of buffers relative to project or critical chain completion.

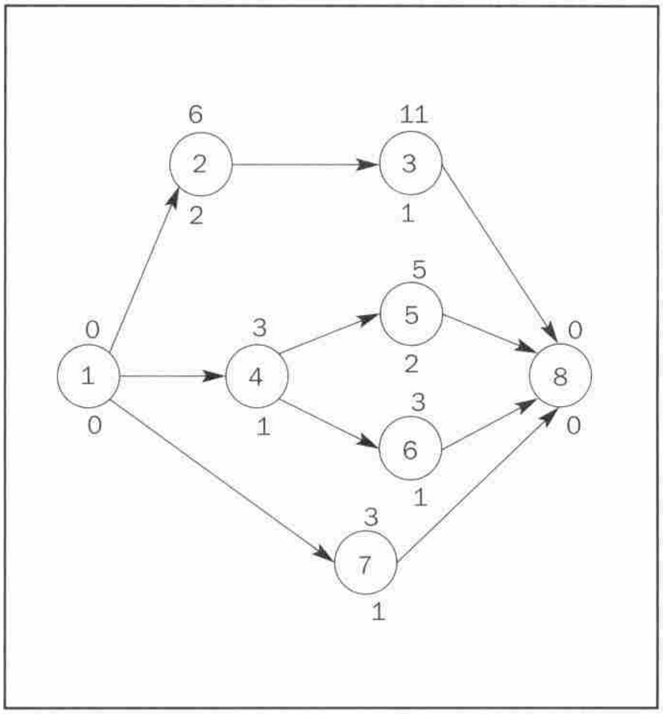


Figure 7. Example Network

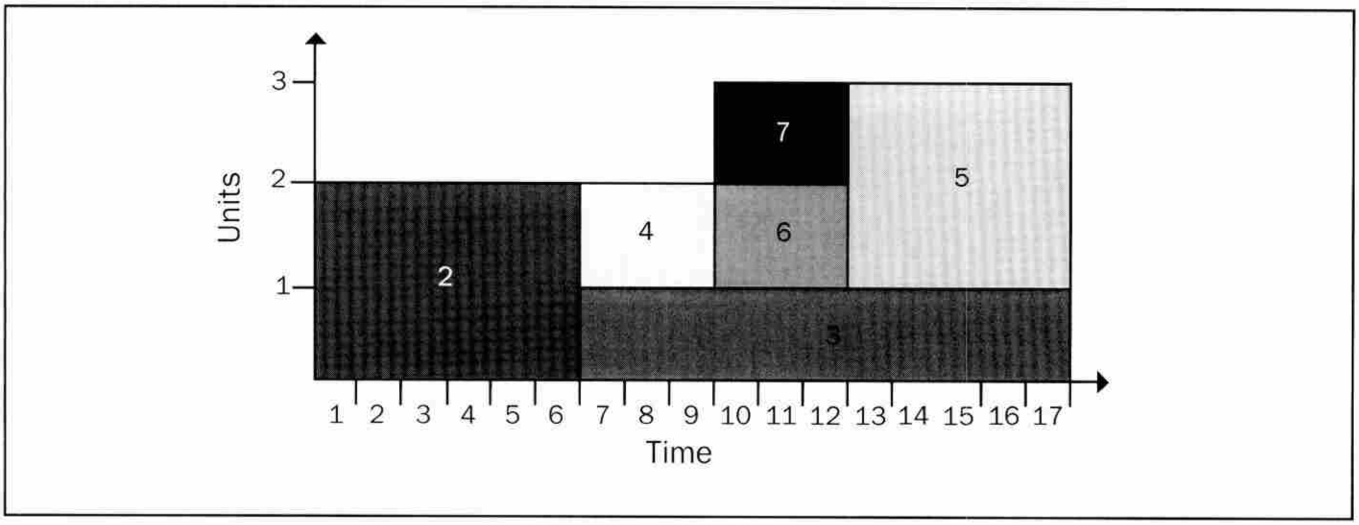
Patrick (2001) discusses various techniques in depth in messages of the Yahoo discussion group. His preference (and the authors') is recalculation of the PB size:

> "One advantage of this approach is that rather than depend on some arbitrary percentage of the buffer ... the threshold is determined not at the launch of the project but in a 'just-in-time' manner that allows the process to take into consideration new information encountered during the execution of the project."

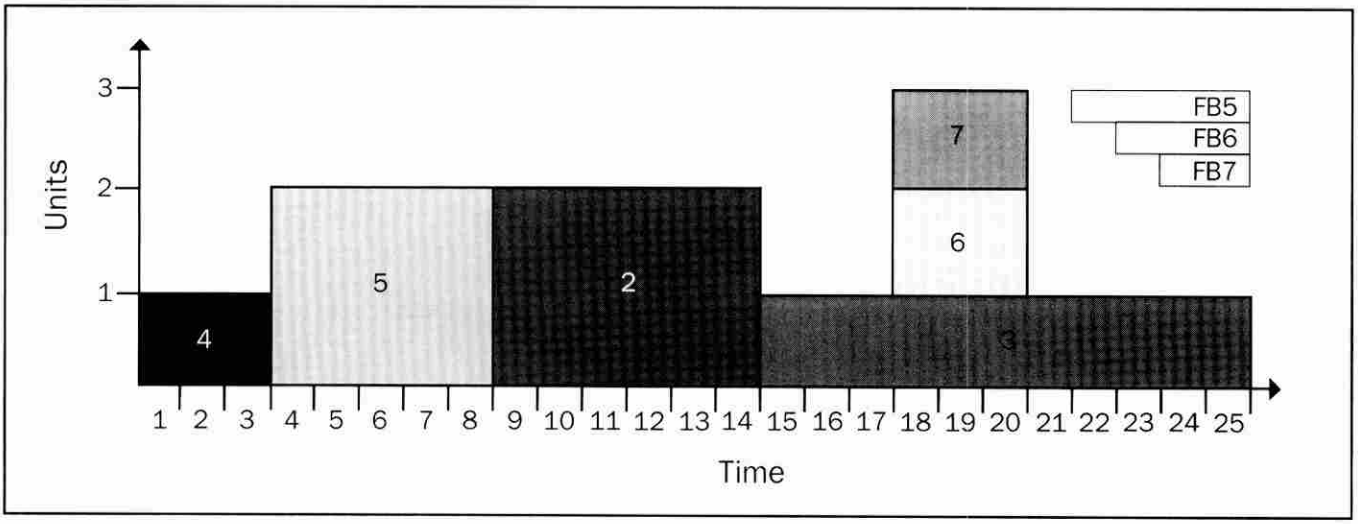
Rescheduling and Stability

Consider the buffered schedule of Figure 8(c). If activity 7 were to take zero time, or activity 2 were to take three time units more than anticipated, activity 6 could be repositioned before activity 5. In general, it makes sense at every project status update to check for opportunities to speed up the projected schedule by rearranging jobs. This is usually done implicitly by available software packages, but it is interesting to explicitly recognize the option.

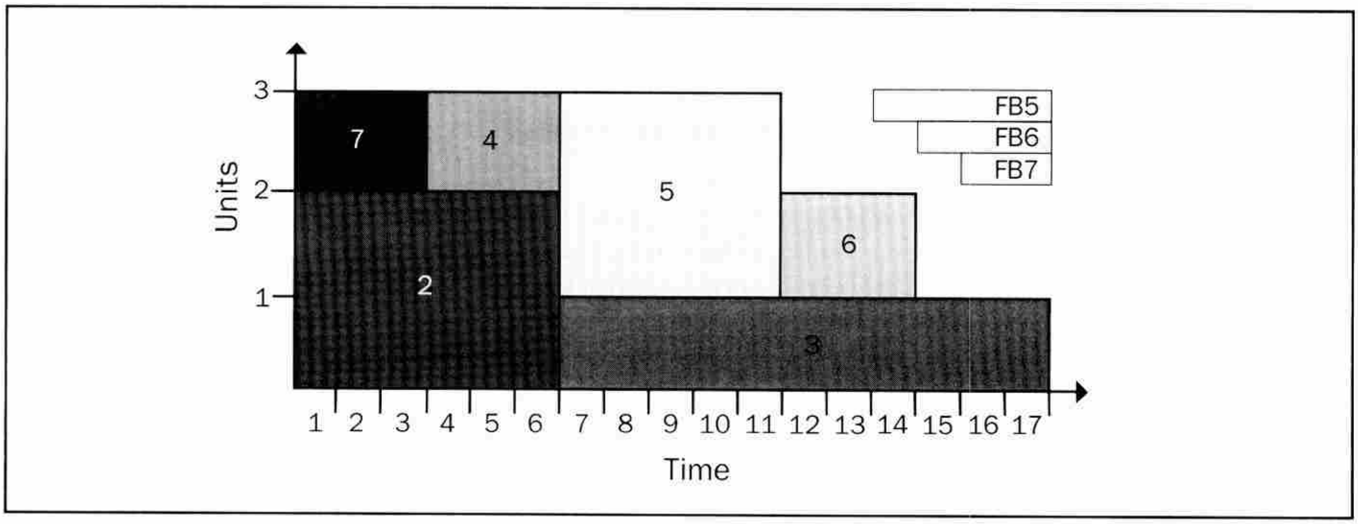
The example project in Figure 10(a) assumes that all the activities have the same unit duration. Activities F and H cannot be scheduled in parallel because they compete for the



(a) Minimal Duration Schedule

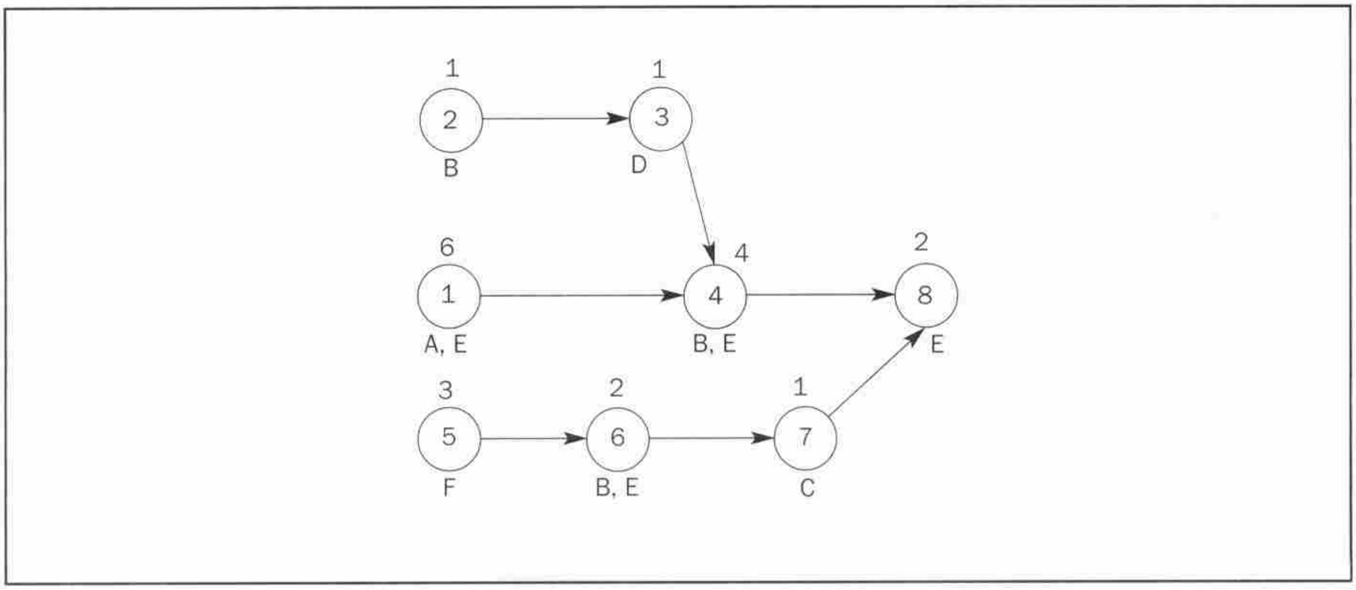


(b) Buffered Schedule Using Push-back Method

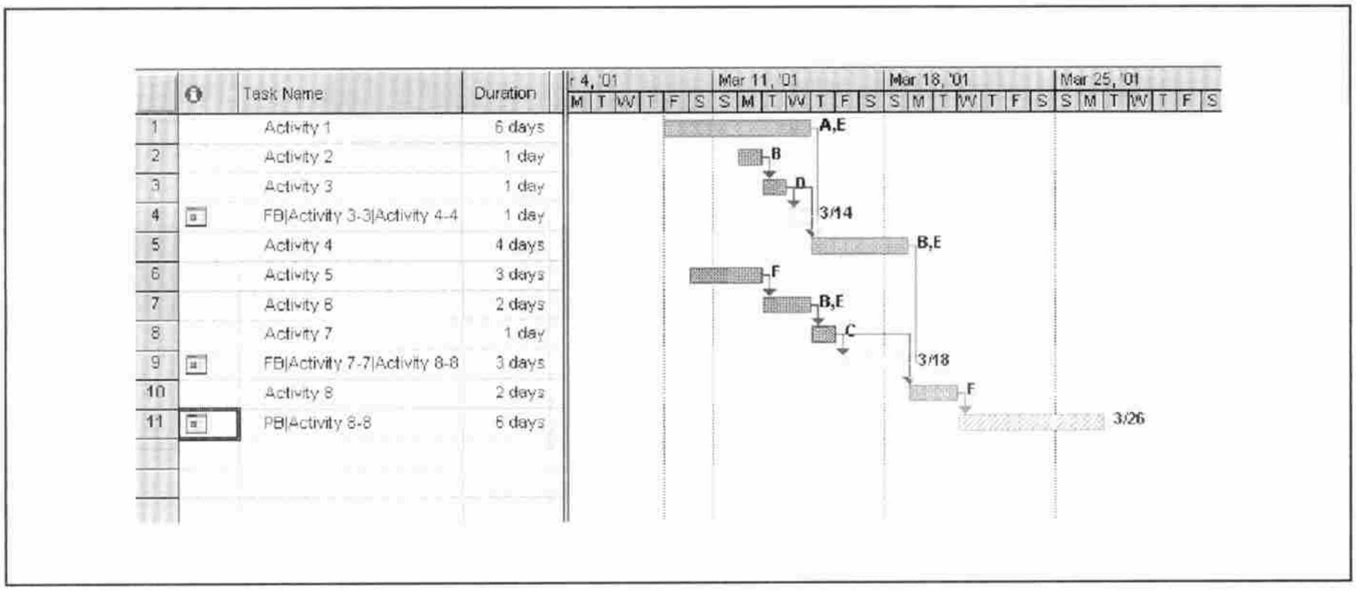


(c) Optimal Buffered Schedule

Figure 8. Insertion of Feeding Buffers



(a) Project Network Example



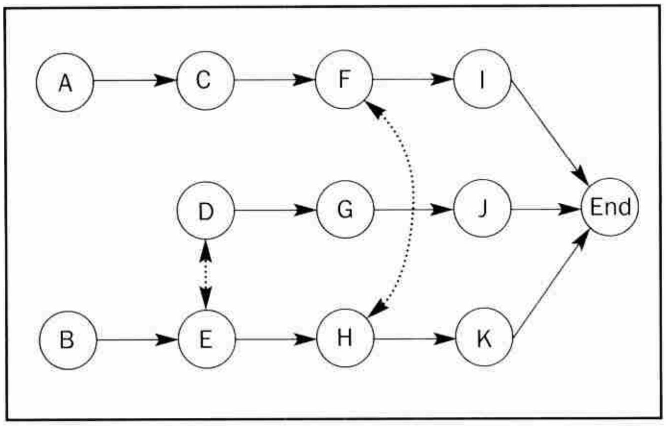
(b) Optimal Buffered Schedule

Figure 9. Project Network and Corresponding Buffered Baseline Schedule

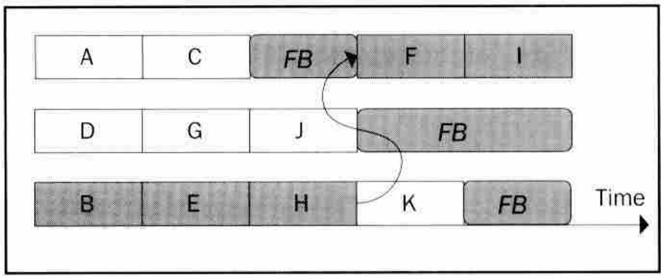
same resource with unit availability; the same holds for D and E. Resource usage of the other activities is not restrictive. One possible critical chain would be B–E–H–F–I. The corresponding buffered baseline schedule is shown in Figure 10(b). The unbuffered projected schedule is depicted in Figure 10(c).

Suppose that at the foreseen end of activity D, this activity is perceived to take longer than expected. Keeping the critical chain unchanged would yield the new projected schedule shown in Figure 11(a). Clearly, the makespan of the projected schedule has increased.

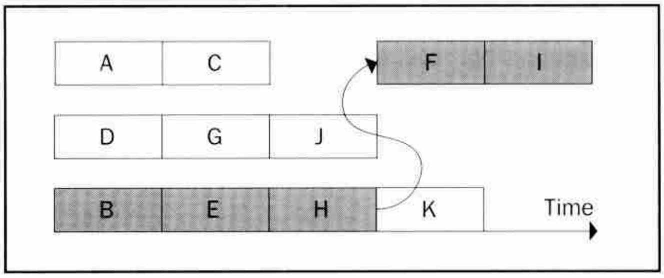
Assume that the critical chain no longer is required to remain in series. In this case, it no longer is necessary to schedule activity H in front of activity F. The result is the projected schedule shown in Figure 11(b), which has a smaller makespan. However, the new projected schedule actually is an implicit recognition of a new critical chain. It would be logical to identify this chain and insert/resize the buffers accordingly. This also allows full exploitation of the updated PB size in making reliable project completion time estimates. Similar observations were made as a result of the simulation experiments in Herroelen and Leus (2001): If the critical chain must be kept in series during project execution, the makespan can be improved by regularly reevaluating the "tightness" of the critical chain, or in other words, one should avoid concentrating on an *ex ante*-derived critical chain that is no longer the real constraint on project makespan.



(a) Example Project



(b) Buffered Baseline Schedule



(c) Projected Schedule

Figure 10. Project Example and Associated Baseline and Projected Schedule

CCS/BM recommends that the baseline schedule and the selected critical chain should not change during project execution (except for very fundamental disruptions that consume the protection offered by the PB), motivated by the idea that rescheduling and changing the critical chain may lead the project team into losing focus. Leach (1999, 2001a, 2001b) illustrates the point through an analogy with the funnel experiment conducted by Deming (1982) to illustrate the differences between common cause and special cause variation. Common cause variation refers to a cause that is inherent in the system. Special cause variation refers to a cause that is specific to some group of workers, a particular production worker, a specific machine, or a specific local condition (Leach, 1999). Special cause variation requires management action, while common cause variation does not.

The experimental set-up used in the funnel experiment consists of a target on the floor, a funnel on a movable stand, and a marble to drop from the funnel. The aim of the system is to cause the marble to land on the center of the target. In the context of project scheduling, the target can be thought of as the committed project completion date and cost. The funnel experiment shows that adjusting a stable process to compensate for an undesirable result or an extraordinarily good result will produce output that is worse than if the process had been left alone. Moving the funnel in the opposite direction whenever the marble misses the target [Leach (2001a) relates this to expediting a newly identified critical path], or moving the funnel back to zero each time before making the adjustment [Leach (2001a) relates this to updating the project baseline schedule with a change or adjusting the plan to actual before taking the management control action] leads to an increase in variation. Leach (2001a) claims that these actions boil down to an adjustment for statistical fluctuation, or common cause variation, instead of real change, or special cause variation.

There is, however, a subtle difference between the parameter setting of a univariate or bivariate statistical process and the rather combinatorial nature of project scheduling, even with variable activity durations: The analogy does not hold. In many companies, rescheduling is undesirable, but mostly because of organizational choices for coordinating resources, rather than because of underlying statistical intuition, as is the case in the funnel experiment. Statistical process control is about observing the same statistical process multiple times (production environment), whereas every single activity is executed only once in a project, which has by itself a unique character. This observation makes any intimate comparison between the two settings faulty.

In addition, the closest equivalent obtainable in the funnel experiment to receiving new information about activity durations would not be the case in which one tampers with the settings of the funnel after one observation. Rather, it is the situation in which one adjusts the location of the funnel because the variability of the output is reduced (the project moves toward completion) and the location of the funnel can be better positioned unambiguously. This is because project activities are precedence and resource dependent. During the execution of a project, the occurrence of disturbances changes the structure of that part of the project that remains to be executed: They provide additional information. This information should not be treated only as the result of statistical fluctuation but should be interpreted as special cause variation that may require management action.

Opportunities for speeding the remaining part of the ongoing project may be exploited by rearranging the schedule. Discarding from the outset information that will allow the remainder of the schedule to be completed fastest may not always be wise. The factorial experiment performed by Herroelen and Leus (2000) confirms that—whether we like it from a practical project management viewpoint or not—regularly recomputing the baseline schedule and updating the critical chain leads to significantly smaller realized project durations. Jorgensen (1999) also has discussed this reactive type of approach in a more theoretical setting.

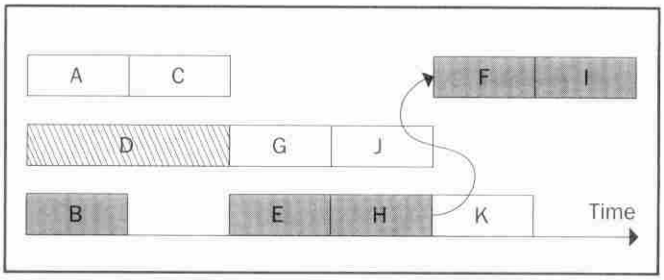
Some authors argue that workers will think management is unqualified and only relies on improvisaton if it routinely reschedules the baseline. Note, however, that the projected schedule, which indicates when workers are to start their next job, is updated every time project status updates occur. Herroelen and Leus (2001) clearly show that this mechanism is most similar to a dispatching approach. At times, it may be a different critical chain that determines the projected schedule updates and job dispatching; this situation should not upset workers more than with an unchanged baseline schedule, especially if project accelerations are achieved in this way.

On the other hand, there are a number of cases where rescheduling is not desirable. Dispatched jobs must not be interrupted normally due to shifting priorities. Such decisions will be counterproductive and introduce system nervousness, and it might be difficult to transfer expert staff and special resources between activities at short notice (Bowers, 1995).

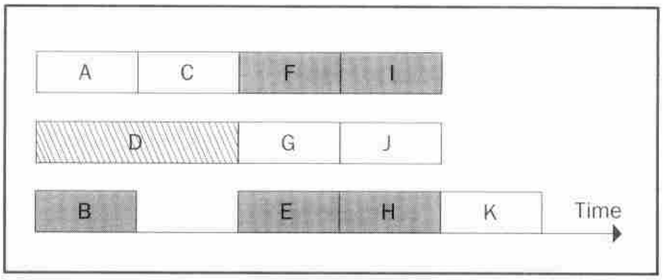
Schedule stability will be required as a result of *ex ante* imposing limits on certain activity start and end times to coordinate resources, for instance, across multiple projects. This is one reason why piloting a single project according to the CCS/BM methodology in a multiproject organization will demand extra care. It may be necessary in a multiproject environment to make advance bookings of key staff or equipment to guarantee their availability (Bowers, 1995). Other sources of the need for such stability can be hard delivery dates for suppliers or subcontractors, or in a larger sense, a hard due date for project deliverables, e.g., milestones—in other words, any time restriction that is external to the project itself. Remember that it is standard CCS/BM procedure to introduce a separate PB per deliverable (ProChain, 1999). If supplier delivery dates or "ready times" are tight or due dates are not, the plan may have to be created backward from or within those dates.

If the due dates are tight on the other hand (after exhausting all possible "activity crashing" alternatives), a baseline schedule loses its significance, and implicit buffer sizes follow from comparing the projected schedule with the imposed dates; there is no real distinction anymore between buffer management and "classical" float management.

In conclusion, the decision to repair the schedule or engage in rescheduling requires extreme care. Simple, clear-cut answers do not seem to exist. This being said, the projected schedule will require regular adaptation anyway to account for resource contention. Also, proponents of unconditionally keeping the critical chain unchanged should ask themselves whether they refuse to revise the baseline schedule because it introduces system nervousness, or rather because such schedule management in the CCS/BM way becomes rather arduous. Reevaluating the most critical part of the schedule in such situations regularly boils down to completely repositioning the buffers and the critical chain. For instance, Wilkens (2000, p. 69) states:



(a) Projected Schedule Keeping Critical Chain in Series



(b) Projected Schedule With New Critical Chain

Figure 11. Effect of a Duration Disturbance

> "Feeding buffers are another story altogether, simply because they are a pain in the neck to manage for all but very small projects. As work is replanned to incorporate changes and work around plans, the buffers need to be relocated and adjusted."

Clearly, the need for intelligent scheduling/repair mechanisms remains, and additional research is needed in the development of powerful mechanisms for the creation of robust baseline schedules and the deployment of effective proactive warning mechanisms.

Conclusions

The CCS/BM methodology has acted as an important eye-opener in project management practice. It correctly recognizes that the interaction between the time requirements of the project activities, the precedence relations defined among them, the activity resource requirements, and the resource availabilities has a crucial impact on the duration of a project. This point, largely unrecognized by many project management practitioners and blind critical path fanatics, is well taken and completely in line with the profound insights gained in the resource-constrained project scheduling literature. Given the complexity and the moderate research progress of stochastic resource-constrained project scheduling, the idea of constructing and protecting a mean activity duration-based precedence and resource-feasible deterministic baseline schedule is sound. It often is the best thing one can hope for and rely on from a computational point of view.

The methodology of inserting and managing resource, feeding, and project buffers—essentially the application of the

TOC to project scheduling—provides a simple and workable tool for setting realistic project due dates and for monitoring the project during execution. It emerged into project management software, which allows project management to shift from a time-oriented (critical path-based) to a resource-constrained (critical chain-based) view. Daily project management practice was in a desperate need for such a shift in focus.

The danger, however, lies in the oversimplification. McKay and Morton (1998, p. 762) relate: "We believe that most project managers will benefit from a conscious reflection of what Goldratt has written ... However, a danger exists for the project manager who does not understand the necessary preconditions." Indeed, the practical implications of applying CCS/BM must be recognized clearly. Pinto (1999, p. 51) concludes: "My concern lies in the strong potential for many companies, who do not understand both TOC's strengths and weaknesses, to latch on to it as TNBT [The Next Big Thing, meaning a management fad] regardless of their particular and unique circumstances."

Resource-constrained project scheduling problems are hard. Constructing and protecting a baseline schedule is not an easy pursuit. CCS/BM offers a practical, software-supported project scheduling tool. Identifying the critical chain is not easy. Due to the complexity of the resource-constrained project scheduling problem involved, management is forced to rely on heuristic procedures for generating a precedence and resource-feasible baseline schedule for real-life projects.

Different scheduling methods may yield different project durations and/or a different set of critical chains. The procedures used for inserting the buffers may have a strong impact on the project completion times committed to the customer, and commercial software packages may provide ambiguous results. The authors have shown that during project execution, the buffers may fail to act as a real proactive protection mechanism and pointed out that management must face the trade-offs involved in schedule repair and rescheduling. Simple, clear-cut answers do not seem to exist. Further research is needed in the development of workable and effective mechanisms for creating robust baseline schedules and effective mechanisms for schedule repair.

References

Bowers, J.A. (1995). Criticality in resource constrained networks. *Journal of the Operational Research Society, 46* (1), 80–91.

Cabanis-Brewin, J. (1999). "So ... So What?" Debate over CCPM gets a verbal shrug from theory of constraints guru Goldratt. *PM Network, 13* (12), 49–52.

Demeulemeester, E.L., & Herroelen, W.S. (1992). A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *Management Science, 38* (12), 1803–1818.

Demeulemeester, E.L., & Herroelen, W.S. (1997). New benchmark results for the resource-constrained project scheduling problem. *Management Science, 43* (11), 1485–1492.

Demeulemeester, E., Herroelen, W.S., Simpson, W.P., Baroum, S., Patterson, J.H., & Yang, K.K. (1994). On a paper by Christofides et al. for solving the multiple-resource constrained, single project scheduling problem. *European Journal of Operational Research, 76* (1), 218–228.

Deming, W.E. (1982). *Out of the crisis.* Cambridge, MA: MIT Press.

Elmaghraby, S.E.E. (1977). *Activity networks—Project planning and control by network models.* New York: Wiley Interscience.

Elton, J., & Roe, J. (1998). Bringing discipline to project management. *Harvard Business Review, 76* (2), 153.

Globerson, S. (2000). PMBOK and the critical chain. *PM Network, 14* (5), 63–66.

Goldratt, E.M. (1997). *Critical chain.* Great Barrington, MA: The North River Press Publishing Corp.

Herroelen, W., De Reyck, B., & Demeulemeester, E. (1998). Resource-constrained project scheduling: A survey of recent developments. *Computers and Operations Research, 25* (4), 279–302.

Herroelen, W., & Leus, R. (2000). On the merits and pitfalls of critical chain scheduling. *Proceedings of the 1st PMI Research Conference 2000,* Paris, France. Newtown Square, PA: PMI, 283–295.

Herroelen, W., & Leus, R. (2001). On the merits and pitfalls of critical chain scheduling. *Journal of Operations Management, 19* (5), 559–577.

Jorgensen, T. (1999). *Project scheduling—A stochastic dynamic decision problem.* Unpublished doctoral dissertation, Norwegian University of Science and Technology, Trondheim, Norway.

Leach, L.P. (1999). Critical chain project management improves project performance. *Project Management Journal, 30* (2), 39–51.

Leach, L.P. (2000). *Critical chain project management.* Boston: Artech House Professional Development Library.

Leach, P. (2001a). Putting quality in project risk management—Part 1: Understanding variation. *PM Network, 15* (2), 53–56.

Leach, P. (2001b). Putting quality in project risk management—Part 2: Dealing with variation. *PM Network, 15* (3), 47–52.

Maylor, H. (2000). *Another silver bullet? A review of the theory of constraints approach to project management.* Paper presented at the 7th International Annual European Operations Management Association Conference, Ghent, Belgium.

McKay, K.N., and Morton, T.E. (1998). Critical chain. *IIE Transactions, 30* (8), 759–762.

Newbold, R.C. (1998). Project management in the fast lane—Applying the theory of constraints. Boca Raton, FL: The St. Lucie Press.

Patrick, F.S. (1999). Critical chain scheduling and buffer management—Getting out from between Parkinson's rock and Murphy's hard place. *PM Network, 13* (4), 57–62.

Patrick, F.S. (2001, 2 January). Critical chain project management, program management, TOC and projects [Msg 33]. Message posted to http://groups.yahoo.com/group/criticalchain

Patterson, J.H. (1984). A comparison of exact procedures for solving the multiple constrained resource project scheduling problem. *Management Science, 30* (7), 854–867.

Pinto, J.K. (1999). Some constraints on the theory of constraints—Taking a critical look at the critical chain. *PM Network, 13* (8), 49–51.

ProChain Solutions Inc. (1999). *Introduction to ProChain.* Retrieved 19 February 2002, from www.prochain.com/university.asp

Product Development Institute. (1999). *Critical chain tutorial.* Retrieved 23 September 2002, from http://www.pdinstitute.com/tutorial/contents.html

Rand, G.K. (2000). Critical chain: The theory of constraints applied to project management. *International Journal of Project Management, 18* (3), 173–177.

Umble, M., & Umble, E. (2000). Manage your projects for success: An application of the theory of constraints. *Production and Inventory Management Journal, 41* (2), 27–32.

Wiest, J.D. (1964). Some properties of schedules for large projects with limited resources. *Operations Research, 12* (3), 395–418.

Wilkens, T.T. (2000). Critical path, or chain or both? *PM Network, 14* (7), 68–74.

Zalmenson, E., & Zinman, E. (2000). *People oriented project management.* Retrieved 23 September 2002, from http://users.actcom.co.il/cyberman/articles/People_Oriented_PM.doc

Zalmenson, E., & Zinman, E. (2001). *Theory of constraints analysis.* Retrieved 23 September 2002, from http://users.actcom.co.il/cyberman/articles/toc_eng.doc

Willy Herroelen, PhD, is professor in operations management at the Department of Applied Economics at the Katholieke Universiteit Leuven. He teaches courses in the area of production scheduling and sequencing, facilities planning, organization of the production process, and project management. He has published widely on resource-constrained project scheduling. His current research activities focus on scheduling and sequencing under various resource and capacity constraints. He is a member of the Project Management Institute and several professional management science and operations management associations.

Roel Leus holds a commercial engineering degree from the Department of Applied Economics at the Katholieke Universiteit Leuven. He is research assistant of the Fund for Scientific Research—Flanders at the same department where he is preparing a PhD dissertation on robust and reactive project scheduling. He is a member of the Project Management Institute and Institute for Operations Research and the Management Sciences.

Erik Demeulemeester, PhD, is professor in operations management at the Department of Applied Economics at the Katholieke Universiteit Leuven. He teaches courses in quality management and production management. He has published widely on resource-constrained project scheduling. His current research activities focus on scheduling and sequencing under various resource and capacity constraints. He is a member of the Project Management Institute and several professional management science and operations management associations.

Statement of Ownership, Management and Circulation

1. Publication Title: *Project Management Journal*
2. Publication Number: 8756-9728
3. Filing date: 12/01/02
4. Issue Frequency: Quarterly
5. Number of Issues Published Annually: 4
6. Annual Subscription Price: $14.00
7. Complete Mailing Address of Known Office of Publication: PMI Publishing Division, Four Campus Boulevard,Newtown Square, PA 19073–3299 USA
8. Complete Mailing Address of Headquarters or General Business office of the Publisher: Project Management Institute, Four Campus Boulevard, Newtown Square, PA 19073–3299 USA
9. Full Names and Complete Mailing Addresses of Publisher: Linda Cherry: Four Campus Boulevard Four Campus Boulevard, Newtown Square, PA 19073–3299 USA
10. Owner: Project Management Institute, Four Campus Boulevard, Newtown Square, PA 19073–3299 USA
11. Known Bondholders, Mortgagees, and Other Security Holders Owning or Holding 1% or More of Total Amount of Bonds, Mortgages, or Other Securities: None
12. For completion by nonprofit organizations authorized to mail at special rates. The purpose, function, and nonprofit status of this organization and the exempt status for federal income tax purposes: Has Not Changed During Preceding 12 Months
13. Publication Name: *Project Management Journal*
14. Issue Date for Circulation Data Below: November 2002

15. Extent and Nature of Circulation

		Average No. Copies Each Issue During Preceding 12 Months	Actual No. Copies Single Issue Published Nearest to Filing Date
a.	Total No. Copies (Net Press Run)	93,894	100,275
b.	Paid and/or Requested Circulation		
	(1) Sales Through Dealers and Carriers, Street Vendors, and Counter Sales (Not Mailed)	Does Not Apply	Does Not Apply
	(2) Paid or Requested Mail Subscriptions (Include Advertisers' Proof Copies/Exchange Copies)	92,794	99,175
c.	Total Paid and/or Requested Circulation [Sum of 15b(1) and 15b(2)]	92,794	99,175
d.	Free Distribution by Mail (Samples, Complementary, and Other Free)	Does Not Apply	
e.	Free Distribution Outside the Mail (Carriers or Other Means)	Does Not Apply	
f.	Total Free Distribution (Sum of 15d and 15e)	Does Not Apply	
g.	Total Distribution (Sum of 15d and 15f)	92,794	99,175
h.	Copies Not Distributed		
	(1) Office Use, Leftovers, Spoiled	1,100	1,100
	(2) Return from News Agents	Does Not Apply	
i.	Total [Sum of 15g, 15h(1), and 15(2)]	93,894	100,275
	(1) Percent Paid and/or Requested Circulation (15c / 15g x 100)	98.7%	98.9%

16. This Statement of Ownership will be printed in the December 2002 issue of this publication.
17. Name and Title of Editor, Publisher, Business Manager, or Owner: Linda Cherry

Date: 12/01/02