

# Chapter 10

## Critical Chain/Buffer Management

**Abstract** Resource-constrained project scheduling is often a complex task due to the presence of dependencies between activities and the limited availability of renewable resources. The previous two chapters gave an extensive overview of different techniques to construct such a schedule, taking various scheduling objectives into account. This chapter extends this resource-constrained scheduling approach to a more flexible baseline schedule in order to be protected against unexpected events. The Critical Chain/Buffer Management approach assumes the construction of a resource feasible schedule as discussed in the previous chapters, but incorporates a certain degree of flexibility in the activity start times in order to easily monitor schedule deviations and quickly respond by taking corrective actions to keep the whole project on schedule.

### 10.1 Introduction

There have been a significant number of international high profile projects failing to be delivered on-time and on-budget. The Channel Tunnel project to provide an undersea connection between France and the UK is probably the most well-known example, but undoubtedly, most readers can also think back of smaller scale projects closer to their work environment that failed miserably. A number of undesirable characteristics are associated with many failing projects: budget overruns, compromised project specifications, and missed milestones. Consequently, the three basic dimensions of project success (time, cost and quality) are often in jeopardy. In his successful business novel, “Critical Chain”, Goldratt reasoned that time was more important than cost for project managers (Goldratt 1997). Support for this idea can be found in numerous articles. As an example, a McKinsey study reported in *Business Week* (Port et al. 1990) that a project that is on-time but over-budget by 50% will earn 4% less than an on-budget project. In contrast, the study predicted that a project that is on-budget but 6 months late will earn 33% less than an on-time project. Both sources support the strategic importance of reducing project time.

Quality or scope of project output (meeting the required specifications) is very much sector and environment dependent and will not be studied separately in this chapter. As usual, the focus lies on a dynamic scheduling perspective and more precisely on the inability of project management to stick to the initially proposed schedule. Budget overruns are clearly visible and attributable. They are often an explicit choice of management to speed up the project (overtime, purchase of extra resources, . . .) or rather the implicit consequences of the project taking more time than initially anticipated, reflecting the financial implications of occupying resources longer than projected. Multiple other sources for budget overruns exist, e.g. the necessity of acquiring new equipment because the present turns out to be insufficient for the project needs, but such risks are either completely unforeseeable, or they are predictable and consequently they should be included in the initial budget development. Such project features are often out of management control, or else need only careful attention but not a lot of management action.

Unlike these unforeseeable events, the main topic of this chapter deals with the presence of variability in the project schedule that can be foreseen to a certain extent. This chapter takes a similar view on project variation as has been taken in Chap. 5, but also takes the renewable resources into account. Consequently, the projects need to be scheduled under high complexity and within the presence of uncertainty and can therefore be classified in the fourth quadrant of Fig. 1.4. Throughout the chapter, it is implicitly assumed that the reader bears in mind at all times that actions to influence project schedule performance often risk to have direct implications on the budget as well as on the quality of the project. It is surprising to see that, given the large amount of projects that have finished late during the last decades, management still fails to quote accurate project deadlines. This is problematic, because virtually all organizations use their project plans not only as tools with which to manage the projects, but also as a basis on which to make delivery commitments to clients. Therefore, a vitally important purpose of project plans and of effective scheduling methods is to enable organizations to estimate the completion dates of the corresponding projects. This is particularly true for organizations that serve industrial clients, because such clients regularly have projects of their own, which require the outputs that the supplier organizations agree to deliver.

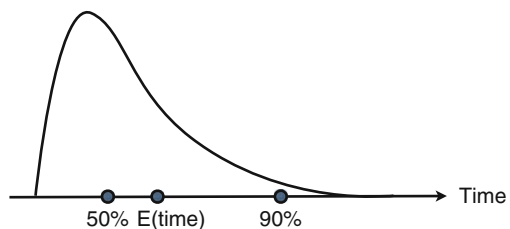
The outline of this chapter can be summarized as follows. Section 10.2 gives an overview of different sources of uncertainty in project management and scheduling. In Sect. 10.3, the main components of the Critical Chain/Buffer Management (CC/BM) scheduling technique is highlighted in detail. Section 10.4 gives a fictitious illustrative example using a six step CC/BM approach. Section 10.5 illustrates how a so-called buffered CC/BM project schedule can be used during the project execution phase to monitor and control the project performance and to trigger corrective actions in case the project deadline is in danger. Section 10.6 gives an overview of the main criticism on the CC/BM approach, highlighting clear merits and showing weaknesses and potential pitfalls. Section 10.7 draws overall chapter conclusions.

## 10.2 Sources of Uncertainty

In a manufacturing environment, machines regularly break down and need to be repaired. This is a random process that can be observed and for which the parameters can be estimated, such that a rather accurate picture of the availability of this machine can be obtained. In most manufacturing settings, such as job shop environments, job routings and machine utilizations are fairly predictable. Since the daily operation is one in which a certain degree of routine reigns, the manager in charge can invoke logic and calculus to estimate average lead times and average system load. A routine project is not like a job submitted to a job shop, but one that is important enough to deserve to be managed separately. As a consequence, averages are not so much important to a project manager as is variability. Unfortunately, because of the unique nature of each project, estimates based on previous experience are often unreliable, if ever previous experience has already been accumulated (compare the rather routine activities involved in the construction of a building with the scheduling effort required for an R&D project). On top of that, people plan and execute projects, not machines or computer programs. Therefore, some insight into human nature is crucial in project management. Every good project manager is equipped with a toolset of human resources management skills.

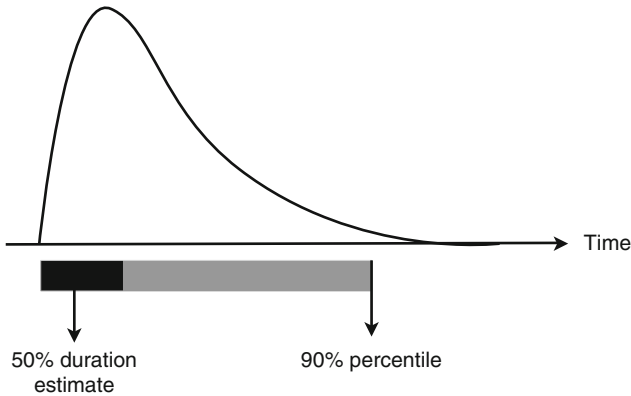
The required input to obtain a deterministic schedule is a single duration estimate for each activity of the project. Nevertheless, this duration is a stochastic variable, which is assumed to be independent of the other activities. Most often, such durations have a probability density function that is skewed to the right, for instance as pictured in Fig. 10.1.<sup>1</sup>

If one asks a programmer in a software development project how much time the development of the component he/she is working on will take, he/she will never select the expected value  $E(d_i)$  or the median (50%-percentile). Rather he/she will mention something in the neighbourhood of the 90%-percentile, such that the duration estimate can be made with a certain amount of safety. Otherwise, in approximately 1 out of 2 cases, his/her programming will finish late (see Fig. 10.2), and this is not at all beneficial to his/her performance appraisal. Of course, the real



**Fig. 10.1** A typical right skewed probability density function

<sup>1</sup>The PERT technique discussed in Chap. 2 also assumes that the duration of a project activity is distributed according to a right-skewed beta distribution.



**Fig. 10.2** 50% time estimate and 90%-percentile

density curve is never known beforehand, so to protect himself/herself even more, the programmer will pad the time estimate even more. As a result, in many project environments, individual activity duration estimates all include a reasonable amount of safety.

Furthermore, setting completion dates is often seen as a negotiation process. In negotiation it is common practice to make an opening bid that allows for cuts later on. Should planners foresee an overall schedule cut, they could be expected to add additional reserve in order to protect their schedules from such a cut. Also, managers at each level of the organizational hierarchy tend to add their own precautionary measures on top of the estimates of managers or coordinators reporting to them. In such a system of arbitrary safety insertion and schedule cuts along the different levels of organizational hierarchy, final projected activity durations will have very little, if any, real value.

### ***10.2.1 Parkinson's Law***

When a project schedule is to be developed based on the estimated single moment durations, some deterministic scheduling algorithm can be invoked, for instance by using the critical path and resource leveling options of standard project management software as discussed in the previous chapters. But what will happen when the software component the programmer is developing is finished in 70% of the estimated time? Most project schedules have milestones associated with activity finish times, meaning that an early finish will not be especially rewarded, but a late finish is undesirable. The worker will probably not pass on the output of his/her programming to the resources assigned to successor activities, but rather start streamlining his/her code, adding extra nice graphical features or so (gold plating

or adding unnecessary bells and whistles). This is an illustration of what is called “Parkinson’s law”.

*Work expands to fill the allotted time (Parkinson’s law)*

The programmer is not rewarded for early finishes, but rather he/she risks seeing future time estimates reduced by a certain factor, because he/she appears to be over-estimating his/her time needs. Also, if he/she hands in his/her outputs early, he/she will probably be assigned new work immediately, and it is more pleasant and less stressful to remain on the initial activity for some time longer. In other cases, people will simply adjust the level of effort to keep busy for the entire activity schedule. As was already mentioned, traditional project environments stress not being late, but they do not promote being early. This environment encourages Parkinson’s law. In many environments, there are still other disincentives to report an activity completion early: work performed on time and material contracts for instance results in less revenue if the work is completed early. If the functional organization completes the work in less time than estimated, they cannot continue to charge the project.

### ***10.2.2 The Student Syndrome***

A second type of undesirable effect that can come into play in standard project management environments, can be nicely described in an academic setting. Consider a course for which the students enrolled have to write a paper and they have a deadline within 3 months from now. The paper itself however would, if worked at full effort and with a reasonable degree of safety included, require no more than 4 weeks. What would be the work planning of any ‘regular’ student? He or she will mostly postpone the real start of research and preparation to only some 4 weeks before the deadline. Undoubtedly, similar behavior can be observed in project management practice. This effect is known as the “student syndrome”: many people have a tendency to wait until activities get really urgent before they work on them.

*Wait until activities get really urgent (student syndrome)*

Both scenarios, Parkinson’s law and the student syndrome, will occur in projects with deterministic schedules with ample safety time built in and where milestones are used to evaluate workers. They will cause the initial duration estimates to become *self-fulfilling prophecies*, at least when activities could hypothetically be completed faster. This implies that although unforeseen disruptions induce delays, there will be no positive schedule variations to compensate for the negative ones. Such delays will also regularly occur exactly because of the student syndrome: when an unexpected problem is encountered when the work is halfway done, all safety is gone already and the estimate will be overrun. This makes it feel like the activity was underestimated to begin with, possibly leading to even higher future estimates.

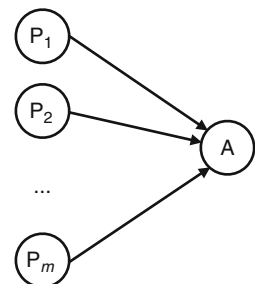
### 10.2.3 Multiple Parallel Paths

If the project network does not simply consist of one simple path of activities, but rather has multiple parallel paths that diverge and join at different places in the network, there is another reason why favorable activity finishes cannot always be exploited, whereas delays often have immediate repercussions on the entire project. This is caused by the predominant use of finish-start precedence relations to model activity networks, which imply that a successor activity can only be started when the latest of its predecessors finishes. This effect is unavoidable, as the type of precedence relation is the most logical choice and models reality in the most natural way. Usually, the path merges tend to concentrate near the end of the project: indeed, “assembly”, “integration” or “test” operations mostly occur close to project completion, requiring many elements to come together. This is one reason why project managers state that “many projects complete 90% the first year, and complete the final 10% in the second year”. Consider Fig. 10.3, in which activity A has an undetermined number of  $m$  immediate predecessors  $P_i, i = 1, \dots, m$ .

If each of the merging paths has a 50% probability of being done by the estimated time, the probability of at least one being late is already almost 88% when three activities merge together. Even if each individual activity had an 85% probability of on-time completion, the probability that at least one is late still approaches 40%. These observations are related to the disadvantages of the application of the classical PERT model and justify the need for more sophisticated simulation or analytical tools when the activity durations can indeed be modeled as independent random variables. The occurrence of multiple parallel paths and the influence on their successor activities is also discussed in Sect. 5.5.2 as the “merge bias”.

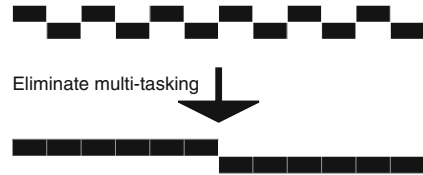
### 10.2.4 Multitasking

A last project management practice that requires attention in this section is that of *multitasking*, which is not that much related to human behavior itself as it is to work organization. Multitasking is the performance of multiple project activities at the same time. In reality, time is divided between multiple activities, for instance by working on one project in the morning and one in the afternoon. Most people



**Fig. 10.3** An assembly activity can only start when multiple predecessors are finished

**Fig. 10.4** Multitasking versus no multitasking



think of multitasking as a good way to improve efficiency: it ensures everyone is busy all the time. However, it has a detrimental effect on activity durations, which is illustrated in Fig. 10.4. Assume two activities that have to be performed by a single resource. The top picture of Fig. 10.4 displays a Gantt chart that represents the case of multitasking: both activities only finish at the end of the scheduling horizon. The bottom part of Fig. 10.4 illustrates the benefits that can be achieved by eliminating multitasking: there is no change in the finish time of activity 2, but activity 1 will be finished in half the time if it is worked on at full effort. Nevertheless, towards management, the worker will at least be able to present progress on all activities. In this example, the influence of set-up times is ignored: each time a worker changes from one activity to another, he/she will need a certain amount of time to handle this change-over, both in case of physical and of intellectual labor.

Single moment time estimates in an environment of multitasking become very much dependent upon the degree of multitasking that will be adhered to during activity execution. If experience from the past is used to develop estimates, those estimates are only meaningful if the same degree of multitasking was present at the time of reference. In most cases, the actually achievable activity duration (at full effort) remains concealed. This reasoning is very similar to the estimation of the lead times of product batches in a manufacturing company: if one looks at lead times achieved in the past to produce a value, he/she should only consider those observations where the company was working under a comparable system load.

Multitasking need not only occur within one project: most of project work is executed on a multi-project basis (as opposed to the use of entirely dedicated resources). Multitasking or jumping between projects could result in a number of negative effects, especially in the case of bottleneck resources. A multi-project environment will pose particular difficulties, because the workers probably report to multiple project managers and will have to comply with the desires of each of them. There is clearly a need for prioritization of projects and reduction of multitasking to an acceptable level. Otherwise, considerable competition for resources among the projects will be created.

### 10.3 Critical Chain/Buffer Management

This section presents an integrated project management methodology that is especially focused on controlling uncertainty during project execution and its undesirable effects discussed in the previous section. The Critical Chain/Buffer

Management (CC/BM) methodology is an application of the Theory of Constraints (TOC). In the end of the 1970s, dr. Eli Goldratt developed a planning methodology and corresponding software under the name OPT (Optimized Production Technology). In the midst of the 1980s, the term OPT was replaced by TOC. TOC offers a structured logic approach to problem solving and applies its brainstorming efforts mainly to the manufacturing environment. The TOC has a management focus on bottlenecks, or constraints, that keep the production process from increasing its output. Once managers identify the bottlenecks, overall operation is planned entirely as a function of the bottleneck schedule. When the whole is as effective as it can be at a given capacity, managers can elevate the constraint by investing extra capacity at the bottlenecks. Once a constraint has been lifted, these steps need to be repeated to identify other emerging constraints. A full overview of TOC is outside the scope of this book. The reader is referred to a brief introduction in Sect. 2.3.2.

### ***10.3.1 Theory of Constraints in Project Management***

Of course, project management texts have long told managers to focus on constraints. For projects, the constraint is perceived to be the critical path, which is the series of activities that determines the minimum time needed for the project to complete (see Part I of this book). Goldratt adds an important second ingredient to this framework that management often overlooks: scarce resources needed by activities both on and off the critical path and possibly also by other projects. In the case of developing a new product, for example, a manager may schedule the different activities according to the pace of the critical path but still face delays because the computer-aided design console is held up by other jobs. The critical chain (CC) is defined as that set of activities that determines the overall duration of the project, taking into account both resource and precedence dependencies. To prevent this critical chain from delays, CC/BM advises managers to build multiple types of safety (time) buffers into the schedule, similar to the inventory buffers used in production lines to make sure that bottleneck machines always have material to work on.

### ***10.3.2 Working Backwards in Time***

A CC/BM schedule is developed backwards in time from a target end date for the project. In the previous chapters of this book, activities have been scheduled as-soon-as-possible (ASAP) from the project start date, as usually done in traditional project scheduling. This scheduling places work as close as possible to the front of the schedule. In CC/BM planning, work is placed as close as possible to the end of the schedule, in an as-late-as-possible (ALAP) fashion. This approach provides advantages similar to those the just in time (JIT) approach offers in a production environment. These benefits include minimizing work-in-progress



(WIP), and not incurring costs earlier than necessary, thus improving project cash flow (under the assumption that only cash outflows are associated with intermediate project activities). Also, possible changes in the scope of the activity (altered client specifications or changes to subsystems interfacing with the activity), imply a higher risk of rework of activities that are started ASAP. Less rework will also result from the fact that workers simply have better information about their assignments. The main drawback directly related to scheduling in an ALAP fashion is that, in traditional critical path terminology, all activities will become critical. Any increase in duration of any activity will result in an equal increase in project end date. As will be explained in detail below, buffers will be inserted at key points in the project plan that will act as shock absorbers to protect the project end date against variations in activity duration. In this way, the benefits of ALAP scheduling are fully exploited with adequate protection against uncertainty.

Consider a project consisting of six activities in series, as shown in Fig. 10.5. The duration of each activity can be modeled as a stochastic variable, for instance with a univariate density function as pictured in Fig. 10.1 (where the variance will vary between the activities). Clearly, an organization’s reputation as a reliable supplier is at stake when it quotes unreliable deadlines to customers, so its project schedules should protect the customers against the variability inherent in the activity durations. However, an overly large protection on the contrary will result in uncompetitive proposals and the loss of business opportunities. In order to cope with this complex task of project deadline estimation, one method of shielding its customers from the effects of the duration variability might be to ensure the timely completion of every individual activity. In fact, the widely accepted method of tracking progress relative to a schedule of milestones is an example of this approach. Choosing a safe time estimate for each activity separately will result in the choice of an approximation of the 90%-percentile estimate of each activity duration separately, resulting in the Gantt chart displayed at the bottom of Fig. 10.5. As discussed earlier, these milestones are self-fulfilling prophecies, so the project will most probably end no sooner than the quoted deadline.

### 10.3.3 The Project Buffer

It is very doubtful that any organization could be competitive in today’s business environment if the organization’s managers attempted to manage variability in this manner. Most managers know this, of course. This is why they struggle with a

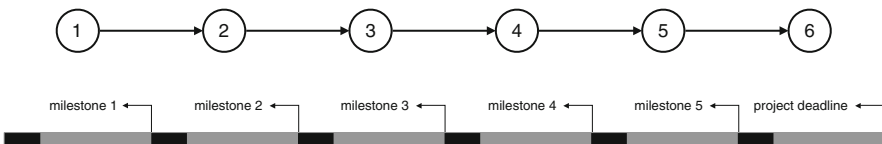
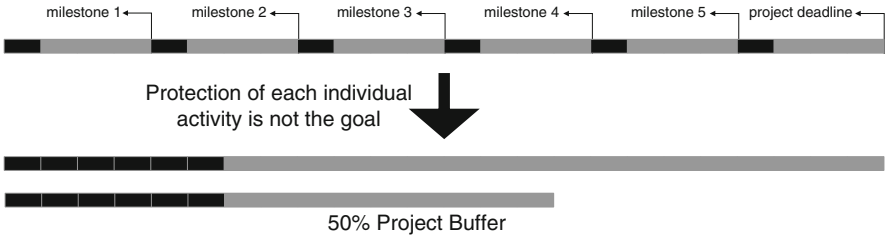


Fig. 10.5 A serial project network with safety time for each individual activity

conflict, between being able to present a competitive proposal to a customer and protecting the same customer from the adverse effects of the inevitable variability in project duration. The basic problem is that, as already mentioned, early finishes are wasted while late finishes are accumulated as the project progresses. In the TOC approach to project management, the seemingly logical protection of the scheduled completion of individual activities is not the goal. Rather, in the spirit of speed-to-market driven project performance, management only desires the rapid and successful completion of the project as a whole. Thus, CC/BM eliminates safety time for individual activities, and aggregates this protection at the end of the project under the form of a project buffer (PB). This implies a review of all activity duration estimates, such that protection against variability is excluded. One could quote the average duration of comparable activities, when they are worked on at full effort, or alternatively, choose a duration that will only be exceeded approximately one out of every two times (the median). The CC/BM approach constructs a project schedule based on so-called *aggressive duration estimates* (average, median, or any other value) that are not (individually at least) padded with safety. The reduction in the activity time estimates to aggressive time estimates also implies that it is essential to execute the project according to the *roadrunner mentality* or the *relay race approach*. This approach forces an activity to start as soon as the predecessor activities are finished. Exactly as in a relay race, the goal is to capitalize on the early finishes of preceding activities. The resulting project schedule based on aggressive time estimates is only an aid to come up with a project deadline, and not to check on individual activity schedule performance (or in other words: there are no milestones for the individual activities).

The removal of the protection from the individual activities must be aggregated into a project buffer PB with an appropriate size. However, since both positive and negative activity finishes will be attained (e.g. 50% estimates), these fluctuations will (partially) compensate for one another along the chain. Consequently, the aggregate protection to be provided at the end of the schedule needs not to be as large as the sum of the removed safety time of the individual activities. This is an intuitive result, but it can also easily be demonstrated mathematically. Assume that all  $n$  activities on a chain have equal variance  $\sigma^2$ . If the safety time of each individual activity is assumed to be equal to two standard deviations, the cumulative safety time will be  $n(2\sigma)$ . The variance of the sum of the durations on the other hand is the sum of variances, so to protect the chain executed according to the roadrunner mentality, the required safety time is  $2\sigma\sqrt{n}$ . The sum of a number of independent random variables tends to a normal distribution (according to the central limit theorem), which implies that the percentage protection provided for the individual strongly skewed distributions is actually even less than for the more normal chain of activities by selecting the same number of standard deviations. The less statistically inclined reader needs not to worry about these details: a valid rough cut approach would be to paste 50% of the removed safety time of each activity into the PB, which is also the method Goldratt proposed in his novel. This 50%-rule results in the reduced PB-size that is represented in Fig. 10.6. A second rule of thumb is the sum of squares or root square error method: the required PB size is set equal to the square root of the



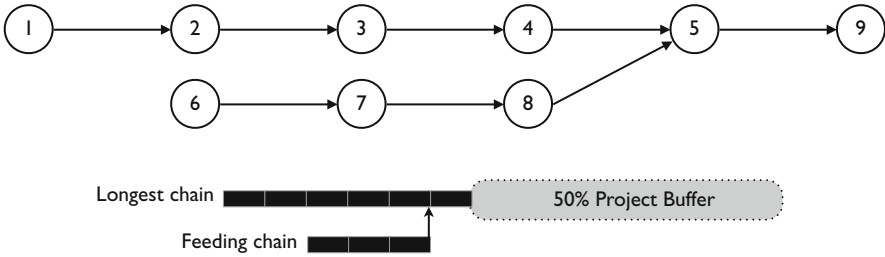
**Fig. 10.6** Inserting a project buffer

sum of squares of the removed safety in the individual activities. This second rule is preferable for projects with a large number of activities, because the 50%-rule will tend to overestimate the required protection in such case. This is because it is a purely linear procedure: a 12-month project could end up with a 6-month PB, a 2-year project could end up with a year-long PB.

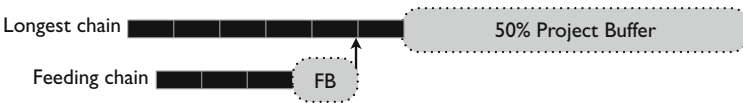
### 10.3.4 Feeding Buffers

The previous section explained how to properly handle individual activities and chains of activities. However, no project consists of a single chain of activities. All projects will have multiple chains in parallel, although mostly, only one will be the longest. The effects of these parallel chains on the variability in the overall duration of a project have been discussed in Fig. 10.3 and should be incorporated in the CC/BM approach. Figure 10.7 shows a fictitious project network with nine activities. The second but last activity of the longest chain (critical chain) with ID=5 is an assembly activity: its start requires the output generated by the first four activities of the longest chain, and also the outputs generated by the so-called feeding chain. The absence of any of these outputs precludes the start of the assembly activity. For the moment, it is assumed that no resource conflicts occur between the different chains, such that they can indeed be executed in parallel, independently of one another. The bottom part of Fig. 10.7 displays a Gantt chart where the feeding chain has been scheduled ALAP, as the CC/BM theory prescribes.

From the discussion of Fig. 10.3, it is known that establishing your project baseline duration projections based on the critical chain alone will yield strongly downwards biased results, and this effect is only increased by our ALAP scheduling. Assume that chain 1–4 has a probability of 50% of finishing at the aggressive schedule duration forecast, and similarly for the feeding chain, then activity 5 will only start on time in one out of every 4 ( $=0.50^2$ ) cases. One mathematically correct way to handle the complication of parallel chains would be to use either simulation or statistical calculations to adapt the size of the PB accordingly, as explained in the schedule risk analysis Chap. 5 without the presence of resources. However, this is where the elegance and simplicity of CC/BM comes in. The PB serves only to



**Fig. 10.7** A project with a feeding chain



**Fig. 10.8** Inserting a feeding buffer

protect the CC itself and the CC is decoupled from all outside (noncritical chain) feeding chains by means of so-called feeding buffers (FB). More precisely, a FB is inserted wherever a nonCC activity feeds into a CC activity. If the 50% rule is used to size the PB and a somewhat smaller (than 50%) FB is inserted, the buffered schedule of Fig. 10.8 will be obtained. Usual practice when multiple chains are interconnected, is to protect only for the longest of all those feeding chains, disregarding the other ones.

### 10.3.5 The Critical Chain

Up to now, the limited availability of renewable resources has been largely ignored. However, the presence of resources often leads to situations where resource conflicts are involved, as shown in Chaps. 7 and 8. Consider the simple project network of Fig. 10.9. Activities 1 and 3 and activities 2 and 4 must be performed in series due to the finish-start precedence relation defined to hold between them. Activity 1 and activity 2 must be performed by the same (renewable) resource X, of which only 1 unit is available. The activity durations are assumed to be aggressive 50% estimates. CC/BM starts by deriving a resource-feasible schedule in which all activities start ALAP (this can be achieved by the “resource leveling” function in standard project management software tools or by the backwards use of the priority rule based scheduling techniques of Sect. 7.4.1). Such an (unbuffered) schedule is depicted in Fig. 10.9. Based on such a schedule, it is easy to identify the CC, defined as the longest chain of activities that considers both technological and resource dependencies: it will be a chain of activities for which the end of each activity equals

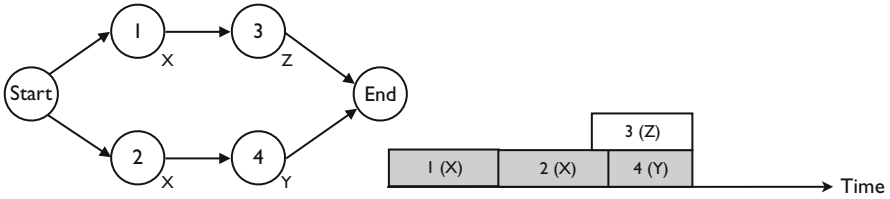


Fig. 10.9 An unbuffered resource feasible schedule (CC = 1-2-4)

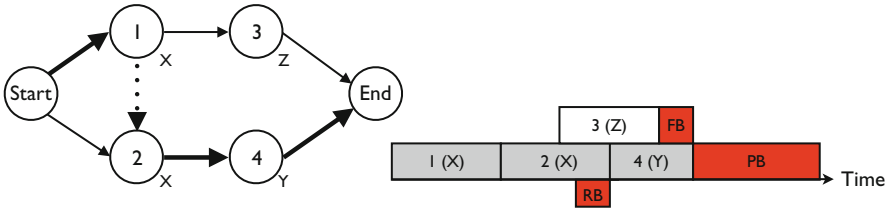


Fig. 10.10 A buffered resource feasible schedule

the start of the next. In the example, the CC is the chain “start-1-2-4-end”. The resource conflict is resolved by forcing activities 1 and 2 to be performed in series, as indicated by the dotted arc in the network in Fig. 10.10. The buffered schedule is shown in the same figure. The resource buffer RB is discussed below.

### 10.3.6 Resource Buffers

One of the leading causes of late projects is that resources are not available or not available in sufficient quantity when they are needed. CC/BM requires a mechanism to prevent the CC activities from starting late or taking longer due to resource unavailabilities (other activities are less important). The selected method is to use a resource buffer (RB) to provide information to the CC resources about when they will be needed. This RB is different from the PB and FBs in that it does not normally occupy time in the project baseline schedule. It is an information tool to alert the project manager and performing resources of the impending necessity to work on a CC activity. RBs are placed whenever a resource has an activity on the CC, and the previous CC activity is done by a different resource. Resource buffers should make sure that resources will be available when needed and CC activities can start on time or (if possible) early. RBs usually take the form of an advance warning, i.e. a wake-up call for every new instance of a resource on the CC. Alternatively, space (idle time) can be created on the resource to provide a kind of protective capacity. An illustration of the placement of a RB is provided in Fig. 10.10: it warns resource Y some time before it is to start working on the CC, that it should be ready.

### 10.4 An Illustrative Example

Having covered all the basic scheduling aspects of CC/BM, a brief summary of the scheduling methodology of CC/BM for deriving the buffered baseline schedule can be outlined as follows:

1. Come up with aggressive estimates.
2. Construct an ALAP schedule.
3. Identify the Critical Chain.
4. Determine appropriate buffer positions.
5. Determine appropriate buffer sizes.
6. Insert the buffers into the schedule.

In the following, these six steps will be applied to a larger example project. The project network is represented in Fig. 10.11. The activity duration is indicated above each activity node while the resource requirements for three renewable resource types are given below the node. Activities 0 and 12 are dummies, representing project start and finish, respectively.

**Step 1.** It is assumed that the activity durations represented in the network are already aggressive 50% time estimates (see Fig. 10.1).

**Step 2.** To construct a resource feasible project schedule, information is needed about the resource requirements of the different activities. In the project, three resource types are used, named A, B and C, with availability of 3, 1 and 2 units, respectively. The resource requirements of each activity are pictured in the Table 10.1. By use of a commercial software tool or scheduling techniques discussed in Sect. 7.4.1, the schedule of Fig. 10.12 can be obtained. In this schedule, all activities are scheduled ALAP.

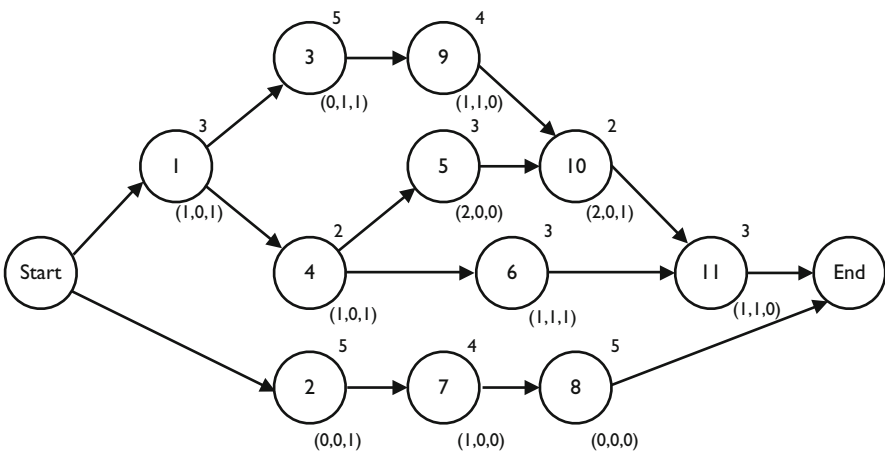
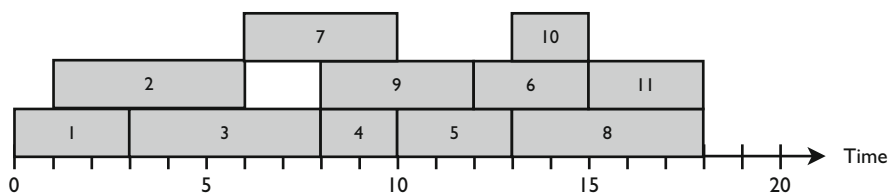


Fig. 10.11 An example project network with 11 nondummy activities

**Table 10.1** Resource requirements for each activity  $i$

$i$	$d_i$	$r_{i1}$	$r_{i2}$	$r_{i3}$
0	0	0	0	0
1	3	1	0	1
2	5	0	0	1
3	5	0	1	1
4	2	1	0	1
5	3	2	0	0
6	3	1	1	1
7	4	1	0	0
8	5	0	0	0
9	4	1	1	0
10	2	2	0	1
11	3	1	1	0
12	0	0	0	0



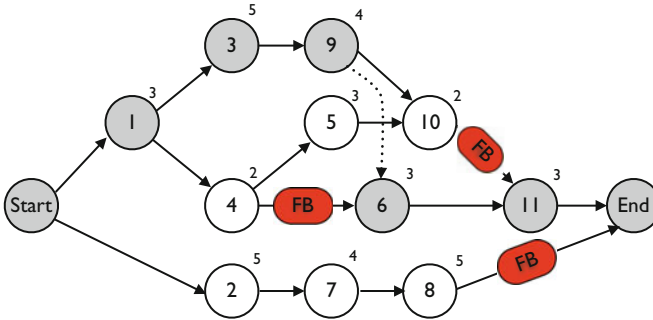
**Fig. 10.12** A resource feasible latest start schedule

**Step 3.** Based on the above schedule, there are three choices for the CC: either “1-3-4-5-8”, “1-3-4-5-10-11” or “1-3-9-6-11”.<sup>2</sup> Based on a project manager’s knowledge of the project environment (subjective!), it can for example be concluded that the third of the three candidate chains is the most constraining: there is only one resource link, and resources are amply available in the company, while the technological precedences are strict. Also, the activities on the second chain are perceived more as “standard” activities, that are better manageable.

**Step 4.** Appropriate buffer positions are indicated in Fig. 10.13, together with the chosen CC.

Three feeding buffers will be inserted, the first between activities 4 and 6, to protect the CC in activity 6 from variability in the noncritical feeding chain consisting only of activity 4 (subsequently referred to as  $FB_{4-6}$ ). A second feeding buffer ( $FB_{10-11}$ ) is inserted before activity 11, to protect it from variability on the feeding chain 4-5-10. Finally,  $FB_{8-12}$  is present to protect the project end from variability in feeding chain 2-7-8. A project buffer will of course

<sup>2</sup>It has been extensively shown in Chap. 7 that the critical chain depends on the algorithm used to construct a resource feasible schedule, and on the scheduling objective, which is assumed to be the minimization of time throughout this chapter.



**Fig. 10.13** The buffered network of Fig. 10.11

also be inserted, after activity 12. Resource buffers should be placed whenever resources are transferred from nonCC to CC-activities. It is left up to the reader to determine the position of these buffers. As RBs will only be implemented as a wake-up call, they will not be further considered in this exercise. In practical settings, it may be wise to wait with the identification of these RBs until the final buffered baseline schedule has been developed, as resources may be planned to be transferred differently in this final schedule.

**Step 5.** Based on studies of similar activity durations of previous comparable projects, the company has estimated that the standard deviation  $\sigma$  of each activity duration is about 0.4 times the duration. Corresponding standard deviation estimates for all activity durations are provided in Table 10.2. Management has decided that a time protection of two standard deviations suffices for buffer sizing. Hence, the following buffer sizes can be calculated:  $FB_{4-6} = 1.6$  time periods  $\rightarrow$  choose 2;  $FB_{8-12} = 2 * \sqrt{2^2 + 1.6^2 + 2^2} = 6.5 \rightarrow$  choose 7;  $FB_{10-11} = 2 * \sqrt{0.8^2 + 1.2^2 + 0.8^2} = 3.3 \rightarrow$  choose 4. In a similar way, the size of the project buffer is equal to  $PB = 2 * \sqrt{1.2^2 + 2^2 + 1.6^2 + 1.2^2 + 1.2^2} = 6.6 \rightarrow$  take 7.

**Step 6.** The FBs and PB can now be inserted into the baseline schedule, as shown in Fig. 10.14.

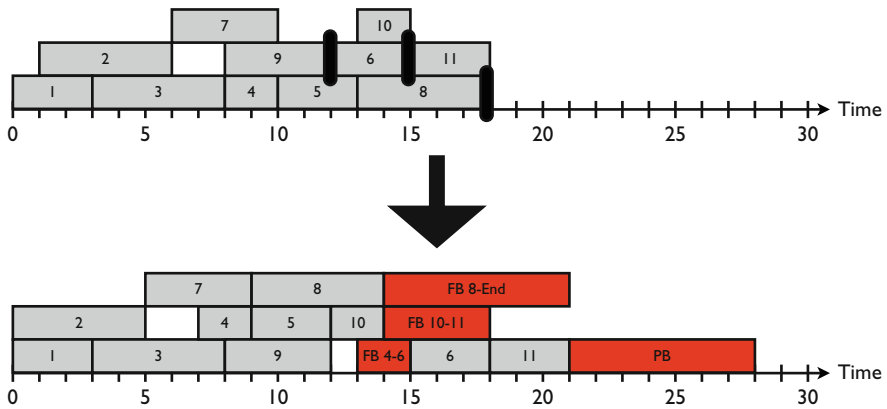
### 10.5 Project Execution and Buffer Management

The construction of a buffered baseline schedule as explained in the previous sections serves as an ideal tool during the project execution phase to monitor the project’s performance and to take corrective actions when necessary. Once the project is set off, the execution of project activities should be done according to the roadrunner mentality. As explained above, this implies that individual activity finish times are not seen as individual milestones or deadlines to guarantee that early



**Table 10.2** Estimated standard deviations for each activity  $i$

$i$	$d_i$	$\sigma_i$
0	0	0
1	3	1.2
2	5	2
3	5	2
4	2	0.8
5	3	1.2
6	3	1.2
7	4	1.6
8	5	2
9	4	1.6
10	2	0.8
11	3	1.2
12	0	0



**Fig. 10.14** Insertion of the buffers into the baseline schedule

finishes of activity predecessors have an immediate effect on the start of the activity. This mentality also implies that during project execution, contrary to initial baseline scheduling, all activities start ASAP. The key to reducing system-wide work-in-process and other disadvantages of starting activities ASAP is to control the flow of work into the system: activities without (nondummy) predecessors, the so-called gating activities, should not start before the scheduled start time, while nongating activities, especially those on the CC, should be started as soon as they can when work becomes available.

The execution of the project is managed by the use of buffers: in addition to providing aggregated protection against statistical variation, buffers are supposed to act as vital warning mechanisms. Buffer management is the key to tracking project performance in CC/BM (notice the distinct but related essential functions of buffers during baseline development and project execution). The CC is the

sequence of dependent events that prevents the project from being planned with a shorter estimate of overall duration. In this way, the CC highlights where additional resources can cause the project to be completed in a shorter interval. Given the goal of completing the project as quickly as possible, the CC is the constraint that prevents the project from making greater progress towards this goal. At the same time, the buffers are the instruments that can be utilized during project execution to determine whether the total project duration in the baseline schedule is still achievable with an appropriate degree of certainty. By comparing the current ASAP schedule with the buffer positions in the baseline, the project manager gets an idea of how many buffers have been used versus how much of the processing of its feeding chains has been completed. If the project's progress is at the start of a chain and the entire buffer has already been consumed, the project is in danger. If the progress is at the end and no buffer has been consumed, the project will probably be early. This buffer management process can be formalized. As long as there is some predetermined proportion of the buffer remaining, everything is assumed to go well (the green OK zone). If activity variation consumes buffers by a certain amount, a warning is raised to determine what needs to be done if the situation continues to worsen (the yellow watch out zone). These actions (expediting, working overtime, subcontract, etc.) are to be put into effect if the situation deteriorates past a critical point (the red action zone). Figure 10.15 provides possible buffer management thresholds. Obviously, the threshold values to trigger actions vary as a function of project or path completion.

One advantage of the FBs and the entire CC/BM method is that the need to re-schedule the project is reduced (which is labelled as *proactive scheduling* in Chap. 1). The schedule in progress is updated continuously, but the baseline schedule ordinarily remains unchanged. CC/BM states that only if the project

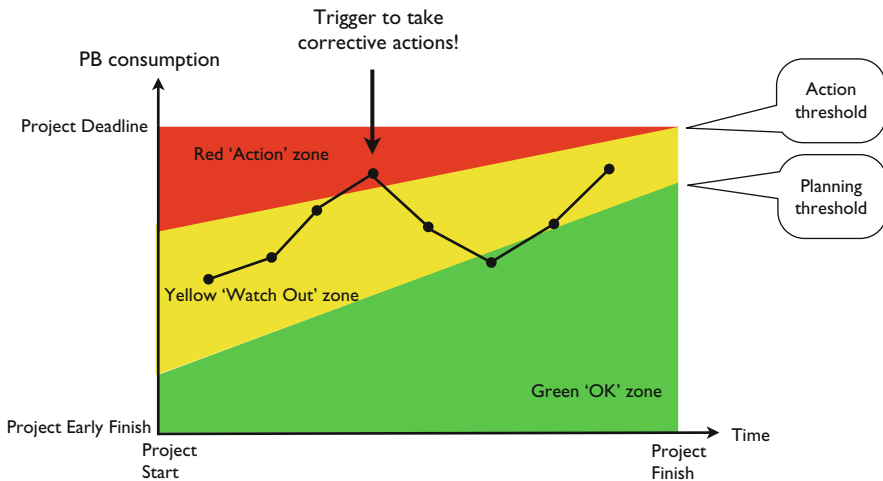


Fig. 10.15 Buffer management thresholds as a function of proportion of project completed

is in real trouble, meaning the PB is in real trouble, it will make sense to reschedule. Such circumstances will occur when it is impossible to restore the schedule in progress to the safe zone by routine actions as a response to buffer monitoring. At that moment, a new CC is identified, and a new baseline schedule needs to be developed that provides the project manager with a new assessment of the date at which he/she can anticipate the project to be completed with a convenient amount of certainty. Most CC/BM sources say that more often than not, recomputing a baseline is a final recourse that should be avoided whenever possible, to avoid system nervousness. Nevertheless, uncertain events during project execution (activity delays, the necessity to insert new activities, unavailability of resources, late deliveries by a subcontractor, etc.) may dramatically change the composition of the critical sequences. A CC may shift just as a bottleneck may shift, and although perhaps the project baseline duration is not in immediate danger, one should remain focused on a chain of activities that may have lost its criticality. This topic has been described in a number of critical review papers in which the authors state that CC/BM suffers from serious oversimplification. A few words on the main CC/BM critical points is the topic of the next section.

## 10.6 A Critical Note

Since its introduction in 1997, CC/BM is seen as an important eye opener to project management and dynamic project scheduling. The idea of protecting a deterministic baseline schedule in order to cope with uncertainties is sound and appeals to management (Herroelen and Leus 2001) and is one of the foundations of dynamic scheduling (see e.g. the topic of Chap. 5). However, shortcomings and oversimplifications are mentioned throughout various sources in the literature, which have resulted in an overwhelming amount of extensions, both research papers and books, on top of the original CC/BM philosophy. In what follows, the main points of criticism highlighted in research papers written by Herroelen and Leus (2001) and Herroelen et al. (2002) are briefly mentioned, without going into much detail.

### 10.6.1 Scheduling Objective

The CC/BM philosophy assumes that time is *the* number one scheduling objective, and ignores other regular and/or nonregular scheduling objectives as discussed throughout Chaps. 7 and 8. Consequently, it is implicitly assumed that each project is a resource-constrained project scheduling problem where the scheduling objective is the minimization of time, as discussed in Sect. 7.3.2. It should be noted, however, that a second important scheduling objective is implicitly taken into account, i.e. the minimization of the work in progress (WIP). As mentioned earlier, this scheduling

objective is taken into account through the use of the as-late-as-possible scheduling approach, and is similar to the leveling objective discussed in Sect. 7.3.4. Moreover, by scheduling activities as-late-as-possible and assuming that these activities have a negative cash flow (i.e. cost), the net present value objective of Sect. 7.3.3 has also been taken into account. However, apart from the time, leveling and net present value objectives, no other objectives that might be relevant in practice are explicitly taken into account. Future research efforts should focus on the influence of incorporating these other scheduling objectives on the relevance and use of the CC/BM approach.

### ***10.6.2 Scheduling Quality***

Goldratt minimizes the effect of the use of high-quality scheduling algorithms and states that the impact of uncertainty is much larger than the impact of using proper scheduling methods. While it can hardly be denied that uncertainty is a crucial dimension of dynamic scheduling and often has a large effect on the project performance, the beneficiary effect of sound scheduling methods should be put into the right perspective. It has been shown extensively throughout previous chapters that the critical chain not only depends on the scheduling objective, but also on the quality of the algorithm used to construct a resource feasible schedule. Consequently, the use of high-quality scheduling methods is not only crucial for the quality of the project scheduling objective (time), but also determines which activities are critical and make part of the critical chain. Moreover, when time is considered as the main scheduling objective, it is a logical choice to focus on the best performing scheduling techniques that lead to the best optimized scheduling objective value.

### ***10.6.3 Critical Chain***

The CC/BM philosophy prescribes the use and presence of a single critical chain that can be best kept constant throughout the whole project life cycle. However, it can be easily verified that in a realistic project setting, more than one chain can be critical and the presence of single or multiple critical chains depends on the way the baseline schedule is constructed (scheduling objective and scheduling quality). Moreover, a dynamic setting results in a shift of the critical chain caused by changes in activity time estimates, precedence relations, etc. The combined effects of multiple dynamic critical chains, that furthermore depend on the scheduling objective and quality of the methods used, puts the buffering approach in a more complex perspective. The CC/BM approach does not properly address these issues.

### ***10.6.4 Buffer Sizing***

Sizing buffers can be done based on the length of the critical chain (project buffer) or feeding chains (feeding buffers) as initially proposed by Goldratt, or by taking risk information of the activities on the (feeding or critical) chain into account. However, potential delays in activities that lead to buffer consumption can also be caused by the unavailability of resources. Although the original CC/BM approach suggests to use resource buffers to guarantee timely availability of these resources, it is conjectured that they are not an ideal solution to solve unexpected delays. The impact of potential delays due to resource unavailability depends on the scarceness of these resources, and therefore, knowledge about the scarceness of resources should also be taken into account when sizing buffers. A way to measure resource scarceness has been proposed in Sect. 8.3.2.

### ***10.6.5 Buffer Management***

The use of time buffers to protect the project deadline can be questioned in highly complex projects where the efficient use of limited resources is the main driver of project progress performance. Both the static insertion of buffers in the baseline schedule (scheduling phase) and the dynamic penetration of buffers during project progress (execution phase) might and often will cause new resource conflicts. Resolving these new resource conflicts might result in a need to adapt the original baseline schedule, leading to changes in the critical chain(s) and feeding chains and in the corresponding buffer sizes. Although this anomaly can be considered as a technical scheduling detail, no rules-of-thumb on best-practices to repair the original baseline schedule are given.

## **10.7 Conclusions**

The translation of the theory of constraints philosophy discussed in “The Goal” to a project environment, as described in “Critical Chain”, was a major step forward in the development of project management theory. Indeed, Goldratt illustrates in his novel the applicability of the bottleneck focus to project management environments and defines the critical chain as the project bottleneck to focus on. Similar to the inventory buffers in production environments (The Goal), he introduces the use of time buffers to protect the bottleneck (Critical Chain) against variability.

Quite a number of studies have focused on the pitfalls of the critical chain philosophy. In these studies, the authors argue that the CC/BM theory is an important eye-opener. Indeed, the point that the interaction between activity durations, precedence relations, resource requirements and availabilities determines the project

duration is well-taken but not at all a new idea (this idea was the central theme of Chaps. 7 and 8). Moreover, the protection of a deterministic baseline schedule through the insertion of buffers (project, feeding and resource buffers) is a pragmatic but sometimes a bit overly simplistic approach to the management of all forms of variability that might arise in project scheduling. Various studies stress the need for efficient algorithms for the creation of robust baseline schedules, powerful and effective warning mechanisms and mechanisms for dynamic evaluation of criticality of project activities. Nevertheless, most studies recognize that the breakthrough of project management was caused by the novel by a man who already claimed two decades ago that the identification and focus on the limiting factor (the bottleneck or the critical chain) is primordial in changing the behavior of the system under study.