

Chapter 7

Resource-Constrained Project Scheduling

Abstract Resource-constrained project scheduling is a widely discussed project management topic which has roots in and relevance for both academic and practical oriented environments. Due to its inherent problem complexity, it has been the subject of numerous research projects leading to a wide and diverse set of procedures and algorithms to construct resource feasible project schedules. Thanks to its practical value, many of the research results have found their way to practical project management projects. This chapter gives an overview of the project scheduling efforts and needs in order to bring the academics closer to real life projects and vice versa. It gives an extensive overview on the use of resources in dynamic scheduling problems and discusses various scheduling objectives to optimize the schedule to the wishes and needs of the company or project owner. The chapter also gives a set of techniques to validate the quality of the resource feasible baseline schedules and highlights some important features and characteristics that should be taken into account when using scheduling tools.

7.1 Introduction

This chapter elaborates on the critical path based scheduling approach discussed in Part I of this book where it is assumed that resources are not taken into account when scheduling projects. However, project activities are executed by resources (usually people) which mostly have a limited availability over the complete time horizon of the project. This chapter puts a detailed emphasis on these resources required to execute the project activities.

Resource-constrained project scheduling is the process of constructing a project schedule within the limited amount of resources available. It requires the examination of the possible unbalanced use of resources over time to resolve over-allocations (the so-called resource conflicts) when more resources are required than available. The critical path based scheduling methods of the previous part will often schedule certain activities simultaneously. When more resources such as machines or people

are needed than there are available, these activities will have to be rescheduled concurrently or even sequentially to resolve the resource constraints. Resource-constrained project scheduling is the process of resolving these resource conflicts under different scheduling objectives.

The outline of this chapter is as follows. Section 7.2 gives a brief description of the two main resource classes used in resource-constrained project scheduling. Section 7.3 discusses various resource-constrained project scheduling types under different scheduling objectives. It will be shown that each scheduling objective serves another practical project purpose and determines the main characteristics of the project scheduling approach. In Sect. 7.4, easy and quick scheduling techniques are presented to construct the resource-constrained project schedules under the different objectives. Section 7.5 puts the assumptions into perspective and extends the scheduling results of previous sections to more practical oriented settings. Section 7.6 discusses how the total activity cost, as a weighted sum of the costs for the activities and their resource assignments, can be calculated. Section 7.7 draws general chapter conclusions.

7.2 Resources

In the previous sections, it is assumed that project activities do not require resources during their execution (or alternatively, the assumption is that the resources are unlimited in availability). In practice, activities need resources during their execution, that are limited in availability. These resources have been classified in two basic categories.

- *Nonrenewable or consumable resources* are available on a total project basis, with a limited consumption availability for the entire project. Typical examples are money, raw materials, energy, . . .
- *Renewable resources* are available on a period-by-period basis, i.e. the available amount is renewed from period to period. Only the total resource use at every time instant is constrained. Typical examples are manpower, machines, tools, equipment, space, . . .

Doubly-constrained resources are a combination of the two previous categories, and are constrained per period (e.g. per period cash flows) as well as for the overall project (e.g. total expenditures, overall pollution limits, . . .).

In this chapter, the resource focus is restricted to the use of renewable resources with a limited availability, since they are responsible for an increase in the scheduling complexity. The use of nonrenewable resources has been previously discussed in Chap. 3, with the trade-off between the activity duration and its corresponding cost, which is clearly a nonrenewable resource. The current chapter and Chaps. 8 and 9 elaborate on project scheduling techniques under low uncertainty and high complexity (see quadrant 3 of Fig. 1.4). Chapter 10 extends this complex scheduling approach to higher degrees of uncertainty.

7.3 Scheduling Objective

The scheduling phase aims at the construction of a timetable to determine the activity start and finish times in order to optimize a certain predefined scheduling objective. Although the minimization of the project lead time is probably the most important objective during the scheduling phase, other scheduling objectives are often crucial from a practical point of view. While the focus of Chap. 2 was on resource-unconstrained project scheduling with the minimization of the total project time as the only scheduling objective, the current chapter will add two important extensions. First, project scheduling tools and techniques are presented with the presence of limited availability of renewable resources and, second, extensions will be made to various other scheduling objectives, each of them classified in one of two main scheduling objective classes.

7.3.1 *Regular and Nonregular Objectives*

Scheduling activities over time implies the optimization of a certain objective function. These objective functions (often referred to as measures of performance) are numerous and often complex. Moreover, when several objectives are relevant at the same time, it is quite possible that a trade-off will be present since the objective functions are often conflicting. Trade-offs between scheduling objectives can be taken into account using two general principles. In the *multi-criteria* case, the different measures must be ranked according to certain criteria in order to specify in which order they should be considered. The *multi-objective* case combines or weighs the different kinds of objectives. During the past decades, a huge number of different scheduling objectives were treated in the project scheduling literature (for an overview, see Icmeli et al. (1993), Elmaghraby (1995), Özdamar and Ulusoy (1995), Herroelen et al. (1998) and Brucker et al. (1999)). Since time is a crucial factor in the global market, the vast majority of the project scheduling methodologies presented in the literature have been developed with the objective of minimizing the project length subject to precedence as well as resource constraints. The measure is attractive from a practical point of view. As an example, Shtub et al. (1994) stated that successful companies will be those that learn to make and deliver goods and services faster than their competitors do. Nowadays, the factor time has become a critical element in the so-called time-based competition.

Nevertheless, the focus on the minimization of the duration of the project ignores some realistic and very important factors. Factors such as costs, quality, risk, safety and maintainability are sometimes much more relevant than solely the time aspect of projects. Besides the minimization of the project duration, an almost endless list of possible measures of performance is treated in the literature. Objectives such as minimizing direct costs, risks or expenditures for material, equipment or labor,

or maximizing revenues, safety, quality, etc. play an important role in practical scheduling problems. According to Pinnell and Busch (1993), the quality of a project can be measured by time (how long it takes), cost (within the budget) and how well the project conforms to specifications. A survey of variants and extensions of the resource-constrained project scheduling problems can be found in Hartmann and Briskorn (2010). Taking these observations into account, the objective functions are classified into two distinct classes: regular measures of performance and nonregular measures of performance.

A *regular measure of performance* (also referred to as early completion measures) is a nondecreasing (in the case of a minimization problem) or a nonincreasing (in the case of a maximization problem) function of the activity finish times. According to French (1982) it means that, when R is a regular function of f_1, f_2, \dots, f_n , such that $f_1 \leq f_1', f_2 \leq f_2', \dots, f_n \leq f_n'$, then

$$R(f_1, f_2, \dots, f_n) \leq R(f_1', f_2', \dots, f_n') \text{ (minimization problem)}$$

or

$$R(f_1, f_2, \dots, f_n) \geq R(f_1', f_2', \dots, f_n') \text{ (maximization problem)}$$

with f_i the finish time of activity i and n the number of activities in a project.

The minimization of the project duration is undoubtedly the most popular regular measure of performance. Other examples of regular measures of performance are the minimization of the mean flow-time, the mean lateness, the mean tardiness and the number of activities that are tardy. The logic of this type of measure lies in the fact that it will never be beneficial to delay an activity towards the end of the project when only the precedence constraints are considered and consequently without considering the constraints on the resources. Therefore, when two feasible project schedules have been constructed such that each activity under the first schedule finishes no later than the corresponding finish time in the second schedule, then the first schedule is at least as good as the second schedule. In recent years, scheduling problems with *nonregular measures of performance* (also referred to as free completion measures) have gained increasing attention. A nonregular measure of performance is a measure for which the above definition does not hold. This implies that delaying an activity may result in an improvement of the performance measure. Practical applications of nonregular measures of performance often introduce financial aspects of project management. Such financial aspects may take the form of cash flows related to the activities of a project, penalty costs related to the due dates of the activities, quality costs, and many more. The project scheduling objectives described in Sects. 7.3.3, 7.3.4 and 8.2 are all nonregular measures of performance.

Similar to Part I of this book, it is assumed that a project consists of a set of activities N , numbered from a dummy start node 1 to a dummy end node n , between which precedence relations exist. For the sake of simplicity, it is assumed that all precedence relations are minimal finish-start relations with a time-lag of zero, unless

otherwise indicated. Moreover, there are K renewable resources with a_k ($1 \leq k \leq K$) the availability of resource type k over the complete scheduling horizon and with r_{ik} ($1 \leq i \leq n$, $1 \leq k \leq K$) as the resource requirements of activity i with respect to resource type k . No nonrenewable resource constraints are taken into account, unless otherwise indicated. The start and end dummy activities representing the start and finish of the project have a duration and a renewable resource requirement equal to zero. A project network is represented by a topologically ordered activity-on-the-node format where A is the set of pairs of activities between which a finish-start precedence relationship with time lag 0 exists. It is assumed that the graph $G(N, A)$ is acyclic. A schedule is defined by an n -vector of start times $s = (s_1, \dots, s_n)$, which implies an n -vector of finish times $f(f_i = s_i + d_i, \forall i \in N)$. A schedule is said to be feasible if the precedence and resource constraints are satisfied. The objective of the scheduling problem type differs along the subsections, as described hereunder.

7.3.2 Time Minimization

The critical path calculations of Chaps. 2 and 3 ignore the limited availability of renewable resources. However, when the use of renewable resources is limited in availability, the construction of a feasible schedule with a minimal entire project duration might be a complex task. The presence of limited renewable resources often results in resource over-allocations (or resource conflicts) at certain points in time where more resources are needed than actually available. In this case, activities need to be shifted in time in order to resolve the resource conflict and to make the schedule both precedence as well as resource feasible. A resource conflict occurs when there is at least one period $]t - 1, t]$ for which $\exists k \leq K : \sum_{i \in S(t)} r_{ik} > a_k$, where $S(t)$ denotes the set of activities in progress in period $]t - 1, t]$.

Table 7.1 is an extended version of Table 2.2 containing information of the activity use of a single renewable resource which is assumed to be available for 10 units (i.e. $a_1 = 10$). The columns with the cash flow and resource group are irrelevant for this section and can be ignored. The network logic of the table is graphically displayed in an activity-on-the-node network of Fig. 7.1 with each number above the node denoting the activity duration and the number displayed below each node the resource demand. The resource graph of the ESS is given in Fig. 7.2. If the availability of the example project equals $a_1 = 10$, then there are resource over-allocations of the single renewable resource type at periods $]0, 3]$ and $]5, 10]$ (as an example, the set of activities in progress at period $]2, 3]$ is equal to $S(3) = \{2, 3, 4\}$). In order to resolve these resource conflicts, some activities that incur such a conflict need to be shifted forwards in time.

This problem is known as the *Resource-Constrained Project Scheduling Problem* (RCPSP), which can be formally stated as follows. A set of project activities is to be scheduled within a minimal project duration and subject to the precedence relations and a set of renewable resource types. It is mentioned earlier that a schedule is said

Table 7.1 Activity and resource data for the example project of Fig. 7.1

Activity	Predecessors	Duration	Resource use	Cash flows (€)	Resource group
1	–	0	0	0	–
2	1	6	7	–50	A
3	1	5	1	50	B
4	1	3	5	–50	–
5	3	1	6	–100	–
6	3	3	1	10	–
7	3	2	2	–10	B
8	4	1	2	20	–
9	2	4	8	100	A
10	5	3	6	0	–
11	7	1	2	–50	B
12	6, 10, 11	3	6	50	B
13	8, 12	5	8	100	–
14	9, 13	0	0	1,000	–

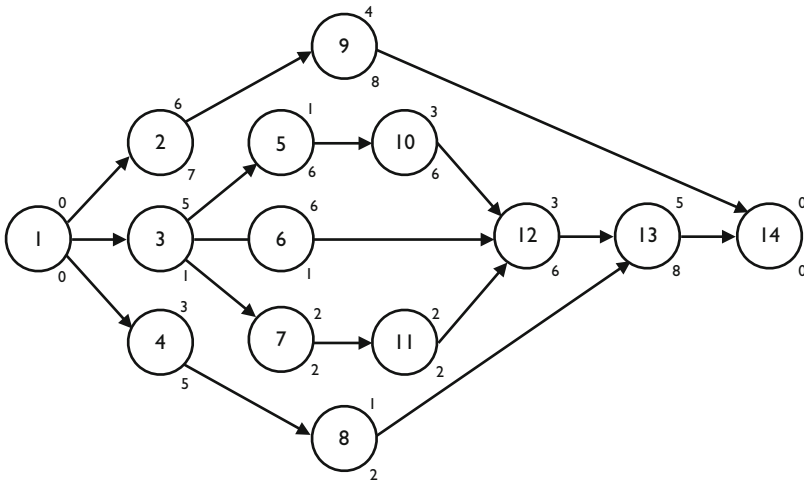


Fig. 7.1 The AoN example network of Table 7.1

to be feasible if the precedence *and* the renewable resource constraints are satisfied. Consequently, the objective of the RCPSP is to find a feasible schedule such that the total project duration (often referred to as the schedule makespan) s_n is minimized. A conceptual formulation for the RCPSP can be given as follows:

$$\text{Minimize } s_n \tag{7.1}$$

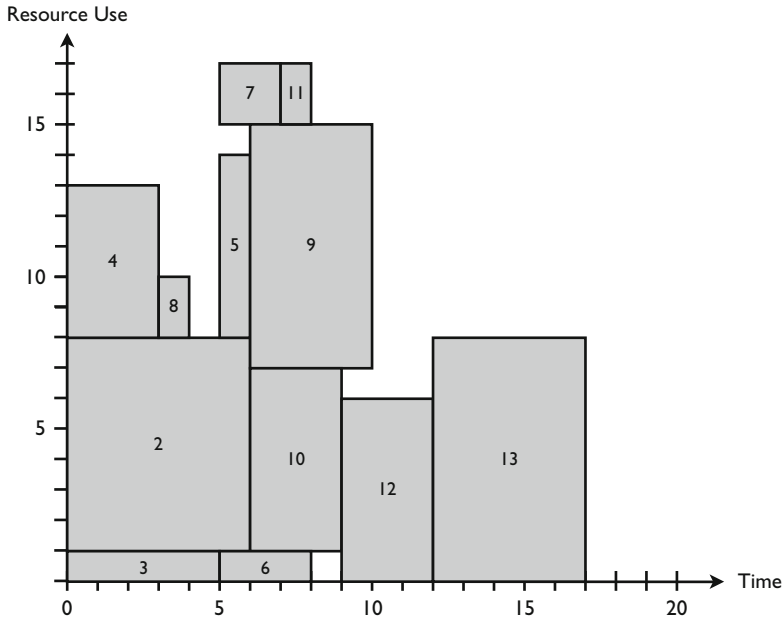


Fig. 7.2 Resource graph for earliest start schedule (ESS)

subject to

$$s_i + d_i \leq s_j \quad \forall (i, j) \in A \tag{7.2}$$

$$\sum_{i \in S(t)} r_{ik} \leq a_k \quad k = 1, \dots, K; t = 1, \dots, T \tag{7.3}$$

$$s_1 = 0 \tag{7.4}$$

$$s_i \in \text{int}^+ \quad i = 1, \dots, n \tag{7.5}$$

The objective in Eq. 7.1 minimizes the entire project duration by minimizing the start time of the dummy end activity. The constraint set given in Eq. 7.2 maintains the finish-start precedence relations with a time-lag of zero among the activities. Equation 7.3 are used to model the limited availability of the renewable resources. It stipulates that the sum of the resource requirements of all activities in progress at time period $]t - 1, t]$ is not allowed to exceed the availability of the renewable resource over the complete time horizon of the project T . Equation 7.4 forces the dummy start activity, and hence the project, to start at time zero. Equation 7.5 ensure that the activity start times assume nonnegative integer values.

In the remainder of this section, it is assumed that each activity follows a “fixed duration” mode, i.e. the time estimates (in duration units, above the node) and resource demand (in units, below the node) cannot be changed during scheduling.

In Sect. 7.5, this assumption will be extended to a “fixed work” mode where the product of both estimates, the so-called work content, is fixed but the individual estimates for the activity duration and resource unit demand can change, respecting the following equation:

$$\text{Work} = \text{Duration} * \text{Units}$$

Figure 7.3 displays the optimal resource chart with a minimal entire project duration of 25 time units. The presence of renewable resources with limited availability increases the minimal critical path length from 17 to 25 time units. The minimal length of the project is now determined by the so-called *critical chain*, which replaces the critical path calculations when resources are limited in availability. Consequently, the critical chain of a project is the set of dependent activities that defines the minimal project duration. These dependencies include both technological precedence relations (like the traditional critical path calculations) as well as resource dependencies (due to the limited availability). The critical chain is similar to a resource-constrained critical path and requires the attention of the project manager. The critical chain of the example project consists of the dependent set of activities 2, 4, 5, 9, 10, 12 and 13, which determines the minimal entire project duration of 25 time units.

Note that the problem formulation without Eq. 7.3 boils down to the ordinary critical path calculations of Chap. 2. The logical extension from minimal precedence relations with a zero time-lag to generalized precedence relations leads to an increase in schedule problem complexity. In the conceptual formulation, the Eq. 7.2 will be replaced by the generalized precedence relations of Bartusch et al. (1988) as discussed in Sect. 2.2.3 as follows:

$$s_i + l_{ij} \leq s_j \quad \forall (i, j) \in A$$

Since the introduction of renewable resources will lead to an increase of the entire project duration (the entire project duration might remain unchanged when

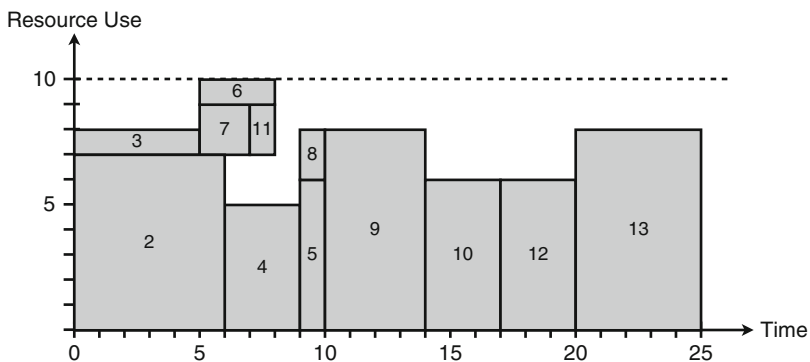


Fig. 7.3 Feasible resource graph with minimal time

Table 7.2 Four renewable resources for the example project of Fig. 7.1

Activity	Duration	Resource 1	Resource 2	Resource 3	Resource 4
1	0	0	0	0	0
2	6	7	15	2	6
3	5	1	8	4	8
4	3	5	8	3	3
5	1	6	15	2	6
6	3	1	13	0	3
7	2	2	16	2	0
8	1	2	9	4	4
9	4	8	12	5	5
10	3	6	17	5	0
11	1	2	10	2	5
12	3	6	5	5	4
13	5	8	10	3	7
14	0	0	0	0	0

no resource conflicts occur in the critical path schedule), the critical path is a lower bound for the entire project duration under the presence of renewable resources. Obviously, the introduction of additional renewable resource types will further increase the total project duration. Consider, for example, Table 7.2 where the example project is subject to four renewable resource types with a limited availability of $a_1 = 10$, $a_2 = 20$, $a_3 = 8$, $a_4 = 10$.

It can be shown that the use of renewable resources under limited availability leads to an increase in the project duration. In the example, the project duration increases when iteratively taking the resources into account, from 25 (only resource 1 is taken into account) to 31 (resources 1 and 2), 33 (resources 1, 2 and 3) and 36 (all resources).

For illustrative purposes, Fig. 7.4 displays the four resource graphs with a project duration of 36 time units. In the remainder of this chapter, a single renewable resource type is used to illustrate all project scheduling objectives, without loss of generality.

7.3.3 Net Present Value Maximization

In Chap. 4 of this book, it is already mentioned that, when the financial aspects of project management are taken into consideration, there is a decided preference for the maximization of the net present value (*npv*) of the project as the more appropriate objective, and this preference increases with the project duration (Herroelen et al. 1997). This criterion is based on a very simple principle: accept the project if the *npv* is greater than or equal to zero and reject it when the *npv* is less than zero. In this *accept-reject decision* mechanism, the value of the cash flows (either cash inflows or cash outflows) will be compounded towards time zero by

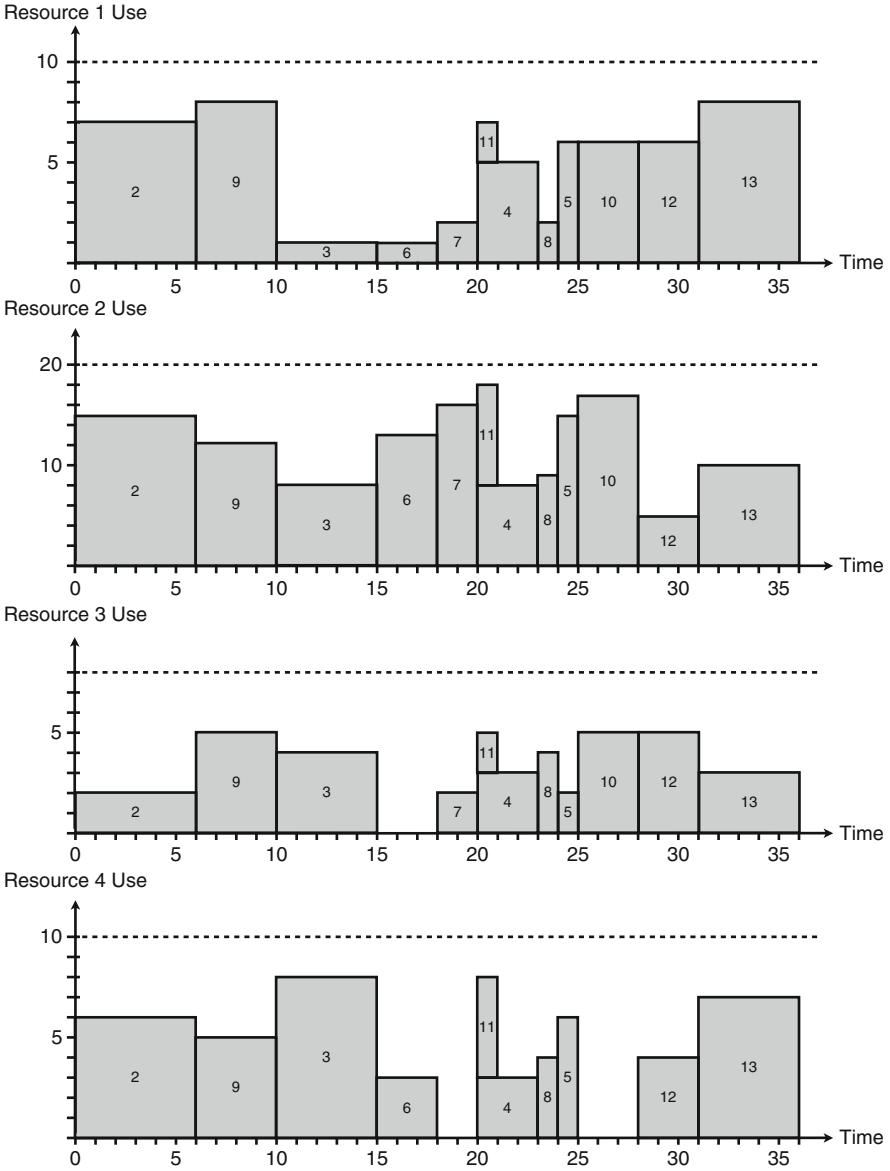


Fig. 7.4 Feasible resource graphs with four renewable resources

means of a discrete discount rate r as follows:

$$npv = cf_{10} + \sum_{i=2}^n \sum_{t=1}^{\infty} \frac{cf_{it}}{(1+r)^t}$$

where cf_{it} denotes the cash flows associated with activity i at the end of period t . Notice that cf_{i0} denotes the initial cash flow for the dummy start activity 1 at the end of period zero, which usually takes the form of an initial investment when the project starts. When a continuous discount factor α is introduced, the npv can be calculated as follows:

$$npv = cf_{i0} + \sum_{i=2}^n \sum_{t=1}^{\infty} cf_{it} e^{-\alpha t}$$

Various assumptions concerning the cash flows cf_{it} are presented in the literature. A lot of research has focused in the case where the cash flows are independent of the finish time of the corresponding activity. In that case, a terminal value of each activity upon completion can be calculated by compounding the associated cash flows cf_{it} towards the end of the activity, i.e. terminal cash flow $c_i = \sum_{t=1}^{d_i} cf_{it} e^{\alpha(d_i-t)}$. This terminal value is assumed to be independent of the time of completion of the corresponding activity. Figure 7.5 shows an illustrative example of an activity with a baseline duration of 5 weeks and a cash flow prediction of €-10, €-30, €-50, €-40 and €200 for weeks 1-5. These cash flows (positive and negative) are assumed to be independent of the start time of the activity, and therefore, the total value can be compounded to a single cash flow value $c_i = -10e^{0.05*4} - 30e^{0.05*3} - 50e^{0.05*2} - 40e^{0.05*1} + 200e^{0.05*0} = €55.62$ when using a discount rate of $\alpha = 0.05$. The compounded values are still independent from the start and finish time of the activity and will be used in the further calculation of the net present value of the project schedule.

Research work has also been done on the case where activity cash flows are dependent on the finish times of the corresponding activities, resulting in a large number of different time dependencies. Examples are given along the following lines:

- Progress payments: The cash outflows occur when an activity is completed as a result of cost of resources, materials, etc. whereas cash inflows are incurred as progress payment at the end of some time period (e.g. each month) for the work completed during this fixed time period. For references in the literature, the reader is referred to Kazaz and Sepil (1996), Sepil and Ortac (1997) and Vanhoucke et al. (2003).
- Linear dependent cash flows: Cash flows are assumed to be linearly dependent or a nonincreasing step function in time. Linear time-dependent cash flow functions occur in situations where penalties are involved. This means that the contractor

€ -10	€ -30	€ -50	€ -40	€ 200
=				
€ 0	€ 0	€ 0	€ 0	€ 55.62

Fig. 7.5 Independent activity cash flows can be represented by a single cash flow value using discount rates

will receive the activity cash flows as agreed from the owner of the project (which can be negative to the contractor after cost reductions) if the schedule meets the targets. Penalties can be imposed for each activity which grow linearly in time, i.e. for every day that the activity finish time is later than its earliest finish time. For references in the literature, the reader is referred to Etgar and Shtub (1999) and Vanhoucke et al. (2001b).

- Cash flow payment problem: Up to now, it is assumed that the amount of cash flow is known at the scheduling phase for each individual activity. In practice, both the amount and timing of activity cash flows (given a certain total predefined project budget) must be determined simultaneously. This so-called payment scheduling problem of simultaneously determining the amount and timing of progress payments in order to maximize the net present value reduces to distributing payments over the duration of the project. This scheduling problem has been studied by, among others, Dayanand and Padman (1997, 2001a,b).
- Equitable solution: The net present value for a contractor is not the same as the net present value for the client of the project. Indeed, cash inflows for the contractor are, in the extreme, cash outflows for the client. A so-called equitable project schedule is a schedule where both the contractor and the client deviate from their respective ideal solutions (i.e. the net present value of the schedule) by an equal percentage. A reference in the literature can be found at Ulusoy and Cebelli (2000).

In this section, the activity cash flows are assumed to be independent of their time of occurrence. Extensions to time-dependent cash flows lead to an increase in scheduling complexity, but do not fundamentally change the underlying scheduling approach. These extended scheduling problems are outside the scope of this chapter. The *resource-constrained project scheduling problem with discounted cash flows* (RCPSDC) where cash flows are assumed to be time-independent can be conceptually formulated as follows:

$$\text{Maximize } \sum_{i=1}^n c_i e^{-\alpha(s_i+d_i)} \quad (7.6)$$

subject to

$$s_i + d_i \leq s_j \quad \forall (i, j) \in A \quad (7.7)$$

$$\sum_{i \in S(t)} r_{ik} \leq a_k \quad k = 1, \dots, K; t = 1, \dots, T \quad (7.8)$$

$$s_1 = 0 \quad (7.9)$$

$$s_n \leq \delta_n \quad (7.10)$$

$$s_i \in \text{int}^+ \quad i = 1, \dots, n \quad (7.11)$$

The objective in Eq. 7.6 maximizes the net present value of the project by discounting the terminal activity cash flows c_i towards the beginning of the project. The constraint set given in Eq. 7.7 maintains the finish-start precedence relations among the activities. The limited availability of the renewable resources is modeled by Eq. 7.8. Equation 7.9 forces the dummy start activity, and hence the project, to start at time zero while Eq. 7.10 limits the project duration to a negotiated project deadline δ_n . Equation 7.11 ensure that the activity start times assume nonnegative integer values.

Figure 7.6 displays the optimal resource-feasible schedule for the example project of Table 7.1 with a discount rate $\alpha = 0.01$, a project deadline $\delta_n = 25$ and activity cash flows given in Table 7.1 that occur at the finish of the activity (i.e. the values are assumed to be equal to c_i as a discounted value of all its cf_{it} values). The resource-constrained project scheduling problem with the use of activity cash flows is very similar to the classical RCPSP since it only differs in its objective function. The objective is to schedule all the activities in order to maximize the net present value of the project, without violating the precedence or resource constraints and subject to a given deadline. The simple translation of the *npv*-philosophy into project scheduling is to schedule all activities with a positive cash flows as soon as possible and to delay all activities with a negative cash flow as late as possible towards the deadline. Notice that the finish times of activities 4, 5, 7, 8 and 11 have been shifted to the project deadline compared to the time objective and activities 9 have been scheduled earlier, which has resulted in an overall improvement of the objective value. In fact, it leads to an improvement for activities 4, 5, 7, 9 and 11 and a deterioration for activity 8. The total net present value equals $npv = -50e^{(-0.01*6)} + 50e^{(-0.01*5)} - 50e^{(-0.01*13)} - 100e^{(-0.01*14)} + 10e^{(-0.01*8)} - 10e^{(-0.01*16)} + 20e^{(-0.01*14)} + 100e^{(-0.01*10)} + 0e^{(-0.01*17)} - 50e^{(-0.01*17)} + 50e^{(-0.01*20)} + 100e^{(-0.01*25)} + 1,000e^{(-0.01*25)} = \text{€}833.65$, which is an improvement of the net present value of $\text{€}12.95$ for the schedule of Fig. 7.3. Note that the project duration is equal to 25, which is the minimal duration possible (see previous section). Thanks to a big lump

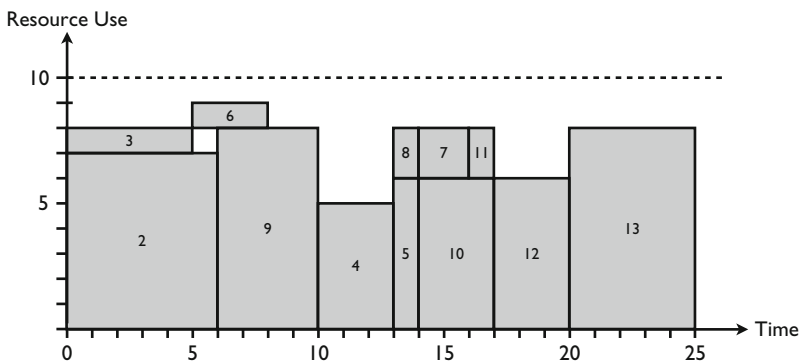


Fig. 7.6 Feasible resource graph with maximal *npv*

sum payment for the dummy end activity of €1,000, this would also have been the case, even if the deadline constraint $\delta_n \leq 25$ would have been omitted.

The importance of the net present value in project scheduling has been described in a paper by Herroelen et al. (1997). In this paper, the use of the max-*npv* objective over a relatively short period of time is seriously questioned. Furthermore, the max-*npv* objective also seems to be questionable under conditions of uncertainty. Consequently, the relevance of the max-*npv* objective is limited to those situations where the time value of money is to be taken into consideration. It is clear that this is the case in capital intensive projects with a sufficiently long duration, significant cash flows, high interest rates and high costs of capital. This may be the case in the construction industry but is certainly not the case in production environments with a time horizon of days or weeks.

7.3.4 Resource Leveling

Both the RCPSP and the RCPSPDC aim at the construction of a precedence and resource feasible schedule with a certain objective (time and *npv*, respectively). When over-allocations of resources occur (known as the so-called resource conflicts), they are resolved by shifting activities further in time. However, none of the previous problem types explicitly takes the resulting pattern of total resource use over time into account, and hence, project schedules with very irregular resource use (peak demands as well as low resource demands) are a matter of degree. The *resource-leveling project scheduling problem* (RLPSP) aims at the construction of a precedence and resource feasible schedule within a predefined deadline with a resource use that is as level as possible within the project horizon. In order to avoid jumps from peaks to low resource demands, the total use of all resources needs to be balanced over the total schedule horizon.

The RLPSP involves the scheduling of project activities in order to level the use of the renewable resources subject to technological precedence relations and a predefined project deadline. Often, no explicit resource availability is taken into account, although many projects are simply subject to a maximum resource availability as specified in the previous scheduling objectives. The objective function depends on the total resource use $\sum_{i \in S(t)} r_{ik}$ for each resource type k during the complete time horizon $t = 1, \dots, \delta_n$, denoted as $f_k \left(\sum_{i \in S(t)} r_{ik} \right)$ and can be formulated in different ways. Note that $f_k(x)$ is used to refer to a function that depends on x and is not used to refer to a finish time of activity. A conceptual formulation for the RLPSP can be given as follows:

$$\text{Minimize } \sum_{k=1}^K \sum_{t=1}^{\delta_n} f_k \left(\sum_{i \in S(t)} r_{ik} \right) \quad (7.12)$$

subject to

$$s_i + d_i \leq s_j \quad \forall (i, j) \in A \quad (7.13)$$

$$\sum_{i \in S(t)} r_{ik} \leq a_k \quad k = 1, \dots, K; t = 1, \dots, \delta_n \quad (7.14)$$

$$s_1 = 0 \quad (7.15)$$

$$s_n \leq \delta_n \quad (7.16)$$

$$s_i \in \text{int}^+ \quad i = 1, \dots, n \quad (7.17)$$

The objective in Eq. 7.12 is a general formulation to express a preference for a balanced use of resources over the scheduling horizon. All other constraints modeled by Eqs. 7.13–7.17 are similar to the scheduling problems of previous sections. The objective function of this formulation can take many forms and measures how leveled the resource use is. Common expressions are as follows:

$$f_k \left(\sum_{i \in S(t)} r_{ik} \right) = w_k \left(\sum_{i \in S(t)} r_{ik} \right)^2$$

$$f_k \left(\sum_{i \in S(t)} r_{ik} \right) = w_k \left(\sum_{i \in S(t)} r_{ik} - a_k \right)^2$$

$$f_k \left(\sum_{i \in S(t)} r_{ik} \right) = w_k \left| \sum_{i \in S(t)} r_{ik} - a_k \right|$$

$$f_k \left(\sum_{i \in S(t)} r_{ik} \right) = w_k \left| \sum_{i \in S(t)} r_{ik} - \sum_{i \in S(t-1)} r_{ik} \right|$$

The first equation measures the leveling objective as a total weighted sum of the squared resource use, with a weighting factor w_k that depends on the importance of resource k . Expressions 2 and 3 take an explicit resource availability into account, and weigh the absolute or squared deviation of the resource use from a given resource availability a_k . The last expression weighs the variation of the resource use from period to period and hence aims at the minimization of the resource use jumps over time. Figure 7.7 shows a feasible project schedule where the use of the renewable resource is more or less leveled. It can be easily verified that the values for the four objective functions are equal to 1,411, 171, 63 and 14 (under the assumption that $w_1 = 1$), respectively.

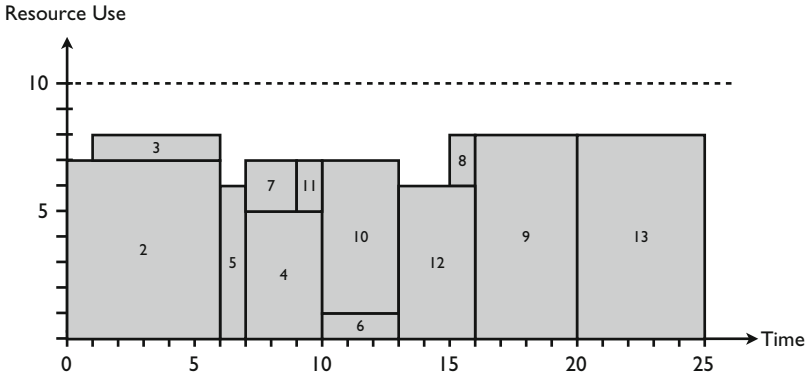


Fig. 7.7 Feasible resource graph with leveled resource use

7.4 Scheduling Methods

Resource-constrained project scheduling has been a research topic for many decades, resulting in a wide variety of optimization procedures that differ in objective functions, activity assumptions, resource constraints and many more. The main focus on the total project time minimization has led to the development of various exact and heuristic procedures for scheduling projects with tight renewable resource constraints where the well-known resource-constrained project scheduling problem (RCPSP) clearly took the lead. This problem type of Sect. 7.3.2 aims at minimizing the total duration of a project subject to precedence relations between the activities and the limited renewable resource availabilities and is known to be NP hard (Blazewicz et al. 1983). Extensions to other objective functions (see e.g. the scheduling objectives discussed in Sects. 7.3.3, 7.3.4 and 8.2), alternative resource constraints (to nonrenewable and doubly-constrained resources), multiple activity modes, etc. often result in highly complex optimization problems and have been studied in the literature by many authors. For an extensive overview of resource-constrained project scheduling procedures in the literature, the reader is referred to the research handbook written by Demeulemeester and Herroelen (2002).

The vast majority of resource-constrained scheduling procedures can be classified in two categories: exact and heuristic procedures. Exact procedures aim at finding the best possible solution for the scheduling problem type and are therefore often restricted to small projects under strict assumptions. This class of optimization procedures is widely available in the literature but is not discussed in this book. The second class of scheduling procedures aims at finding good, but not guaranteed to be optimal schedules for more realistic projects (i.e. under different assumptions and for larger sizes) in a reasonable (computational) time. Although these procedures do not guarantee an optimal solution for the project, they can be easily embedded in any scheduling software tool due to their simplicity and generality to a broad range of different projects.

In the remaining sections of this chapter, *constructive heuristic procedures* will be discussed to schedule resource-constrained projects under various scheduling objectives. Sections 7.4.1, 7.4.2 and 7.4.3 are devoted to techniques to construct and assess project schedules with regular measures of performance (in casu, the RCPSP with time minimization) while Sect. 7.4.4 presents a general technique for project scheduling problems under nonregular measures of performance.

7.4.1 *Constructive Heuristics*

Priority rule based scheduling is a simple and quick scheduling method used to construct project schedules under a regular measure of performance. A priority rule based heuristic is a single-pass method to construct a feasible schedule based on two components: a priority rule and a schedule generation scheme. A *priority rule* is used to select the next activity during the heuristic search process. Therefore, a priority rule requires a priority list that ranks all project activities in a certain order. A *schedule generation scheme* aims at the generation of a feasible schedule by extending the partial schedule (i.e. a schedule where only a subset of the activities has been assigned a start and finish time) in a stage-wise fashion. Hence, the heuristic priority-based procedure consecutively selects project activities from the constructed priority list and schedules them according to the rules of the schedule generation scheme. These two components are discussed along the following subsections.

Priority Rules

A priority rule determines the activity that is selected next during the heuristic search process, and results in a priority list in which the activities are ordered such that the network structure has been preserved. This means that no activity should be ordered before one of its predecessors. In what follows, four classes of priority rules are discussed, and a number of examples for each class is given. A fifth class is mentioned as a combined version of the four classes. The alert reader can extend each class to many other priority rules.

1. Activity based priority rules: Construction of a priority list is based on the characteristics of the project activities. The most straightforward characteristic is the duration of an activity.
2. Network based priority rules: The priority list is constructed based on the network logic, i.e. the set of activities and their precedence relations.
3. Critical path based priority rules: Critical path based scheduling information is used to construct the priority list.
4. Resource based priority rules: Priority lists are constructed based on the network logic and the resource information.

Table 7.3 Various priority rules used to construct a schedule for the example project of Fig. 7.1

Priority rule	Priority list	SSGS	PSGS
Activity based rules			
Shortest Processing Time (SPT)	[4,8,3,5,7,11,6,10,12,13,2,9]	27	25
Longest Processing Time (LPT)	[2,3,9,4,6,7,5,10,8,11,12,13]	25	25
Network based rules			
Most Immediate Successors (MIS)	[3,2,4,5,6,7,8,9,10,11,12,13]	25	25
Most Total Successors (MTS)	[3,5,7,4,6,10,11,2,8,12,9,13]	27	25
Least nonrelated Jobs (LNRJ) ^a	[3,5,7,10,11,6,12,4,8,13,2,9]	27	25
Greatest Rank Positional Weight (GRPW) ^b	[3,2,6,4,8,5,10,9,7,11,12,13]	25	25
Critical path based rules			
Earliest Start Time (EST)	[2,3,4,8,5,6,7,9,10,11,12,13]	25	25
Earliest Finish Time (EFT)	[4,8,3,2,5,7,6,11,10,9,12,13]	25	25
Latest Start Time (LST)	[3,5,6,7,10,2,4,11,12,8,13,9]	27	25
Latest Finish Time (LFT)	[3,5,7,6,10,11,4,8,12,2,9,13]	27	25
Minimum Slack (MSLK)	[3,5,10,6,7,11,12,2,9,4,8,13]	27	26
Resource based rules			
Greatest Resource Demand (GRD)	[2,9,4,3,5,10,7,6,8,11,12,13]	25	25
Greatest Cumulative Resource Demand (GCRD)	[2,9,3,5,10,6,4,8,7,11,12,13]	25	25

^aA job (activity) is not related to another job if there is no precedence related path between the two activities in the project network

^bThe GRPW is calculated as the sum of the duration of the activity and the durations of all its immediate successors

5. Composite priority rules: A weighted combination of previous priority rules is used to construct the priority list.

Table 7.3 gives an incomplete list of priority rules from each of the first four classes for the example project of Fig. 7.1. The solution obtained by using a serial (SSGS) and parallel (PSGS) schedule generation schemes, discussed hereafter, is also reported.

Schedule Generation Schemes

A schedule generation scheme determines the way in which a feasible schedule is constructed by assigning start times to the project activities. A serial schedule generation scheme (SSGS) follows an activity-incrementation approach while a parallel schedule generation scheme (PSGS) makes use of a time-incrementation scheme.

Using a *serial schedule generation scheme*, the priority values can be used to construct a project schedule by scheduling each activity one-at-a-time and as soon as possible within the precedence and resource constraints. Consequently, the SSGS scans the priority list and selects at each stage the next activity from the priority

list in order to schedule it at its first possible start time without violating both the precedence and resource constraints.

A *parallel schedule generation scheme* iterates over the time horizon of the project (instead of the priority list) and adds activities that can be scheduled at a certain predefined moment in time. It selects at each decision point t the eligible activities and assigns a scheduling sequence of these eligible activities according to the priority list. At each decision point, the eligible activities are scheduled with a start time equal to the decision point (on the condition that there is no resource conflict). Activities that cannot be scheduled due to a resource conflict are skipped and become eligible to schedule at the next decision point t' , which equals the earliest finish time of all activities active at the current decision point t .

Figure 7.8 displays the solution obtained by using the serial (top) and parallel (bottom) schedule generation schemes using the SPT priority rule, resulting in a project duration of 27 and 25 time units, respectively. The assignment of activities to the project schedule using a parallel schedule generation scheme is illustrated in Table 7.4. The set of eligible activities for each time decision point t is ranked according to the SPT rule, and activities are added to the schedule at time point t in the order given by the ‘Ranked’ column and in case it does not lead to a resource conflict.

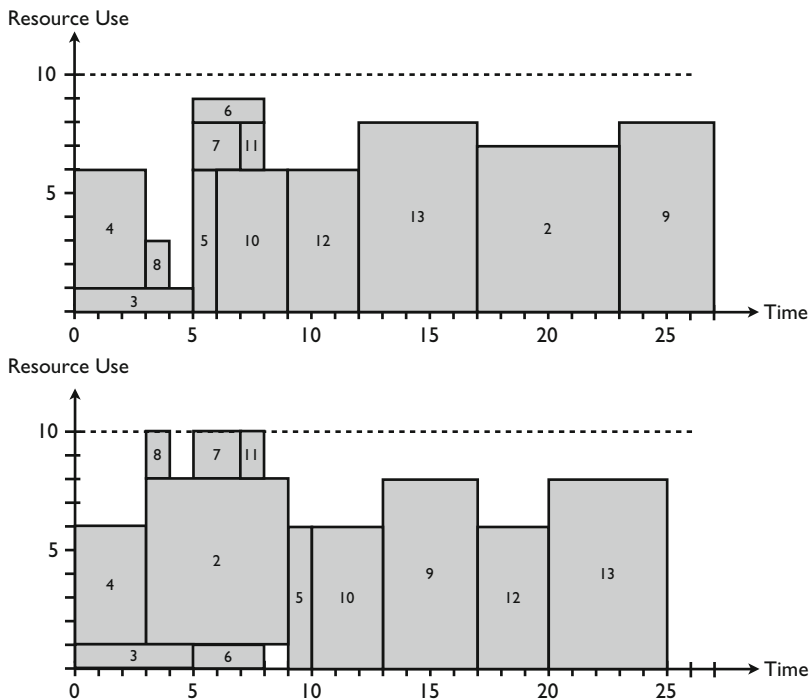


Fig. 7.8 A feasible project schedule obtained using the SPT rule with the serial and parallel schedule generation schemes

Table 7.4 The parallel schedule generation scheme procedure to obtain the bottom schedule of Fig. 7.8

t	Eligible	Ranked	Scheduled
0	{2,3,4}	{4,3,2}	{3,4}
3	{2,8}	{8,2}	{2,3,4,8}
4	–	–	{2,3,4,8}
5	{5,6,7}	{5,7,6}	{2,3,4,6,7,8}
7	{5,11}	{5,11}	{2,3,4,6,7,8,11}
8	{5}	{5}	{2,3,4,6,7,8,11}
9	{5,9}	{5,9}	{2,3,4,5,6,7,8,11}
10	{9,10}	{10,9}	{2,3,4,5,6,7,8,10,11}
13	{9,12}	{9,12}	{2,3,4,5,6,7,8,9,10,11}
17	{12}	{12}	{2,3,4,5,6,7,8,9,10,11,12}
20	{12}	{13}	{2,3,4,5,6,7,8,9,10,11,12,13}

7.4.2 Lower Bounds

The purpose of lower bound calculations is to validate the quality of project schedules obtained by software tools or simple methods such as the priority rule based scheduling techniques presented earlier. While these heuristics were able to quickly present a precedence and resource feasible project schedule, a bound does not construct a feasible schedule but instead calculates a lower bound on the minimal project duration (upper bound in case of maximization problem). In the following subsections, three lower bound calculations will be explained.

Critical Path Based Lower Bound

Since the critical path method ignores the resource constraints and constructs an earliest start schedule with minimal project duration by only taking the precedence relations into account, it is obviously a lower bound on the RCPSP schedule. In the example project of Fig. 7.1, the critical path based lower bound is equal to 17 units resulting from the path “1-3-5-10-12-13-14”.

Resource Based Lower Bound

The resource based lower bound ignores the precedence relations of the project network and can therefore be considered as an inverse calculation of the critical path based lower bound. Since each activity is characterized by a duration and a renewable resource demand per resource type k , it can be expressed as an activity work content $r_{ik}d_i$. The work content, expressed in man-hours, man-days or man-weeks can be used to calculate a lower bound on the total project duration, as follows:

$$\max_k \left\lceil \frac{\sum_{i=1}^n r_{ik}d_i}{a_k} \right\rceil$$

where $\lceil x \rceil$ is used to calculate the smallest integer that is larger than x .

The resource based lower bound for the example project of Fig. 7.1 can be calculated as follows ($k = 1$): The numerator is equal to $6 * 7 + 5 * 1 + 3 * 5 + 1 * 6 + 3 * 1 + 2 * 2 + 1 * 2 + 4 * 8 + 3 * 6 + 1 * 2 + 3 * 6 + 5 * 8 = 187$ and hence the lower bound equals $\lceil \frac{187}{10} \rceil = 19$.

Critical Sequence Lower Bound

The critical sequence lower bound is a combination of the two previous lower bounds since it simultaneously considers both the precedence relations as well as the renewable resource constraints. The calculation of this bound consists of four steps, as follows:

1. Consider a critical path in the network.
2. Determine for each noncritical activity i how many time periods e_i this activity can be scheduled consecutively in parallel with the activities on the critical path (between its es_i and lf_i).
3. Set $e_i = \min(e_i, d_i)$ for each noncritical activity.
4. Increase the critical path length with the maximum of all $d_i - e_i$ values for the noncritical activities to obtain the critical sequence lower bound.

Figure 7.9 shows the critical activities 3, 5, 10, 12 and 13 resulting in a critical path with length of 17 time units. The remaining activities are displayed by means of an arc, showing their time window between their es_i and lf_i given the critical path length.

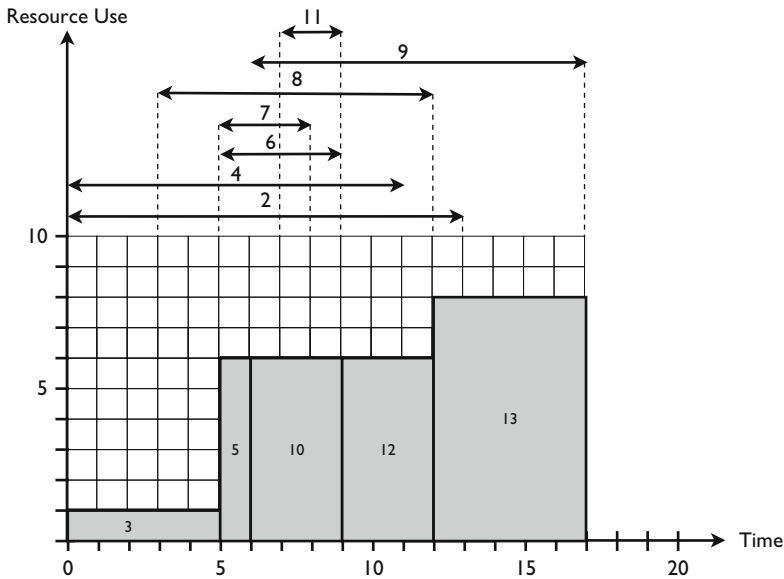


Fig. 7.9 The critical path of the example project and the time window for all noncritical activities

Table 7.5 Intermediate calculations to obtain the critical sequence lower bound value

Activity	d_i	es_i	lf_i	e_i	$d_i - e_i$
2	6	0	13	5	1
4	3	0	11	3	0
6	3	5	9	3	0
7	2	5	8	2	0
8	1	3	12	1	0
9	4	6	17	0	4
11	1	7	9	1	0

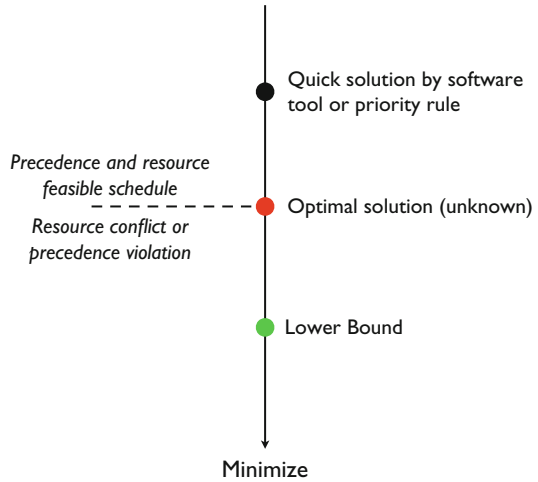
Table 7.5 displays the calculations of the e_i and shows the increase in the critical path length when each activity is individually added to the critical path schedule of Fig. 7.9. As an example, activity 2 can only be scheduled in parallel with the current critical path for 5 time units without violating the renewable resource constraints, while its duration is equal to 6. Consequently, $d_2 - e_2 = 6 - 5 = 1$ which means that the critical path length of 17 will increase by 1 time unit when introducing activity 2 in the schedule. All e_i and $d_i - e_i$ calculations are made for each noncritical activity individually (see the table) and the largest $d_i - e_i$ value is added to the current critical path value for the calculation of the lower bound. In the example, the critical sequence lower bound is equal to $17 + \max(d_i - e_i) = 17 + (d_9 - e_9) = 17 + 4 = 21$ time units.

7.4.3 Assessing Schedule Quality

It has been mentioned earlier that lower bounds can be used to assess the quality of a project schedule constructed by a software tool, a scheduling technique or any other procedure. Since lower bounds often report infeasible results (due to e.g. resource conflicts or violations of the precedence relations), they lead to a scheduling objective value that is unrealistically optimistic. Heuristic scheduling techniques, on the contrary, such as priority rule based scheduling methods, provide a resource feasible schedule with a realistic value for the scheduling objective, but with an unknown deviation from the theoretically best possible schedule. The gap between the two values for the scheduling objective gives an indication about the room for improvement, as shown in Fig. 7.10 for a minimization objective (such as time for the RCPSp).

In case of the example project of Fig. 7.1, the best solution obtained by all priority rule based scheduling rules can be considered as an upper bound for the total project duration, and equals 25 time units. The best lower bound is equal to 21. In this example, the gap between the two bounds does not provide much information. It might give the (false) impression that the heuristic solution can be further improved to lower values than 25, however, it has been shown earlier that this is not possible since it is the optimal minimal project duration. More advanced lower bound calculations, or other project network examples, might lead to smaller gaps and more precise estimates about the heuristic solution quality.

Fig. 7.10 Assessment of the project schedule quality



7.4.4 Other Scheduling Methods

Burgess and Killebrew Algorithm

The serial and parallel schedule generation schemes discussed in Sect. 7.4.1 are used to transform a priority rule into a resource feasible earliest start schedule aiming at minimizing a regular scheduling objective, e.g. the total project duration. However, since many of the real-life scheduling objectives are different from the time objective (see the previous sections of this chapter), another schedule generation scheme that is able to construct project schedules that deviate from a resource feasible earliest start schedule would be useful in case of a nonregular scheduling objective. The Burgess and Killebrew (B&K) algorithm presented in this section is a general, easy and effective algorithm to transform a priority rule into a project schedule where the variability in resource use is minimized (expressed by the resource leveling scheduling objective discussed in Sect. 7.3.4). It relies on the main characteristic of nonregular scheduling objectives, i.e. delaying an activity in time might lead to an improvement in the scheduling objective. The pseudo code of the B&K algorithm to transform a predefined priority rule into a resource feasible project scheduling with a resource leveling objective can be displayed as follows:

```

Schedule all activities at their early start time:  $s_i = e_{s_i}$ 
while change = 1
  For each activity  $i$  from the first one in the priority list to the last
     $CB = \infty$ 
    For each possible start time  $t$  of activity  $i$  from  $s_i$  to  $l_{s_i}$ 
      Calculate the scheduling objective  $O_{it}$ 
      If  $O_{it} \leq CB$  then  $CB = O_{it}$  and  $s_i^{best} = t$ 
      If  $s_i^{best} \geq s_i$  then  $s_i = s_i^{best}$  and change = 1
  
```

with

s_i	Current start time of activity i
es_i	Earliest start time of activity i
ls_i	Latest start time of activity i (given the s_i of all successor activities!)
O_{it}	Scheduling objective value when activity i starts at time t = calculation of the nonregular objective value when all other activities remain unchanged in the schedule
CB	Scheduling objective value of current best schedule found
s_i^{best}	Best start time of activity i
$change$	0/1 variable denoting a change or not = defines the stop criterion of the while loop in the B&K algorithm

The reader should have noted that the B&K algorithm does not explicitly take the renewable resource constraints into account. The aim of the algorithm is to schedule the project activities in time such that the variability of resource use is leveled to the best possible, without constraining the maximal use per time specified in a resource availability.

The B&K algorithm can be illustrated on the example project of Fig. 7.1 where the scheduling objective is to level the use of resources. It is assumed that the scheduling objective is calculated as $f_k(\sum_{i \in S(t)} r_{ik}) = (\sum_{i \in S(t)} r_{ik})^2$ which measures the leveling objective as a total sum of the squared resource use, with a weighting factor $w_k = 1$. The priority rule used is a randomly generated priority list equal to [12, 10, 3, 4, 2, 8, 13, 11, 7, 6, 9, 5] (start and end nodes are excluded from the list). The earliest start times of the nondummy activities are equal to $es_2 = 0, es_3 = 0, es_4 = 0, es_5 = 5, es_6 = 5, es_7 = 5, es_8 = 3, es_9 = 6, es_{10} = 6, es_{11} = 7, es_{12} = 9$ and $es_{13} = 12$ (see Fig. 7.2). The current start times of all activities are initially set to their earliest start times. It is assumed that a project deadline of 25 is set, and the latest start times are calculated as $ls_2 = 0, ls_3 = 0, ls_4 = 0, ls_5 = 5, ls_6 = 6, ls_7 = 5, ls_8 = 11, ls_9 = 21, ls_{10} = 6, ls_{11} = 8, ls_{12} = 9$ and $ls_{13} = 20$. Note that these latest start times are not calculated as done in Chap. 2 using backward calculations, but instead are calculated *given* the current start times of all successor activities. As an example, ls_9 is equal to 21, since it has no successors (except the dummy end activity 14 which starts and ends at the project deadline 25) while ls_{12} is equal to 9 since its successor activity 13 currently starts at $s_{13} = 12$. Consequently, the latest start times need to be continuously updated when activities shift in time. The B&K procedure runs as follows (only activities that are shifted in time are shown):

Run 1: $ls_2 = 0, ls_3 = 0, ls_4 = 0, ls_5 = 5, ls_6 = 6, ls_7 = 5, ls_8 = 11, ls_9 = 21, ls_{10} = 6, ls_{11} = 8, ls_{12} = 9$ and $ls_{13} = 20$. The current schedule has a total objective value of 2,289. This value is calculated as the sum of the squared resource use for all activities shown in Fig. 7.2 and is equal to $13^2 + 13^2 + 13^2 + 10^2 + 8^2 + 16^2 + 17^2 + 17^2 + 14^2 + 14^2 + 6^2 + 6^2 + 8^2 + 8^2 + 8^2 + 8^2 + 8^2$.

- Shift activity 8 from 3 to 11 and update $l_{s_4} = 8$ and $l_{s_8} = 11$
- Shift activity 13 from 12 to 20 and update $l_{s_8} = 19$, $l_{s_{12}} = 17$ and $l_{s_{13}} = 20$
- Shift activity 11 from 7 to 8 and update $l_{s_7} = 6$ and $l_{s_{11}} = 8$
- Shift activity 9 from 6 to 16 and update $l_{s_2} = 10$ and $l_{s_9} = 21$

Run 2: $l_{s_2} = 10$, $l_{s_3} = 0$, $l_{s_4} = 8$, $l_{s_5} = 5$, $l_{s_6} = 6$, $l_{s_7} = 6$, $l_{s_8} = 19$, $l_{s_9} = 21$, $l_{s_{10}} = 6$, $l_{s_{11}} = 8$, $l_{s_{12}} = 17$ and $l_{s_{13}} = 20$. The current schedule has a total objective value of 1,797.

- Shift activity 12 from 9 to 13 and update $l_{s_6} = 10$, $l_{s_{10}} = 10$, $l_{s_{11}} = 12$ and $l_{s_{12}} = 17$
- Shift activity 10 from 6 to 10 and update $l_{s_5} = 9$ and $l_{s_{10}} = 10$
- Shift activity 4 from 0 to 7 and update $l_{s_4} = 8$
- Shift activity 8 from 11 to 15 and update $l_{s_4} = 12$ and $l_{s_8} = 19$
- Shift activity 11 from 8 to 9 and update $l_{s_7} = 7$ and $l_{s_{11}} = 12$
- Shift activity 7 from 5 to 6 and update $l_{s_7} = 7$
- Shift activity 6 from 5 to 6 and update $l_{s_6} = 10$
- Shift activity 5 from 5 to 6 and update $l_{s_3} = 1$ and $l_{s_5} = 9$

Run 3: $l_{s_2} = 10$, $l_{s_3} = 1$, $l_{s_4} = 12$, $l_{s_5} = 9$, $l_{s_6} = 10$, $l_{s_7} = 7$, $l_{s_8} = 19$, $l_{s_9} = 21$, $l_{s_{10}} = 10$, $l_{s_{11}} = 12$, $l_{s_{12}} = 17$ and $l_{s_{13}} = 20$. The current schedule has a total objective value of 1,419.

- Shift activity 3 from 0 to 1 and update $l_{s_3} = 1$
- Shift activity 7 from 6 to 7 and update $l_{s_7} = 7$
- Shift activity 6 from 6 to 10 and update $l_{s_6} = 10$

Run 4: $l_{s_2} = 10$, $l_{s_3} = 1$, $l_{s_4} = 12$, $l_{s_5} = 9$, $l_{s_6} = 10$, $l_{s_7} = 7$, $l_{s_8} = 19$, $l_{s_9} = 21$, $l_{s_{10}} = 10$, $l_{s_{11}} = 12$, $l_{s_{12}} = 17$ and $l_{s_{13}} = 20$. The current schedule has a total objective value of 1,411.

- No further shifts could be found. The schedule is given in Fig. 7.7.

7.5 Scheduling Extensions

In the previous sections, a number of implicit assumptions have been made during the scheduling of activities, such as a fixed activity duration and resource use, no splitting possibilities of activities, a fixed resource availability over the complete scheduling horizon and the absence of project and resource calendars.

In many practical cases, these assumptions can be relaxed to more general settings, and hence, over-allocations of renewable resources can be eliminated by using other methods. In the next subsections, a number of extensions are discussed that can influence the resource feasibility of the schedule.

7.5.1 Variable Resource Availability

In the previous sections, it is assumed that the availability of the renewable resources is a predefined constant over the entire project horizon. More precisely, a_k was used to denote the amount of available resources for resource type k . Alternatively, the total availability of the resource availability may vary over the total time horizon of the project, denoted by a_{kt} (the amount of available resources of resource type k at time instance t).

7.5.2 Multi-mode: A Time/Resource Trade-Off

In the previous sections, each activity duration and renewable resource demand are assumed to be fixed values which cannot be changed during the scheduling step. This so-called “fixed duration” mode can be extended to a “fixed work” mode by fixing the product of the two values to a fixed work content. In practical projects, the workload of project activities is often estimated by its work content (e.g. amount of man-days) such that several duration/resource demand combinations are possible for each individual project activity. Given an activity’s work content, a set of possible duration/resource demand combinations can be specified during the project scheduling phase, each characterized by a fixed duration and a constant resource requirement.

Table 7.6 shows all possible combinations for the example project of Fig. 7.1. Combinations with renewable resource demand higher than its availability $a_1 = 10$ can be removed, leading to the remaining set of combinations among which one needs to be chosen. In case the scheduling objective is the minimization of

Table 7.6 All possible (duration, resource demand) combinations for the example project of Fig. 7.1 under a fixed work mode

Activity	Predecessors	Work content	(Duration, Resource use)
1	–	0	(0,0)
2	1	42	(1,42) , (2,21) , (3,14) , (6,7), (7,6) , (14,3), (21,2), (42,1)
3	1	5	(1,5), (5,1)
4	1	15	(1,15) , (3,5), (5,3) , (15,1)
5	3	6	(1,6), (2,3) , (3,2), (6,1)
6	3	3	(1,3), (3,1)
7	3	4	(1,4) , (2,2), (4,1)
8	4	2	(1,2) , (2,1)
9	2	32	(1,32) , (2,16) , (4,8), (8,4) , (16,2), (32,1)
10	5	18	(1,18) , (2,9), (3,6) , (6,3), (9,2), (18,1)
11	7	2	(1,2), (2,1)
12	6, 10, 11	18	(1,18) , (2,9), (3,6) , (6,3), (9,2), (18,1)
13	8, 12	40	(1,40) , (2,20) , (4,10) , (5,8), (8,5), (10,4), (20,2), (40,1)
14	9, 13	0	(0,0)

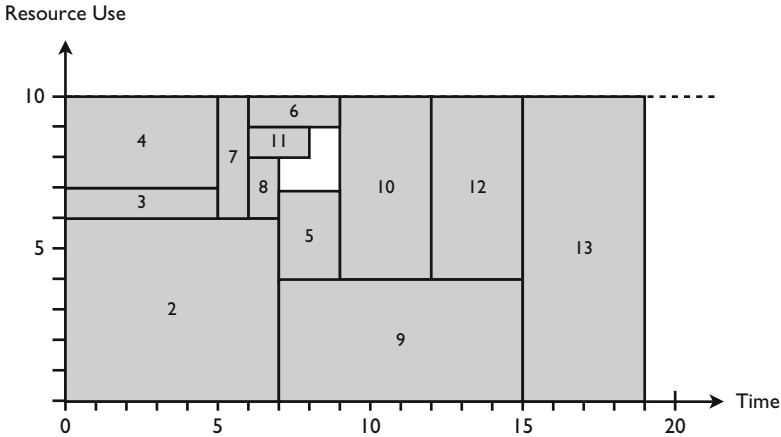


Fig. 7.11 Multi-mode resource-constrained project schedule

time (i.e. the RCPSP), the selected combinations of duration and renewable resource use are indicated in bold, leading to an overall project duration of 19 time units. Figure 7.11 shows the resource graph for the solution of Table 7.6.

7.5.3 Others

In the previous sections, resource conflicts, i.e. over-allocations of renewable resources have been eliminated by using two possible methods, as follows:

- Shifting a task forwards or backwards.
- Reducing the duration of an activity within a fixed work content, known as multi-mode scheduling.

However, the project manager has access to a wide range of alternative methods, which will not be discussed in detail in this book, such as:

- Splitting a task, known as activity preemption.
- Adjusting the resources assigned to a task.
- Assigning more resources to a task.
- Replacing an over-allocated resource with an under-allocated one or removing a resource.
- Adjusting or contouring the amount of work assigned to a resource.

The interested reader is referred to two classification schemes of project scheduling, which both give an extensive overview of various project scheduling problems with and without renewable resources (Brucker et al. 1999; Herroelen et al. 1999). A further discussion on various extensions is outside the scope of this book. However, in Chap. 8, a number of extensions to realistic settings are discussed in detail.

7.6 Resource Cost

The calculation of costs is an important aspect of project scheduling and control. In this section, four main types of costs that should be linked to the project activities and their resources are discussed.

7.6.1 Types of Costs

The total cost of an activity is a sum of various cost factors that are classified in this subsection into four classes. Two are linked to the activity without the use of resources while the two others are due to the use of resources per activity. Obviously, more than one type of cost can be defined for a single resource. The various cost types operate in differing ways, depending upon whether the resource is a renewable or nonrenewable resource. In this section, the variable r_{ik} used in the previous section of this chapter is split up into a r_{ik}^r and r_{ik}^n variable to distinguish between the activity demand of renewable and nonrenewable resources. Both variables express the number of units required by activity i for the resource k . While the r_{ik}^r variable is always used to express the use per time unit, the total use r_{ik}^n of a nonrenewable resource can be a fixed value over the duration of the activity (*fixed use*) or can be a variable value to express the use per time (*variable use*) similar to the renewable resource use r_{ik}^r . Details on the four cost classes are given along the following lines.

Activity Cost

Fixed cost: A fixed cost can be assigned to an activity regardless of its use of renewable and nonrenewable resources. It is a fixed amount of money that is independent of the duration of the activity and the work content for one or more resources assigned to this activity.

Variable cost: A variable cost per time unit can be assigned to an activity regardless of its use of renewable and nonrenewable resources. It is a variable amount of money that is dependent on the activity duration and can be used to describe resource-independent costs such as activity overhead.

Resource Cost

Cost per use: The cost for the use of a resource can be considered as a one-time cost incurred every time that the resource is used by the activity. The per use cost for a nonrenewable resource is applied only once from the moment the resource is used. The per use cost for a renewable resource, however, depends on the resource

demand of the activity (i.e. its resource requirement r_{ik}^r and *not* its total work content $r_{ik}^r * d_i$).

As an example, the cost per use for a nonrenewable resource like gallons of gasoline needed to feed excavators during a construction project is €250 per delivery, regardless of the amount of gallons brought per delivery per truck (obviously, this is cost of delivery and *not* the cost for the gasoline itself, which will be measured by the cost per unit discussed hereafter). The excavator itself, on the contrary, is a renewable resource and has a per use cost of €1,000. Working with three excavators in parallel to finish the activity makes the total per use cost equal to €3,000.

Cost per unit: Unlike the one-time cost per use, the cost per unit is a cost that typically depends on the amount of the resource demanded by the activity and its duration. The calculation of the total activity per unit cost differs along the resource type, renewable or nonrenewable, as explained along the following lines:

- Renewable resources: the costs per unit are cost rates calculated per time unit (hours, days, weeks, etc.) and per resource unit and hence are based on the total work content $r_{ik}^r * d_i$ of the activity i for the renewable resource k .
- Nonrenewable resources: Normally, the use of nonrenewable resources by project activities is expressed in units that are typically different from time units. Consequently, the costs per unit are monetary rates not calculated per hour but expressed in other units (per weight, per length, per pallet, etc.). However, there are examples where nonrenewable resource use is expressed as a time dimension. To that purpose, the assignment of a nonrenewable resource to an activity can be done in two alternative ways:
 - Fixed use: the unit for the cost/unit calculation of a nonrenewable resource is anything but a time dimension. It is an indication that the quantity of the resource used by an activity is independent from its duration. A typical example is the cost per unit for materials, such as bricks needed to build a wall, which is equal to €100 per pallet, regardless how much time it takes to build that wall. The gasoline example for the excavator could also be considered as a nonrenewable resource with fixed use, since its cost completely depends on the amount used, expressed in gallons, to finish a certain activity.
 - Variable use: the unit for the calculation of the cost/unit for a nonrenewable resource can be expressed in a time dimension (hours, days, ...). It is an indication that the quantity of resources used by an activity changes proportionally as its duration changes. In the gasoline example, the resource use could have been specified as the number of gallons needed per day. In this case, it is implicitly assumed that the daily occupation of the excavator is known and more or less stable, and the nonrenewable resource of gasoline then depends on the daily gasoline consumption of the excavator, and hence, on the number of days it takes to finish that activity with the help of that excavator.

Table 7.7 The calculation of activity costs from their resource use

Cost	Renewable resource = <i>limited availability</i> during scheduling	Nonrenewable resource = <i>no limited availability</i> during scheduling
c_{use}	$c_{use} * r_{ik}^r$	c_{use}
c_{unit}	$c_{unit} * d_i * r_{ik}^r$	variable: $c_{unit} * d_i * r_{ik}^n$ fixed: $c_{unit} * r_{ik}^n$

The calculations of the costs for the two types of resources is summarized in Table 7.7, which gives a clear indication of the difference in total cost calculations depending on units of resource demands. The total activity cost is equal to the sum of these two resource type costs increased by the resource independent fixed and variable activity cost.

7.6.2 Cost Sensitivity

It has been mentioned in Chap. 5 that the sensitivity of project activities in a baseline schedule is of crucial importance and offers insight into the strengths and weaknesses of the baseline schedule and serves as a tool to trigger and control a project manager's corrective actions during project progress. However, Chap. 5 was restricted to a schedule risk analysis (SRA) where only the time sensitivity of project activities was analyzed. However, the sensitivity of the cost dimension of a project, and the influence of changes in the baseline schedule on the total cost of the project is of equal, if not more, importance. Table 7.7 shows that the SRA studies of Chap. 5 can also provide cost sensitivity information, since changes in activity durations have an effect on the total activity cost, which depends on the cost type (per unit versus per use) and the resource type (renewable versus nonrenewable). Consequently, variation in activity duration leads to variations in the total activity cost, and hence, a SRA study will also provide insight into the sensitivity of the activity and resource costs of the project.

7.7 Conclusions

This chapter gave an overview of resource-constrained project scheduling techniques under different scheduling objectives. It has been shown that the choice of an appropriate scheduling objective is one of the main decisions that needs to be made during the project scheduling phase, since it determines the look and shape of the project schedule, the critical chain and the efficiency of resource use.

Due to the inherent complexity of scheduling projects with renewable resources under a limited availability, many quick and easy heuristic methods have been developed. In the current chapter, priority based scheduling has been proposed as

a heuristic method to schedule projects with a regular and nonregular scheduling objective. The obtained project scheduling objective values can be compared with lower bounds to validate and assess the quality of the schedule, and to determine room for possible improvement.

Extensions to more practically oriented project scheduling settings have been briefly described in the chapter. In practice, a never-ending list of undiscussed extensions is possibly leading to more relevant tools and techniques for certain projects. However, the general scheduling principles and lessons learned in this chapter will remain unchanged. More advanced resource-constrained scheduling topics will be discussed in the next chapter.