1-1999

A classification scheme for project scheduling problems

Willy HERROELEN

Erik DEMEULEMEESTER

Bert DE REYCK
Singapore Management University, bdreyck@smu.edu.sg

1 A CLASSIFICATION SCHEME FOR PROJECT SCHEDULING

Willy Herroelen[1]
Erik Demeulemeester[1]
Bert De Reyck[2]

1. Katholieke Universiteit Leuven (Belgium)
2. Erasmus University Rotterdam (The Netherlands)

1.1. Introduction

The basic concern of scheduling is commonly described as the allocation of limited resources to tasks over time (Lawler et al. 1993, Pinedo 1995). The resources and tasks may take many forms. In project scheduling the tasks refer to the activities belonging to one or more projects. The execution of project *activities* may require the use of different types of resources (money, crews, equipment, ...). The scheduling *objectives* may also take many forms (minimizing project duration, minimizing project costs, maximizing project revenues, optimizing due date performance, ...). The result is a wide and steadily growing variety of problem types which motivates the introduction of a systematic notation that can serve as a basis for a classification scheme.

A classification scheme for project scheduling may serve a variety of objectives. First, a classification scheme would greatly facilitate the presentation and discussion of project scheduling problems. Intensive research efforts over the past few years have greatly expanded the variety of project scheduling problems under study (for recent reviews we refer to Icmeli et al. 1993, Elmaghraby 1995, Herroelen and Demeulemeester 1995, Özdamar and Ulusoy 1995, Herroelen et al. 1997, 1998). These problems are often identified within the project scheduling community in a non-standardized manner by a rather confusing set of acronyms, most often consisting of a simple concatenation of characters. Examples are *RCPSP* for the resource-constrained project scheduling problem, *MRCPSP* for the multi-mode resource-constrained project scheduling problem, *RCPSP-GPR* for the resource-constrained project scheduling problem with generalized precedence relations, just to cite a few. A concise and rigorous classification scheme immediately highlights the fundamental problem characteristics and avoids the use of these lengthy and often ambiguous character concatenations. In addition, it saves both authors and speakers the repetitive use of lengthy, verbal, introductive statements about the precise characteristics and assumptions of the project scheduling problem under study.

Second, a comprehensive classification scheme allows for the immediate identification of viable areas of research through the identification of interesting open problems which have remained unstudied or largely ignored by the researchers in a fastly growing field. It helps in identifying the common characteristics of project scheduling problems and reveals the important fact that certain problems are in fact subproblems of more generic ones.

Third, a classification scheme simplifies the assessment of problem complexity. It reveals the close relationships between the various project scheduling problems through the use of reduction graphs which show the various interrelations among the different values of the particular classification parameters. As such it helps in identifying the fundamental characteristics which account for the inherent complexity of the problem under study.

Last, but not least, a classification scheme facilitates the match of solution procedures to problem settings and as such facilitates the preparation of problem surveys and literature reviews.

It is common practice to classify deterministic machine scheduling problems by a standard three-field notation proposed by Graham et al. (1979) and

Blazewicz et al. (1983). The extensive scheme we propose in this chapter resembles the standard scheme for machine scheduling problems in that it is also composed of three fields $\alpha | \beta | \gamma$. In machine scheduling problems (Blazewicz et al. 1994, Brucker 1995) the first field α describes the machine environment. This field allows for the identification of single machine problems, various types of parallel machine problems, flow shops, job shops, general shops, open shops, mixed shops and multiprocessor task systems. The second field β is used to describe the task and resource characteristics. This field includes parameter settings for characterizing the possibility for preemption, the precedence constraints, ready times, deadlines, task processing times, batching and additional resources. The characterization of the additional resources (Blazewicz et al. 1986) is done using a field parameter which takes the value of the empty symbol ° to denote the absence of additional resources and $res\lambda\sigma\rho$ to specify the resource constraints. $\lambda, \sigma, \rho \in \{\cdot, k\}$ respectively denote the number of additional resource types, resource limits and resource requests. If $\lambda, \sigma, \rho = \cdot$, then the number of additional resource types, resource limits and resource requests are arbitrary, and if $\lambda, \sigma, \rho = k$, then the number of additional resource types is equal to k, each resource is available in the system in the amount of k units and the resource requests of each task are at most equal to k units. The third field γ denotes a performance measure.

The characterization of the additional resources in the $\alpha | \beta | \gamma$ notation of the deterministic machine scheduling classification scheme, however, does not allow for the precise characterization of the wide variety of problems which manifest themselves in the specific and much more complex environment of *project* scheduling. This motivates the introduction of a specific classification scheme for project scheduling problems. The scheme proposed in this chapter is also based on a three-field notation, but the composition of the fields and the precise meaning of the various parameters, however, are mostly new and specific to the field of project scheduling. It should be understood from the outset, however, that our objective is not to build an extremely rigid classification scheme which attempts to create futile classification holes to accommodate any possible project scheduling problem. The proposed scheme tries to combine rigidity with flexibility. It provides sufficient detail to allow for a concise taxonomy of the project scheduling field which covers the majority of the project scheduling problems described in the literature and at the same time offers sufficient degrees of freedom to the user in the specification of the various parameters.

The organization of this chapter is as follows. The classification scheme is presented in the next section, in which we also discuss the simple reductions between the various project scheduling problems and problem parameters. We present the reduction graphs which can be defined on the field parameters of the classification scheme. In Section 1.3 we illustrate the potential use of the scheme by using the various parameters in an effort to characterize and classify the most important project scheduling problems commonly discussed in the literature. The last section is then reserved for overall conclusions.

1.2. Classification of project scheduling problems

The classification scheme is composed of three fields $\alpha|\beta|\gamma$. The meaning of the three fields is explained below.

1.2.1. Field α: resource characteristics

The resource characteristics of a project scheduling problem are specified by a set α containing at most three elements α_1, α_2 and α_3. Let $°$ denote the empty symbol which will be omitted when presenting specific problem types. Parameter $\alpha_1 \in \{°, 1, m\}$ denotes the number of resource types:

$\alpha_1 = °$: no resource types are considered in the scheduling problem,
$\alpha_1 = 1$: one resource type is considered,
$\alpha_1 = m$: the number of resource types is equal to m.

Parameter $\alpha_2 \in \{°, 1, T, 1T, v\}$ denotes the specific resource types used. In the project scheduling literature a common distinction is made (Blazewicz et al. 1986) between renewable resources, nonrenewable resources and doubly-constrained resources. *Renewable resources* (e.g. manpower, machines, tools, equipment, space, ...) are available on a period-by-period basis, that is, the available amount is renewed from period to period. Only the total resource usage at every time instant is constrained. *Nonrenewable* resources (e.g. money, raw materials, energy, ...), on the contrary, are available on a total project basis, with a limited consumption availability for the entire project. *Doubly-constrained resources* are constrained per period (e.g. per period cash flow) as well as for the overall project (e.g. total expenditures, overall pollution limits, ...). Recently, researchers (Böttcher et al. 1996, Schirmer and Drexl 1996, Drexl 1997) have introduced the concept of *partially (non)renewable resources* referring to resources the availability of which is defined for a specific time interval (subset of periods). For each resource type there are a number of subsets of periods, each characterized by a specific (nonrenewable) availability of the resource type. Essentially, partially (non)renewable resources can be viewed as a generic resource concept in project scheduling, as they include both renewable and nonrenewable (and, hence, also doubly-constrained) resources. A partially renewable resource with a specified availability for a time interval equal to the unit duration period (identified below by the parameter setting equal to 1) is essentially a renewable resource. A partially renewable resource with a specified availability for a time interval equal to the project horizon (identified below by a parameter setting equal to T) is essentially a nonrenewable resource. Partially renewable resources with a specified availability on both a unit duration and a total project horizon basis (denoted in the parameter setting by $1T$) can be interpreted as doubly-constrained resources. As a result we can use the partially renewable resource concept in our classification scheme in a generic way which allows for a straightforward identification of the various resource categories considered:

$\alpha_2 = \circ$: absence of any resource type specification,
$\alpha_2 = 1$: renewable resources, the availability of which is specified for the unit duration period (e.g. hour, shift, day, week, month, ...),
$\alpha_2 = T$: nonrenewable resources, the availability of which is specified for the entire project horizon T,
$\alpha_2 = 1T$: both renewable and nonrenewable resources (also called doubly-constrained resources the availability of which is specified on both a unit duration period and a total project horizon basis),
$\alpha_2 = v$: partially (non)renewable resources the availability of which is renewed in specific time periods, i.e. they are nonrenewable in variable time intervals (e.g. in total 5 units are available in the set of periods 1 up to 5, 7 units are available in period 6, in total 0 units are available in the set of periods 7 up to 9, ...).

Parameter $\alpha_3 \in \{\circ, va\}$ describes the resource availability characteristics of the project scheduling problem. Some scheduling problems assume that renewable resource availabilities are a given constant while others assume that resource availability varies over time. We assume that the availability of partially renewable resources may vary over the various time intervals. The following parameter specifications are used in our classification scheme:

$\alpha_3 = \circ$: (partially) renewable resources are available in constant amounts,

$\alpha_3 = va$: (partially) renewable resources are available in variable amounts.

Stochastic settings for the resource availability characteristics are discussed in Section 1.3.3.

The simple reductions between the various resource parameters are shown by the reduction graphs in Figure 1:1. The nodes in the graph represent particular assumptions made about the parameters. The directed arcs show the direction of polynomial transformations.

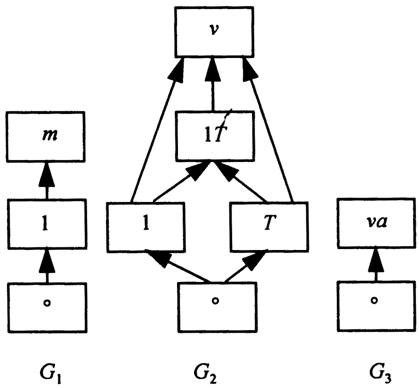

Figure 1:1. Simple reductions for the resource parameters

First consider the reduction graph G_1. If we replace ° by 1 in the specification for α_1, we get a simple reduction because the problem without any resource constraints is a special case of the problem with a single resource type. In a similar fashion, replacing 1 by m yields a simple reduction as we move from a problem with a single resource type to a problem which uses m resource types. The reduction graph G_2 specifies the reductions for α_2. Replacing ° by 1 or T yields a simple reduction because we move from a problem without any resource specification to a problem which involves either renewable or nonrenewable resources. Both 1 and T reduce to $1T$, since both the renewable and nonrenewable resources are a special case of a doubly-constrained resource. The case where the partially renewable resources are renewed in specific time periods constitutes a special case of 1, T and $1T$. The reduction graph G_3 denotes the simple reduction from the case of constant resource availabilities to the situation where resources are available in time varying amounts.

1.2.2. Field β: activity characteristics

The second field β specifies the activity characteristics of a project scheduling problem. It contains at most nine elements β_1, β_2, β_3, β_4, β_5, β_6, β_7, β_8 and β_9. Parameter $\beta_1 \in \{°, \mathit{pmtn}, \mathit{pmtn\text{-}rep}\}$ indicates the possibility of activity preemption:

$\beta_1 = °$: no preemption is allowed,
$\beta_1 = \mathit{pmtn}$: preemptions of the preempt-resume type are allowed,
$\beta_1 = \mathit{pmtn\text{-}rep}$: preemptions of the preempt-repeat type are allowed.

Preemption (activity splitting) implies that the processing of an activity may be interrupted and resumed at a later time (preempt-resume). The situation where activities may be interrupted but cannot be resumed at the point of interruption, i.e., must be completely redone (preempt-repeat), can be accommodated by setting $\beta_1 = \mathit{pmtn\text{-}rep}$.

Parameter $\beta_2 \in \{°, \mathit{cpm}, \mathit{min}, \mathit{gpr}, \mathit{prob}\}$ reflects the precedence constraints:

$\beta_2 = °$: no precedence constraints (the activities are unordered),
$\beta_2 = \mathit{cpm}$: the activities are subject to strict finish-start precedence constraints with zero time lag, as used in the basic PERT/CPM model,
$\beta_2 = \mathit{min}$: the activities are subject to precedence diagramming constraints of the type start-start, finish-start, start-finish and finish-finish with *minimal* time lags,
$\beta_2 = \mathit{gpr}$: the activities are subject to generalized precedence relations of the type start-start, finish-start, start-finish and finish-finish with both *minimal* and *maximal* time lags.

We deem it necessary to make an explicit distinction between the use of minimal and maximal time lags. A minimal time lag specifies that an activity can only start (finish) when the predecessor activity has already started (finished) for a certain time period. A maximal time lag specifies that an activity should be started (finished) at the latest a certain number of time periods beyond the start (finish) of another activity. A minimal time lag is essentially a generalized precedence relation which can be transformed into a non-negative start-start time lag, with the additional assumption that no cycles may occur in the network. The introduction of maximal time lags offers a wide amount of relevant and practical modelling capabilities (De Reyck 1995) which are well beyond the scope of minimal time lags. The introduction of both minimal and maximal time lags, however, also shifts the problem setting to a much higher complexity level (quite often, the feasibility problem already becomes NP-complete). The fact that many commercial project planning software packages do not allow for the use of maximal time lags does not come as a surprise.

β_2 = *prob:* the activity network is of the probabilistic type where the evolution of the corresponding project is not uniquely determined in advance.

This category encompasses generalized activity networks (Elmaghraby 1977) such as GERT (Neumann and Steinhardt 1979). This field leaves the user the freedom to be very specific about the precise type of probabilistic network in use. As such, the β_2-parameter can be set to *gert* to specify a GERT network, to *deor* to specify GERT networks with exclusive-or node entrance and deterministic node exit, to *steor* to specify GERT networks with exclusive-or node entrance and stochastic node exit, etc.

The third parameter $\beta_3 \in \{°, \rho_j\}$ describes ready times:

β_3 = °: all ready times are zero,
$\beta_3 = \rho_j$: ready times differ per activity.

Parameter $\beta_4 \in \{°, cont, d_j=d\}$ describes the duration of the project activities:

β_4 = °: activities have arbitrary integer durations,
β_4 = *cont*: activities have arbitrary continuous durations,
$\beta_4 =(d_j=d)$: all activities have a duration equal to d units.

The specification of stochastic activity durations is discussed in Section 1.3.3.

Parameter $\beta_5 \in \{°, \delta_j, \delta_n\}$ describes deadlines:

β_5 = °: no deadlines are assumed in the system,
$\beta_5 = \delta_j$: deadlines are imposed on activities,
$\beta_5 = \delta_n$: a project deadline is imposed.

It should be noted that the specification $\beta_5 = \delta_j$ also allows deadlines to be imposed on *events* in the network. Events are identified as the initial or terminal point of one or several activities and can easily be accommodated by the network logic. Moreover, we have opted not to specify the use of activity (event) or project *due dates* in the β_5-field. A *deadline* indicates that activities (events) or projects must finish not later than the deadline and may not be violated. The eventual use of due dates which may be violated at a specific cost should be apparent from the specification of the corresponding due-date based performance criterion in the γ-field of the classification scheme.

Parameter $\beta_6 \in \{°, vr, disc, cont, int\}$ denotes the nature of the resource requirements of the project activities. A common assumption is that activities request their resources in constant amounts or in variable amounts over their periods of execution. Some models (e.g. Weglarz 1980) assume that the resource requests of the tasks are continuous, i.e. concern resource amounts which are arbitrary elements of given intervals. For example, if a resource request of a task is characterized by the interval $(0,N]$, it means that an arbitrary amount of this resource greater than 0 and not greater than N can be used for processing the task at any moment. Such resource requests concern continuously-divisible resources like electric current (or power), fuel flow, etc. Other models study the simultaneous requirement of discrete and continuous resources (e.g. Jozefowska and Weglarz 1994, Weglarz and Jozefowska 1997). Still other models (Hackman and Leachman 1989, Leachman 1983, Leachman et al. 1990) assume that there is a feasible range of *intensity* of resource assignments to each activity, resulting in a range of possible durations. These models assume that all the different types of resources which are required by an activity are applied proportionally throughout the activity execution, i.e. each activity utilizes a constant mix of resources as it progresses exactly proportional to the mix of total resource requirements to complete the activity. The rates of applications of different types of resources to an activity can be indexed in terms of one rate which is called the *intensity of the activity*. The activity intensity is assumed to be continuously variable within given upper and lower limits. The rate of progress of an activity is assumed to be proportional to its intensity. Both types of models essentially boil down to similar problem environments in which the resource requirement of an activity is a function of time. Our classification scheme uses the following specifications for the resource requirements:

β_6 = °: activities require the resources in a constant discrete amount (e.g. a number of units for every time period of activity execution),
β_6 = *vr*: activities require the resources in variable discrete amounts (e.g. a number of units which varies over the periods of activity execution).

Stochastic resource requirement settings are discussed in Section 1.3.3.

For those cases where the activity durations have to be determined by the solution procedure on the basis of a resource requirement function, the following settings are used:

β_6 = *disc*: the activity resource requirements are a discrete function of the activity duration,
β_6 = *cont*: the activity resource requirements are a continuous function of the activity duration,
β_6 = *int:* the activity resource requirements are expressed as an intensity or rate function.

We leave it up to the user to be more specific in the specification of the resource requirement function. If so desired, the setting β_6 = *cont* can be made more specific: β_6 = *lin* can be used to specify that activity resource requirements are a linear funtion of the activity duration, the setting β_6 = *conc* can be used to specify that the activity resource requirements are a concave function of the activity duration, and β_6 = *conv* may denote the fact that resource requirements are a convex function of the activity duration.

The type and number of possible execution modes for the project activities is described by parameter $\beta_7 \in \{°, mu, id\}$. Most problems assume a single execution mode per activity. Various problems assume time/cost, time/resource and/or resource/resource trade-offs which give rise to various possible execution modes for the activities of the project. Recently, researchers (Salewski 1996, Salewski and Lieberam-Schmidt 1996, Salewski et al. 1997) have started to study project scheduling problems which generalize multiple activity modes to so-called mode identity constraints in which the set of activities is partitioned into disjoint subsets. All activities in a subset must then be executed in the same mode. Both the time and cost incurred by processing a subset of activities depend on the resources assigned to it. The following parameter settings are used:

β_7 = °: activities must be performed in a single execution mode,
β_7 = *mu*: activities have multiple prespecified execution modes,
β_7 = *id*: the activities are subject to mode identity constraints.

Parameter $\beta_8 \in \{°, c_j, per, sched\}$ is used to describe the financial implications of the project activities. In most models with cash flows, the cash flow amounts are assumed to be known and are either associated with network activities or network events. Both situations can be represented by associating the cash flows with the nodes in an activity-on-the-node network. Other models assume that the cash flows are periodic in that they occur at regular time intervals or with a known frequency. Still other models assume that both the amount and the timing of the cash flows have to be determined (Herroelen et al. 1997). The following settings are used in the classification scheme:

$\beta_8 = °$: no cash flows are specified in the project scheduling problem,
$\beta_8 = c_j$: activities have an associated arbitrary cash flow,
$\beta_8 = c_j^+$: activities have an associated positive cash flow,
$\beta_8 = per$: periodic cash flows are specified for the project (e.g. payments at regular intervals),
$\beta_8 = sched$: both the amount and the timing of the cash flows have to be determined.

In Section 1.3.3. a brief discussion is given on the use of stochastic cash flow specifications.

A common assumption in project scheduling is that change-over times (i.e. setup times or transportation times) are sequence-independent and included in the activity durations. As recognized by several authors (e.g. Kaplan 1991, Kolisch 1995, Dodin and Elimam 1997, Bartsch et al. 1997), however, sequence-dependent change-over times may be very important in project settings. Examples include equipping excavators with different types of scoops, the travel of workers between jobs and the transportation of heavy equipment to different construction sites. Parameter $\beta_9 = \in \{°, s_{jk}\}$ is used to denote change-over times:

$\beta_9 = °$: no change-over (transportation) times,
$\beta_9 = s_{jk}$: sequence-dependent change-over times.

Mode-dependent change-over times are modelled implicitly.

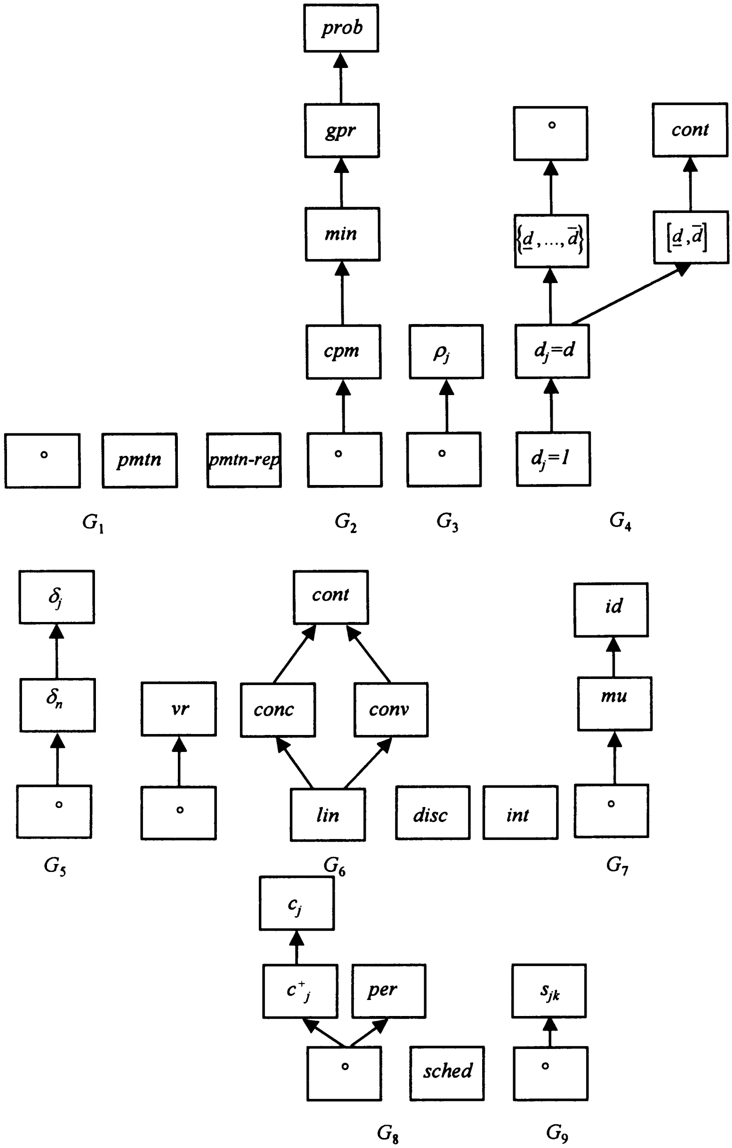

Figure 1:2. Simple reductions for the activity parameters

The reduction graphs in Figure 1:2 show the simple reductions between the various activity parameters. Reduction graph G_1 shows the case of a project scheduling problem in which preemptions are not allowed and the environment which does allow for preemption.

Reduction graph G_2 shows the simple reduction from a problem with unordered activities to a problem with finish-start precedence constraints with zero time lag, to a problem with precedence constraints with minimal time lags, to a problem with generalized minimal and maximal time lags. The latter reduces to the case where the network is probabilistic.

Reduction graph G_3 shows the simple reduction from a project scheduling problem in which all activities have a ready time (release date) equal to zero to a problem setting which specifies ready times which differ per activity. The simple reductions in reduction graph G_4 are for the duration parameters in the discrete and continuous case. A scheduling problem in which all activity durations are equal to 1 reduces to a scheduling problem with all activities having equal durations. This problem in turn reduces to a problem in which the given activity durations are bounded from below and above or ($\{\underline{d}, \ldots, \overline{d}\}$ or $[\underline{d}, \overline{d}]$), which can be reduced to a problem with arbitrary integer or continuous durations, respectively. The deadline reductions are shown in graph G_5. These reductions indicate that a problem without activity deadlines is a subproblem of a problem in which a deadline on the last activity (project duration) is imposed. The latter problem in turn reduces to a problem with individual activity deadlines.

Reduction graph G_6 describes the interrelations between the resource requirement characteristics. Problems in which activities have constant discrete resource requirements are subproblems of problems in which activities require the resources in variable discrete amounts. Problems with linear resource requirement functions are subproblems of problems in which the resource requirement functions are concave or convex. These in turn are subproblems of problems in which general resource functions are used. Reduction graph G_7 explains the reductions between the execution mode parameters. Single mode problems are subproblems of multi-mode problems, which reduce to mode-identity constrained problems.

Reduction graph G_8 shows the reduction between the cash flow parameters. A project scheduling problem in which no cash flows are specified for the activities (nodes in the network) reduces to a problem where activities have positive cash flows associated with the activities and a problem in which cash flows are specified on a periodic basis. The former reduces to the problem with arbitrary cash flows associated with the activities.

Finally, reduction graph G_9 shows the simple reduction from a project scheduling problem with no or sequence-independent change-over times to a problem where change-over times are sequence-dependent.

1.2.3. Field γ. performance measures

The third field γ is reserved to denote optimality criteria (performance measures). The performance measures are either *early completion measures* or *free completion measures*. *Early completion measures* (commonly denoted in the scheduling literature as *regular performance measures*) involve penalty functions which are nondecreasing in activity completion times (Conway et al. 1967). Common examples of early completion criteria often used in project scheduling are the minimization of the project duration (makespan), the minimization of the project lateness or tardiness and the minimization of project costs. A less common example used in project scheduling with discounted cash flows is the maximization of the project net present value under the assumption that only positive cash flows are considered (see Herroelen et al. 1997). There are, however, many applications in which *free completion measures* (commonly denoted as *nonregular performance measures* in the scheduling literature) are appropriate. A typical example is the maximization of the net present value of a project characterized by arbitrary cash flow values or the minimization of the weighted earliness-tardiness.

The following settings are used:

$\gamma = reg$: the performance measure is any early completion (regular) measure,
$\gamma = nonreg$: the performance measure is any free completion (nonregular) measure.

Obviously the list of possible performance measures is almost endless. We provide the user with sufficient degrees of freedom to introduce suitable measures through a proper setting of the parameter value or through the specification of the mathematical expression of the objective function(s). The following is a nonexhaustive list of example settings:

makespan:

$\gamma = C_{max}$: minimize the project makespan.

flow time:

$\gamma = \bar{F}$: minimize the average flow time over all subprojects or activities.

due date performance:

$\gamma = L_{max}$: minimize the project lateness (i.e. the maximum of the lateness of the subprojects or activities),
$\gamma = T_{max}$: minimize the project tardiness (i.e. the maximum of the tardiness of the subprojects or activities),
$\gamma =$ *early/tardy*: minimize the weighted earliness-tardiness of the project,
$\gamma = n_T$: minimize the number of tardy activities.

levelling:

$\gamma = \Sigma sq.dev.$: minimize the sum of the squared deviations of the resource requirements from the average,
$\gamma = \Sigma jump$: minimize the weighted jumps in resource usage for all resource types over all time periods,
$\gamma = \Sigma abs.dev.$: minimize the sum of the absolute deviations of the resource requirements from the average,
$\gamma = av$: minimize the resource availability in order to meet the project deadline,
$\gamma = rac$: minimize the resource availability costs, i.e. the weighted availability of each resource type,
$\gamma = curve$: determine the complete time/cost trade-off curve.

financial:

$\gamma = npv$: maximize the net present value of the project.

stochastic:

$\gamma =$ E[.]: optimize the expected value of a performance measure,
$\gamma = cdf$: determine the cumulative density function of the project realization

date,

$\gamma = ci$: determine the criticality index of an activity (the probability that it will be on the critical path) or of a path (the probability that it will be critical),

$\gamma = mci$: determine the most critical path(s) or activities based on the criticality index values.

The above examples refer to the use of *single objective functions*. Obviously *multiple objectives* may be used (see for example Weglarz 1990, Nabrzyski and Weglarz 1994, Slowinski et al. 1994). We suggest the setting $\gamma = multi$ to specify the multi-objective case where different objectives are weighted or combined and the setting $\gamma = multicrit$ to specify multicriteria functions. Again we leave it up to the user to be more specific in the specification of the multiple objectives used.

In the above examples no distinction is made between single and multi-project scheduling. *Multi-project scheduling* problem settings can easily be accommodated. The network logic allows for the combination of the activity networks of the various projects into a single network and the user is given the freedom to specify the proper performance measures used. In practice *hybrid multi-project programs* occur which are made up of several classes of projects, each possibly having its own distinctive characteristics. In such cases, we suggest to identify the overall multi-project scheduling problem by the parameter settings corresponding to the most general case. The various reductions among the resource and activity parameters discussed earlier may prove to be very useful in this respect. The eventual use of multiple objectives can be specified along the lines indicated above.

1.3. Use of the classification scheme

The purpose of this section is to illustrate the potential use of the classification scheme in the characterization and classification of the project scheduling problems encountered in the literature.

1.3.1. Scheduling in the absence of resource constraints

1.3.1.1. Time analysis of activity networks. The classical problem of computing the longest path (critical path length) in a PERT/CPM project with finish-start precedence relations with zero time lag and known activity durations is represented as $cpm|C_{max}$. The same problem in precedence diagramming networks, i.e. networks with finish-start, start-finish, start-start and finish-finish relations with minimal time lags only is denoted as $min|C_{max}$. The problem of computing the critical path in a project with generalized precedence relations with minimal and maximal time lags is denoted as $gpr|C_{max}$.

1.3.1.2. Maximizing the net present value in project networks. The problem of scheduling project activities subject to finish-start precedence constraints with zero time lag in order to maximize the net present value of the project (often referred to as the unconstrained max-npv problem; for a review see Herroelen et al. 1997) is represented as $cpm,c_j|npv$. This notation implies that the cash flows are associated with the network activities and that the cash flow amounts are assumed to be given. The same problem in networks with minimal time lags is denoted as $min,c_j|npv$. For networks with generalized precedence relations the problem is denoted as $gpr,c_j|npv$. The so-called payment scheduling problem (Dayanand and Padman 1997) which involves the simultaneous determination of both the amount and timing of progress payments in an unconstrained max-npv environment can be denoted as $cpm,sched|npv$.

1.3.2. Project scheduling under resource constraints

The literature on project scheduling under various types of resource constraints has expanded drastically over the past few years. For a review of the recent developments we refer the reader to Herroelen et al. (1998).

1.3.2.1. The resource-constrained project scheduling problem. The resource-constrained project scheduling problem (often denoted as RCPSP) involves the scheduling of project activities subject to finish-start precedence constraints with zero time lag and constant renewable resource constraints in order to minimize the project duration. Activities have a single execution mode with a fixed integer duration, preemption is not allowed and renewable resource requirements are constant throughout the duration of an activity. This problem is denoted as $m,1|cpm|C_{max}$. The problem is known to be a generalization of the job shop scheduling problem.

1.3.2.2. The preemptive resource-constrained project scheduling problem. The preemptive resource-constrained project scheduling problem (often denoted as PRCPSP) extends the $m,1|cpm|C_{max}$ in that it allows for activity preemption at integer points in time. The problem can be denoted as $m,1|pmtn,cpm|C_{max}$.

1.3.2.3. The generalized resource-constrained project scheduling problem. The generalized resource-constrained project scheduling problem (often referred to as GRCPSP) extends problem $m,1|cpm|C_{max}$ to the case of minimal time lags, activity release dates and deadlines and variable resource availabilities. The proposed classification scheme classifies the problem as $m,1,va|min,\rho_j,\delta_j|C_{max}$.

1.3.2.4. The resource-constrained project scheduling problem with generalized precedence relations. The resource-constrained project scheduling problem with generalized precedence relations (often denoted as RCPSP-GPR or RCPSP/max) extends problem $m,1|cpm|C_{max}$ in that it allows for start-start, finish-start, start-finish and finish-finish precedence constraints with both minimal and maximal time lags. This basic extension can be denoted as $m,1|gpr|C_{max}$. The use of minimal and maximal time lags also allows, however, for the modelling of ready times and activity deadlines, variable resource requirements and availabilities. Procedures have been developed which allow for the use of any regular objective function. This general problem setting is denoted as $m,1,va|gpr,\rho_j,\delta_j,vr|reg$.

1.3.2.5. Time/cost trade-off problems. The classical time/cost trade-off problem in CPM-networks (described in project management textbooks such as Elmaghraby 1977, Moder et al. 1983, Wiest and Levy 1977, Shtub et al. 1994) basically assumes that resources are available in infinite amounts and hence does not *explicitly* take resource decisions into account. A direct activity cost function is used instead, representing the direct activity costs as a function of the activity duration. Activity durations are bounded from below by the crash duration (corresponding to a maximum allocation of resources) and bounded from above by the normal duration (corresponding to the most efficient resource allocation). Given a project deadline, the objective is to determine the activity durations and to schedule the activities in order to minimize the project cost; i.e. the sum of the direct costs of the activities. Essentially, the project costs correspond to a requirement for a nonrenewable resource, the total requirement of which is to be minimized. This corresponds to minimizing the (required) availability of the nonrenewable resource. Hence, the problem can be denoted as $1,T|cpm,\delta_n,lin,mu|av$ when the activity cost functions are linear. The notation $1,T|cpm,\delta_n,conc,mu|av$ is used for the case of concave activity cost functions while the notation $1,T|cpm,\delta_n,conv,mu|av$ is used in case the activity cost functions are convex. The notation $1,T|cpm,\delta_n,cont,mu|av$ denotes the time/cost trade-off problem for the case of general continuous activity cost functions.

The discrete time/cost trade-off problem (for a review see De et al. 1995) assumes a single nonrenewable resource. The duration of an activity is a discrete, nonincreasing function of the amount of a single resource allocated to it. An activity assumes different execution modes according to the possible resource allocations. When a limit on the total availability of the single nonrenewable resource is specified, the problem is to decide on the vector of activity durations that completes the project as early as possible under the limited availability of the single nonrenewable resource type. This problem is denoted by our classification scheme as $1,T|cpm,disc,mu|C_{max}$. Obviously, the notation becomes $1,T|min,disc,mu|C_{max}$ and $1,T|gpr,disc,mu|C_{max}$ for the case of minimal and both minimal and maximal time lags, respectively. A second objective function reverses the problem formulation. Now a project deadline is specified and the minimization is over the sum of the resource use over all activities. The notation for this problem is $1,T|cpm,\delta_n,disc,mu|av$, where again the parameter *cpm* can be changed into *min* and *gpr* to denote the corresponding type of precedence relations. It should be noted that the only difference with the classical time/cost trade-off problem in CPM-networks lies in the use of the *disc* parameter specification, referring to the use of a discrete resource requirements function. In some studies room is made for a third objective which involves the computation of the complete time/cost trade-off function for the total project. Exploiting the degrees of freedom allowed by our classification scheme, this problem could be represented as $1,T|cpm,disc,mu|curve$, where the value *curve* is given to the parameter γ.

1.3.2.6. Discrete time/resource trade-off problems. The discrete time/resource trade-off problem (often referred to as DTRTP) assumes that the duration of an activity is a discrete, non-increasing function of the amount of a single *renewable* resource committed to it. Given a specified work content for an activity, all its efficient execution modes are determined based on time/resource trade-offs. An activity when performed in a specific mode has a duration and a resource requirement during each period it is in progress, such that the resource-duration product is at least equal to the specified work content. The single resource has a constant per period availability. The objective is to schedule each activity in one of its modes, subject to the precedence and the renewable resource constraints, under the objective of minimizing the project makespan.

The discrete time/resource trade-off problem can be denoted as $1,1|cpm,disc,mu|C_{max}$. Obviously, *min* or *gpr* can be used instead of *cpm* to denote the proper type of precedence relations. Moreover, the problem resembles the discrete time/cost trade-off problem $1,T|cpm,disc,mu|C_{max}$ which studies time/cost trade-offs for a single nonrenewable resource. The only difference in the notation lies in the second parameter of the α-field: a T is used for the time/cost trade-off problems while a 1 is used for the time/resource trade-off problems.

It is also possible to define a kind of dual problem to the discrete time/resource trade-off problem which involves the minimization of the resource availability subject to the project deadline. The corresponding notation is $1,1|cpm,\delta_n,disc,mu|av$.

1.3.2.7. Multi-mode resource-constrained project scheduling problems. The multi-mode resource-constrained project scheduling problem (sometimes referred to as MRCPSP) includes time/resource and resource/resource trade-offs, multiple renewable, nonrenewable and doubly-constrained resources and a variety of objective functions. In the basic problem setting activities have to be scheduled in one of their possible execution modes subject to renewable and nonrenewable resources. Under the minimum makespan objective the general problem, including renewable and nonrenewable resources, can be denoted as $m,1T|cpm,disc,mu|C_{max}$ for projects with finish-start precedence constraints with zero time lag. It should be noted that algorithms have been developed to deal with multi-mode problems involving deadlines, variable availabilities and any type of regular objective function. The problem tackled by such procedures would be denoted as $m,1T,va|cpm,\delta_j,disc,mu|reg$. The problem with *mode identity constraints* can be denoted as $m,1T|cpm,disc,id|C_{max}$. The multi-mode problem with partially renewable resource constraints can be denoted as $m,v|cpm,disc,mu|C_{max}$.

1.3.2.8. Resource levelling problems. Various types of resource levelling problems have been studied in the project scheduling literature. The classical *resource levelling problem* (denoted by some authors as *RLP*) is somewhat the dual of the resource-constrained project scheduling problem. Instead of minimizing the project duration subject to renewable resource constraints, the problem now is to schedule the activities in order to level the resource use subject to a project deadline. An explicit resource availability constraint is not taken into account. Various levelling objectives have been used in the literature. In essence, they can be considered as resource cost functions. The resource levelling problem can be denoted according to the rules of our classification scheme as $m,1|cpm,\delta_n|\Sigma sq.dev.$, if the objective is to minimize the squared deviations of the resource requirements around the average resource requirement, which is equivalent to minimizing the sum of squares of the resource requirements for each period in the project schedule as such. If the objective is to minimize the absolute deviations of the resource requirements, the notation becomes $m,1|cpm,\delta_n|\Sigma abs.dev.$ The γ-field in the notation becomes $\Sigma jump$ for the objective of minimizing the weighted jumps in resource consumption for each resource type over all time periods.

Instead of taking the view of finding the minimum project length which does not violate the precedence and resource constraints, the *resource availability cost problem* (*RACP*) takes the view that the individual resource availabilities determine the cost of executing the schedule. It aims at determining the cheapest resource availability amounts for which a feasible project schedule exists that does not violate the project deadline. Resource costs are to be determined under the assumption of a discrete, non-decreasing cost function of the constant availability of the renewable resource types. The notation for this problem is $m,1|cpm,\delta_n|rac$.

1.3.2.9. Resource-constrained project scheduling with discounted cash flows. The basic problem of maximizing the net present value of a project subject to finish-start precedence and renewable resource constraints is denoted as $m,1|cpm,c_j|npv$. This problem setting assumes that the cash flow amounts, c_j, are known and associated with the network activities. If the known cash flows occur at regular intervals or with a known frequency, the corresponding problem can be denoted as $m,1|cpm,per|npv$. In the case where both the amount and timing of the cash flows have to be determined the so-called payment scheduling problem results, which can be denoted as $m,1|cpm,sched|npv$. Obviously, the parameter setting *min* can be used in the β-field to denote the case with minimal time lags and the setting *gpr* can be used for the case of both minimal and maximal time lags. In a similar fashion ready times and deadlines can be introduced in the notation through the parameter settings ρ_j and δ_j.

1.3.3. Classification of stochastic problem settings

The above classifications mainly apply to a deterministic problem setting. The classification of the stochastic problem equivalents should pose no major problems as the degrees of freedom present in our classification scheme allow for proper classification in those cases. Setting the activity duration parameter $\beta_4 = \tilde{d}_j$ to denote that the activity durations are stochastic, the problem involving the determination (estimation) of the cumulative probability density function of the project realization date in projects with stochastic activity durations can be represented as *cpm*, $\tilde{d}_j$ |*cdf*. The problem of computing the expected duration of a GERT network can be denoted as *gert*, $\tilde{d}_j$ |E[C_{max}].

In most cases, the stochastic characteristics apply to the activity durations and the amount of the cash flows. Setting the cash flow parameter $\beta_8 = \tilde{c}_j$ in order to indicate that the cash flows are stochastic, and using $E[npv]$ to indicate the objective of maximizing the *expected* net present value of the project, the stochastic resource-constrained project scheduling problem with discounted cash flows subject to finish-start precedence constraints and renewable resource constraints is denoted as $m, 1|cpm, \tilde{d}_j, \tilde{c}_j|E[npv]$.

It is also possible to specify stochastic resource availabilities and requirements. Setting $\alpha_3 = \tilde{a}$ denotes a stochastic resource availability which remains constant over time. The setting $\alpha_3 = v\tilde{a}$ denotes a stochastic resource availability which varies over time. In a similar fashion the setting $\beta_6 = \tilde{r}$ denotes a stochastic discrete resource requirement which is the same for every time period of activity execution. The setting $\beta_6 = v\tilde{r}$ denotes a stochastic resource requirement of a discrete number of units which varies over the periods of activity execution.

1.4. Conclusions

Over the past few years the variety of project scheduling problems studied in the literature has been drastically increased. The conception of new problem types and the research of existing basic models under more realistic problem assumptions has created the need for a detailed classification scheme which allows for a precise and unambiguous classification of the problems under study.

In this chapter a classification scheme is introduced which is composed of three fields $\alpha|\beta|\gamma$, denoting the resource characteristics, activity characteristics and performance measures, respectively. Precise settings for the various parameters are introduced together with the corresponding reduction graphs which allow the description of the interrelations between the various problem settings. The scheme is illustrated by applying it to the most common project scheduling problems which have been studied in the literature.

The classification scheme proves to possess sufficient rigidity *and* flexibility to accommodate in an unambiguous way the full spectrum of problem type characteristics. It offers sufficient detail to allow for a *unique* codification of the relevant project scheduling problems studied by the project scheduling community. Obviously, this imposes a rather high degree of rigour on the various parameter specifications. At the same time, however, we tried to keep the scheme *flexible* and *workable*, i.e. as concise and simple as possible, with sufficient degrees of freedom to be used by the individual user in order to specify individual "desires".

We are convinced that the scheme is usable. It allows for the unique codification of the overwhelming variety of project scheduling problems under study, as illustrated by the sample we provide in Section 1.3. Obviously, the scheme takes some time to digest. It must be studied before it can be used. Our experience is that once one tries it out on a few problem settings, the smoke curtain which initially may hide the logic behind the three field parameter settings steadily disappears.

We do hope that the scheme gains wide acceptance within the project scheduling community and we hope that (a) it facilitates the presentation and discussion of project scheduling problems, (b) it relieves authors and presenters of "wasting" a lot of time in preparing lengthy descriptions of problem assumptions and characteristics, (c) it allows for immediate problem identification and simplifies the assessment of problem complexity, (d) it allows for the identification of viable, relevant areas of research, (e) it provides additional insight in the ideosyncracies of the various problem settings studied in the literature by showing problem interrelations, dependencies, common characteristics and complicating factors.

Obviously there may always be a project setting that does not fit exactly in the "pigeon holes" we provide. We have done our best to combine "rigour" and "freedom" and have tried to provide sufficient generality in the categories, which offers the user sufficient degrees of specification freedom.

Acknowledgement

The development of the classification scheme presented in this chapter was inspired by discussions held at the Workshop on Scheduling & Heuristic Search, held on May 8, 1997 at the A. Gary Anderson Graduate School of Management, University of California, Riverside. We are grateful to the constructive comments and criticism received from many colleagues within the project scheduling community. Especially the detailed and constructive remarks of Richard Deckro (Air Force Institute of Technology), Bajis Dodin (University of California at Riverside), Salah Elmaghraby (North Carolina State University at Raleigh), Concepcion Maroto (Universidad Politécnica de Valencia), Jim Patterson (Indiana University) and Jan Weglarz (Poznan University of Technology) are deeply appreciated. Their suggestions to remove ambiguity and modify various parameter settings have been crucial in making the scheme to what it is now.

References

BARTSCH, T., R. NISSEN AND F. SALEWSKI. 1997. Genetic Algorithms for Resource-Constrained Project Scheduling with Changeover Times. Research Report, Christian-Albrechts-Universität zu Kiel.

BLAZEWICZ, J., W. CELLARY, R. SLOWINSKI AND J. WEGLARZ. 1986. *Scheduling under Resource Constraints - Deterministic Models*. Baltzer, Basel.

BLAZEWICZ, J., K. H. ECKER, G. SCHMIDT AND J. WEGLARZ. 1994. *Scheduling in Computer and ManufacturingSystems*. Springer-Verlag, Berlin.

BLAZEWICZ, J., J. K. LENSTRA AND A. H. G. RINNOOY KAN. 1983. Scheduling Subject to Resource Constraints: Classification and Complexity. *Discrete Appl. Math.* **5**, 11-24.

BÖTTCHER, J., A. DREXL AND R. KOLISCH. 1996. A Branch-and-Bound Procedure for Project Scheduling with Partially Renewable Resource Constraints. *Proceedings of the Fifth Workshop on Project Management and Scheduling*, Poznan, April 11-13, 48-51.

BRUCKER, P. 1995. *Scheduling Algorithms*. Springer-Verlag, Berlin.

CONWAY, R. W., W. L. MAXWELL AND L. W. MILLER. 1967. *Theory of Scheduling*. Addison-Wesley, Reading, Mass.

DAYANAND, N. AND R. PADMAN. 1997. On Modelling Payments in Projects. *J. Opnl. Res. Soc.*, to appear.

DE, P., E. J. DUNNE, J. B. GOSH AND C. E. WELLS. 1995. The Discrete Time/Cost Trade-Off Problem Revisited. *Eur. J. Opnl. Res.* **81**, 225-238.

DE REYCK, B. 1995. Project Scheduling under Generalized Precedence Relations - A Review: Part 1 and Part 2. Research Reports 9517-9518, Department of Applied Economics, Katholieke Universiteit Leuven.

DODIN, B. AND A. A. ELIMAM. 1997. Audit Scheduling with Overlapping Activities and Sequence-Dependent Setup Costs. *Eur. J. Opnl. Res.* **97**, 22-33.

DREXL, A. 1997. Local Search Methods for Project Scheduling under Partially Renewable Resource Constraints. Paper presented at the INFORMS San Diego Spring Meeting, May 4-7, 1997.

ELMAGHRABY, S. E. 1977. *Activity Networks - Project Planning and Control by Network Models*. Wiley Interscience, New York.

ELMAGHRABY, S. E. 1995. Activity Nets: A Guided Tour through Some Recent Developments. *Eur. J. Opnl. Res.* **82**, 383-408.

GRAHAM, R. L., E. L. LAWLER, J. K. LENSTRA AND A. H. G. RINNOOY KAN. 1979. Optimization and Approximation in Deterministic Sequencing and Scheduling Theory: A Survey. *Annals Discrete Math.* **5**, 287-326.

HACKMAN, S. T. AND R. C. LEACHMAN. 1989. An Aggregate Model of Project-Oriented Production. *IEEE Trans. Systems, Man and Cybernetics.* **19**, 220-231.

HERROELEN, W. AND E. DEMEULEMEESTER. 1995. Recent Advances in Branch-and-Bound Procedures for Resource-Constrained Project Scheduling Problems, Chapter 12 in *Scheduling Theory and Its Applications* (Chrétienne, Ph. et al. (Eds)), John Wiley & Sons, Chichester.

HERROELEN, W., E. DEMEULEMEESTER AND B. DE REYCK. 1998. Resource-Constrained Project Scheduling - A Survey of Recent Developments. *Comput. and O.R.* **25**, 279-302.

HERROELEN, W. S., P. VAN DOMMELEN AND E. L. DEMEULEMEESTER. 1997. Project Network Models with Discounted Cash Flows: A Guided Tour through Recent Developments. *Eur. J. Opnl. Res.* **100**, 97-121.

ICMELI, O., S. ERENGÜÇ AND J. C. ZAPPE. 1993. Project Scheduling Problems: A Survey. *Int. J. Prod. Opns. Mgmt.* **13**, 80-91.

JOZEFOWSKA, J. AND J. WEGLARZ. 1994. Approximation Algorithms for Some Discrete-Continuous Scheduling Problems. *Proceedings of the Fourth International Workshop on Project Management and Scheduling*, Leuven, July 12-15, 62-63.

KAPLAN, L. 1991. Resource-Constrained Project Scheduling with Setup Times. Research Report, University of Tenessee.

KOLISCH, R. 1995. *Project Scheduling under Resource Constraints*. Physica-Verlag, Heidelberg.

LAWLER, E. L., J. K. LENSTRA, A. H. G. RINNOOY KAN AND D. B. SHMOYS. 1993. Sequencing and Scheduling: Algorithms and Complexity, Chapter 9 in *Logistics of Production and Inventory*, Handbooks in Operations Research and Management Science, Volume 4 (S.C. Graves et al. (Eds.)), North-Holland, Amsterdam, 445-522.

LEACHMAN, R. C. 1983. Multiple Resource Leveling in Construction Systems through Variation in Activity Intensities. *Nav. Res. Log. Quart.* **30**, 187-198.

LEACHMAN, R. C., A. DINCERLER AND S. KIM. 1990. Resource-Constrained Scheduling of Projects with Variable-Intensity Activities. *IIE Trans.* **22**, 31-40.

MODER, J. J., C. R. PHILLIPS AND E. W. DAVIS. 1983. *Project Management with CPM, PERT and Precedence Diagramming*. Van Nostrand Reinhold Company, Third Edition.

NABRZYSKI, J. AND J. WEGLARZ. 1994. A Knowledge-Based Multiobjective Project Scheduling System. *Revue des systèmes de décision*, **3**, 185-200.

NEUMANN, K. AND U. STEINHARDT. 1979. *GERT Networks and the Time-Oriented Evaluation of Projects*. Lecture Notes in Economics and Mathematical Systems, 172, Springer-Verlag, Berlin.

ÖZDAMAR, L. AND G. ULUSOY. 1995. A Survey on the Resource-Constrained Project Scheduling Problem. *IIE Transactions*, 27, 574-586.

PINEDO, M. 1995. *Scheduling - Theory, Algorithms and Systems*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

SALEWSKI, F. 1996. Tabu Search Algorithms for Project Scheduling under Resource and Mode Identity Constraints. Paper presented at the IFORS 14th Triennial Meeting, Vancouver, B.C., July 8-12.

SALEWSKI, F. AND S. LIEBERAM-SCHMIDT. 1996. Greedy Look Ahead Methods for Project Scheduling under Resource and Mode Identity Constraints, *Proceedings of the Fifth International Workshop on Project Management and Scheduling*, Poznan, April 11-13, 207-211.

SALEWSKI, F., A. SCHIRMER AND A. DREXL. 1997. Project Scheduling under Resource and Mode Identity Constraints - Model, Complexity, Methods and Application. *Eur. J. Opnl. Res.* **102**, 88-110.

SCHIRMER, A. AND A. DREXL. 1996. Partially Renewable Resources - A Generalization of Resource-Constrained Project Scheduling. Paper presented at the IFORS Triennial Meeting, Vancouver, B.C., July 8-12.

SHTUB, A., J. F. BARD AND S. GLOBERSON. 1994. *Project Management: Engineering, Technology, And Implementation*. Prentice Hall International, Inc., Englewood Cliffs.

SLOWINSKI, R., B. SONIEWICKI AND J. WEGLARZ. 1994. DSS for Multiobjective Project Scheduling. *Eur. J. Opnl. Res.* **79**, 220-229.

WEGLARZ, J. 1980. Control in Resource Allocation Systems. *Foundations of Control Engineering*, **5**, 159-180.

WEGLARZ, J. 1990. Synthesis Problems in Allocating Continuous, Doubly Constrained Resources among Dynamic Activities. *Proceedings of the IFORS 12th Triennial Meeting*, 715-724.

WEGLARZ, J. AND J. JOZEFOWSKA. 1997. Parallel Machine Scheduling with Additional Continuous Resource: Mean Completion Time Results. Paper presented at the INFORMS San Diego Spring Meeting, May 4-7.

WIEST, J. D. AND F. K. LEVY. 1977. *A Management Guide to PERT/CPM: with GERT/PDM/DCPM and Other Networks*. Prentice-Hall, Inc., Englewood Cliffs.