To appear in the *European Journal of Operational Research*

A Survey of Variants and Extensions of the Resource-Constrained Project Scheduling Problem

Sönke Hartmann[a,*], Dirk Briskorn[b,c,†]

[a] HSBA Hamburg School of Business Administration
Alter Wall 38, D-20457 Hamburg, Germany
e-mail: soenke.hartmann@hsba.de
www.hsba.de

[b] Lehrstuhl für Produktion und Logistik
Christian-Albrechts-Universität zu Kiel, D-24098 Kiel, Germany
e-mail: briskorn@bwl.uni-kiel.de
www.bwl.uni-kiel.de/Prod/team/briskorn/

[c] Leonard N. Stern School of Business
New York University, 44 West 4th Street, New York, NY 10012, USA

* supported by the HSBA Foundation

† supported by a fellowship within the Postdoc-Program of the German Academic Exchange Service (DAAD)

Abstract

The resource-constrained project scheduling problem (RCPSP) consists of activities that must be scheduled subject to precedence and resource constraints such that the makespan is minimized. It has become a well-known standard problem in the context of project scheduling which has attracted numerous researchers who developed both exact and heuristic scheduling procedures. However, it is a rather basic model with assumptions that are too restrictive for many practical applications. Consequently, various extensions of the basic RCPSP have been developed. This paper gives an overview over these extensions. The extensions are classified according to the structure of the RCPSP. We summarize generalizations of the activity concept, of the precedence relations and of the resource constraints. Alternative objectives and approaches for scheduling multiple projects are discussed as well. In addition to popular variants and extensions such as multiple modes, minimal and maximal time lags, and net present value-based objectives, the paper also provides a survey of many less known concepts.

Keywords: Project scheduling, modeling, resource constraints, temporal constraints, networks.

1 Introduction

Project scheduling is an important task in project management. The presence of scarce resources as well as precedence relations between activities makes project scheduling a difficult task. In practice, specific software is used to support the scheduling process. The foundation of such software is a formal model that allows to describe the actual project by a set of scheduling constraints and an objective function.

During the last decades, the resource-constrained project scheduling problem (RCPSP) has become a standard problem for project scheduling in the literature. It can be summarized as follows. The RCPSP considers a project with J activities which are labeled $j = 1, \ldots, J$. The processing time (or duration) of an activity j is denoted as p_j. Once started, an activity may not be interrupted, i.e., preemption is not allowed. Due to technological requirements, there are precedence relations between some of the activities. They are given by sets of immediate predecessors P_j indicating that an activity j may not be started before each of its predecessors $i \in P_j$ is completed. The precedence relations can be represented by an activity-on-node network which is assumed to be acyclic. Each activity requires certain amounts of resources to be performed. The resources are called renewable because their full capacity is available in every period. We have K renewable resources labeled $k = 1, \ldots, K$. For each resource k the per-period availability is assumed to be constant over time, it is given by R_k. Activity j requires r_{jk} units of resource k in each period it is in process. We consider two additional activities $j = 0$ and $j = J + 1$ representing the start and the completion of the project, respectively. Both are "dummy" activities with durations of 0 and no resource requests.

All information is assumed to be deterministic and known in advance. The parameters are assumed to be nonnegative and integer valued. A schedule is an assignment of start times S_j to the activities $j = 0, 1, \ldots, J + 1$. The objective is to find a schedule which leads to the earliest possible end of the project, i.e., the minimal makespan. Blazewicz et al. [21] have shown that the RCPSP belongs to the class of the strongly NP-hard problems. A mathematical model for the RCPSP has been developed by Pritsker et al. [150]. Brucker et al. [31] have provided a notation to classify resource-constrained project scheduling problems. This notation follows the famous three field notation $\alpha|\beta|\gamma$ for machine scheduling problems introduced in Graham et al. [75]. In the context of project scheduling α specifies characteristics of the resources, β describes activities (and can be assigned more than one value), and γ denotes the problem's objective. The standard RCPSP is denoted by $PS|prec|C_{\max}$.

While the RCPSP as given above is already a powerful model, it cannot cover all situations that occur in practice. Therefore, many researchers have developed more general project scheduling problems, often using the standard RCPSP as a starting point. Since the 1990s, several survey papers on project scheduling have been published. Those papers focus on methods for the RCPSP (see Hartmann and Kolisch [81], Kolisch and Hartmann [108, 109]) and a few common variants (see Brucker [29], Brucker et al. [31], Herroelen et al. [88], Herroelen [85], Kolisch and Padman [111], Özdamar and Ulusoy [146] and Tavares [172]).

The purpose of this paper is to provide a broad overview over variants and extensions of the RCPSP that have been proposed in the literature. We restrict ourselves to deterministic approaches (for an introduction to stochastic problems, the reader is referred to Neumann [133] and Herroelen and Leus [86]). Since a huge number of papers are available, we cannot give a fully exhaustive reference list, but we work out the diversity of problem settings currently available for project scheduling. If several papers are available for a concept, we focus on references from the last ten years. For those concepts presented here that are covered by the classification scheme of Brucker et al. [31] we point out to the related notation. Moreover, we propose extensions for some common concepts that are not yet covered by the existing classification scheme.

The outline is as follows: Section 2 lists generalizations of the activity concept. Alternative precedence constraints and network characteristics are summarized in Section 3. Section 4 gives extensions of the resource concept. Different objectives are outlined in Section 5. Finally, Section 6 deals with the simultaneous consideration of multiple projects, and Section 7 briefly summarizes generators that have been developed to produce test instances for various project scheduling probems.

2 Generalized Activity Concepts

2.1 Preemptive Scheduling

The basic RCPSP assumes that an activity cannot be interrupted once it has been started. Bianco et al. [20], Brucker and Knust [30], Debels and Vanhoucke [48], Demeulemeester and Herroelen [51] and Nudtasomboon and Randhawa [144] allow activity preemption at discrete points in time, that is, an activity can be interrupted after each integer unit of its processing time. In the $\alpha|\beta|\gamma$ notation of Brucker et al. [31], preemption is represented by adding the value *prmt* to β.

Ballestin et al. [11] consider a variant in which an activity may be interrupted at most m times; their focus is on the case $m = 1$.

Debels and Vanhoucke [48] extend the concept of preemption by a so-called fast tracking option: The parts of an activity that result from preemption need not to be processed in sequence, i.e. they can also be carried out in parallel.

Franck et al. [73] propose a calendar concept for project scheduling which includes preemptive scheduling. A calendar is defined as a binary function that determines for each period whether activity execution is possible or a break occurs during which an activity may not be started or continued. (Note that there is an analogy between the calendar concept and resource capacities varying with time, cf. Section 4.6.) Activities are allowed to be interrupted but only due to calendar breaks (a similar concept is considered in Schwindt and Trautmann [162]). Moreover, each activity j has a minimum processing time ε_j during which it may not be interrupted.

Buddhakulsomsiria and Kim [32, 33] follow a similar approach. Their approach permits activity splitting due to resource vacations (i.e., resource capacities varying with time). This concept is embedded into a multi-mode RCPSP with renewable resources only (cf. Section 2.4).

Damay et al. [43] consider two types of activities. An activity of the first type may not be interrupted. An activity of the second type may be interrupted at arbitrary (not necessarily integer) points in time.

2.2 Resource Requests Varying with Time

The activities in the standard RCPSP require constant amounts of renewable resources, that is, the per-period request for a resource does not change during the duration of an activity. This can be generalized by resource requests varying with time. To incorporate this formally, we denote with r_{jkt} the request of activity j for renewable resource k in the t-th period of its processing time.

Hartmann [80] describes a real-world medical research project with time-dependent resource requests. There, each activity requires certain laboratory equipment, but only in the last period of its duration.

Cavalcante et al. [35] follow a similar problem setting. They define activities with time-dependent resource requests for one renewable resource which is interpreted as labor.

Drezet and Billaut [65] deal with time-dependent requests for resources which represent software developers. A minimum and a maximum resource request per period is given. Additional constraints reflect legal requirements such as the maximal working time per day.

Bartusch et al. [16] point out that resource requests varying with time can be transformed into constant requests if maximal time lags (see Section 3.2) are available. An activity is split into parts with constant requests. Between two sub-activities i and $i+1$ a precedence relation is introduced as well as a maximal time lag to avoid preemption of the original activity.

2.3 Setup Times

In some cases, it may be required that a resource (e.g., a machine) must be prepared before a certain activity can be started. The time needed for such a preparation is called setup time.

Mika et al. [127] consider three types of setup times. Sequence-independent setup times depend only on the activity and the resource on which the activity will be performed. Sequence-dependent setup times additionally depend on the sequence of the activities, i.e., on the previous activity executed on the same resource. Finally, schedule-dependent setup times depend on the occupation of resources by activities over time, that is, the setup time related to the processing of activity j depends on the resources the direct predecessors of j are processed on. These concepts are embedded into an RCPSP with multiple modes (see Section 2.4). The authors note that schedule-dependent setup times could be captured by adding activities with specific modes, although this would lead to a huge number of additional activities and modes. In the notation of Brucker et al. [31], s_j and s_{ij} as values of β can be used to represent sequence-independent and sequence-dependent setup times, respectively.

Mika et al. [126] discuss many further aspects of setup times. They consider setup times between single activities as well as between families of activities. Moreover, they distinguish between inseparable setups which must be executed immediately before an activity starts and separable activities which may also start earlier. A setup may be executed either after the predecessors are finished or independently from the predecessors, or it may overlap with the predecessor for a given time. The setups of several resources may be required to be synchronous or not. Finally, so-called removal times are analogous to setup times but occur after an activity is finished.

Vanhoucke [178] incorporates sequence-independent setup times into an RCPSP with preemption and fast tracking (see Section 2.1). Each time an activity is resumed after having been interrupted, the setup time occurs.

Schwindt and Trautmann [162] incorporate sequence-dependent setup times into an RCPSP with various extensions. Their goal is to capture a scheduling problem that arises in batch production.

Drexl et al. [64] employ sequence-dependent setup times (which are called changeover times here) into a multi-mode RCPSP (see Section 2.4). The setup time of activity j performed in mode m_j depends on the previous activity i and its mode m_i.

Nonobe and Ibaraki [141] model setup times as additional activities that have to be completed before the activity which requires the setup is started.

2.4 Multiple Modes

The standard RCPSP assumes that an activity can only be executed in a single way which is determined by a fixed duration and fixed resource requirements. Starting with the work of Elmaghraby [66], the activity concept as given in the standard RCPSP has been extended by allowing several alternatives or modes in which an activity can be performed. Each mode reflects a feasible way to combine a duration and resource requests that allow to accomplish the underlying activity.

The assumptions of the multi-mode RCPSP can be summarized as follows. An activity j must be performed in one of its modes which are labelled $1,\dots,M_j$ with M_j being the number of modes. Once started in one of its modes, an activity must be completed in that mode; mode changes and preemption are not permitted. The processing time of activity j being executed in mode m is given by p_{jm}. The request of activity j being executed in mode m for resource k is r_{jmk}. In addition to renewable resources, often also nonrenewable resources are considered in multi-mode models, see Section 4.1. A schedule for the multi-mode RCPSP assigns a start time S_j and a mode m_j to each activity j. The most popular objective in the literature is the minimization of the makespan.

The multi-mode RCPSP is often referred to as MRCPSP. Clearly, if only one mode per activity and no nonrenewable resources are given, we obtain the standard RCPSP. Note that there might not be a feasible schedule if a nonrenewable resources is given, and as shown by Kolisch and Drexl [107] the related feasibility problem (i.e., the problem to determine whether there is a feasible schedule) is NP-complete if at least two nonrenewable resources are present. To capture multiple modes for activities in the notation of Brucker et al. [31], α is set to *MPS*.

Recent papers on this problem class include Alcaraz et al. [6], Bouleimen and Lecocq [25], Hartmann [79], Jarboui et al. [94], Józefowska et al. [97], Özdamar [145] and Pesch [149] while Varma et al. [187] discuss a multi-mode problems without nonrenewable resources. Multi-mode problems with generalized precedence constraints have been considered by Barrios et al. [14], Brucker and Knust [30], Calhoun et al. [34], de Reyck and Herroelen [47], Drexl et al. [64], Heilmann [82, 83], Nonobe and Ibaraki [141], and Sabzehparvar and Seyed-Hosseini [156] (see also Section 3). Zhu et al. [198] employ a multi-mode problem with generalized resource constraints (see also Section 4).

Erenguc et al. [68] introduce so-called crashable modes. The duration of a mode can be shortened at the expense of additional costs, which is essentially a special case of the mode concept described above.

Bellenguez and Néron [18] consider activities which require staff members with certain skill levels. This can be seen as a special case of the mode concept where each mode corresponds to a feasible subset of staff members that can carry out the activity with regard to the required skill levels.

Li and Womer [121] employ the mode concept to take quality considerations into account. Each mode m of an activity j is associated with a quality γ_{jm}. The sum of the qualities related to the actual modes in the schedule must be at least Γ. The authors use the resulting multi-mode RCPSP to capture the configuration of a supply chain, where the quality represents the reliability of a node in the supply chain.

Tareghian and Taheri [171] follow a similar approach. Their objective is to maximize the quality while the project's deadline and budget (nonrenewable resource capacity) must be observed.

Tiwari et al. [173] also consider quality. In their problem setting, an activity may be started in a mode that does not allow to complete the activity at a required quality level. Processing in such a mode then has to be followed by a rework mode that completes the activity. Their approach was motivated by customer training projects in a telecommunication firm. There, an activity may be started by a less skilled employee who is available and then completed by an employee with the required skills.

Nudtasomboon and Randhawa [144] allow preemption of activities in a multi-mode RCPSP. When an activity is resumed, the mode may not change. Various alternative objectives are included (see Section 5).

Salewski et al. [158] and Drexl et al. [64] extend the multi-mode RCPSP by introducing so-called mode identity constraints. The motivation for this is that there may be several activities that should be performed in the same way, e.g., by allocating the same resources to them. To cover this, the set of all activities is partitioned into sets of activities H_u, $u = 1, \dots, U$. The activities of each set H_u must be performed in the same mode. That is, $m_i = m_j$ is must hold for all activities $i, j \in H_u$ (note that this requires $M_i = M_j$).

Schultmann and Rentz [160] present a case study that demonstrates how the multi-mode RCPSP can be applied to projects which consist of the dismantling of buildings.

2.5 Tradeoff Problems

In the *discrete time-resource tradeoff problem*, the workload ψ_j for each activity j with regard to a single renewable resource $k = 1$ is given. Activity j can be performed in each discrete combination of processing time p_j and resource request r_{jk} that allows to reach the workload, that is, $p_j \cdot r_{jk} \geq \psi_j$. Note that a combination of p_j and r_{jk} must be considered only if it is efficient, that is, $(p_j - 1) \cdot r_{jk} < \psi_j$ and $p_j \cdot (r_{jk} - 1) < \psi_j$. This problem setting has been discussed by Demeulemeester et al. [53] and Ranjbar and Kianfar [151]. Ranjbar et al. [153] consider the case of multiple renewable resources.

The *discrete time-cost tradeoff problem* includes one nonrenewable resource (which is interpreted as the budget of the project) and no renewable resource; for a description of nonrenewable resources refer to Section 4.1. Demeulemeester et al. [52] consider this problem setting with makespan minimization as objective, and they also discuss a "dual" version with a deadline and a resource-based objective (see Section 5.4). A further approach to the discrete time-cost tradeoff problem has been presented by Akkan et al. [4].

Note that these two discrete tradeoff problems are special cases of the multi-mode RCPSP and, thus, are covered by $MPS|prec|C_{\max}$ employing the notation of Brucker et al. [31]. A mode can be defined for each efficient combination of processing time and resource request that covers the workload. The difference is that tradeoff problems usually specify the workload whereas multi-mode problems list every possible mode explicitly. Due to this difference, we propose to use $\alpha = T_{tr}PS$ and $\alpha = T_{tc}PS$ to specify the discrete time-resource and the discrete time-cost tradeoff problem, respectively, considering the notation of Brucker et al. [31].

Deckro et al. [49] consider the continuous version of the time-cost tradeoff problem where activity durations are not limited to discrete values. A deadline is imposed, and the cost function is quadratic and increases with an increasing deviation of the actual duration from the given normal duration. Extensions include a budget contraint and bonus or penalty payments in case of early or late project completion, respectively.

2.6 Further Activity Concepts

Drexl et al. [64] introduce the concept of forbidden periods. Each activity is associated with a set of periods in which the activity may not be carried out.

In the classical RCPSP, the activity durations are assumed to be integer-valued. Icmeli and Rom [92] and Rom et al. [154] allow the durations to be continuous. They present a mathematical model with a continuous timeline that is not partitioned into periods of equal length.

3 Generalized Temporal Constraints

3.1 Minimal Time Lags

In the classical RCPSP, an activity must have finished before any of its successors can be started. This basic precedence concept can be extended by so-called minimal time lags d_{ij}^{FS} between the completion time C_i of an activity i and the start time S_j of successor activity j, which leads to constraints $C_i + d_{ij}^{FS} \leq S_j$ (note that we always have $d_{ij}^{FS} = 0$ in the standard RCPSP). Allowing negative minimal time lags implies that the corresponding activities may overlap. Minimal time lags are captured by the value *temp* for β in the $\alpha|\beta|\gamma$ notation of Brucker et al. [31].

Minimal time-lags have recently been considered by, e.g., Chassiakos and Sakellaropoulos [37], Klein [101, 102], Klein and Scholl [103, 104], Kolisch [106], and Vanhoucke [180]. Demeulemeester and Herroelen [50] show how minimal time-lags can be used to capture sequence-independent setup times as well as batches in production projects. Drexl et al. [64] consider a multi-mode problem with minimum time-lags that depend on the modes.

In addition to a minimal time lag between the completion time of activity i and the start time of activity j, we may also consider minimal time lags between the start time of i and the start time of j, between the start time of i and the completion time of j, and between the finish time of i and the completion time of j. Note that these four types of minimal time-lags can all be transformed into each other such that one type is sufficient to obtain the full expressive power of minimal time lags. We omit further details here and refer to the transformation rules provided in Bartusch et al. [16]. It should be mentioned, however, that these transformations are not applicable if a multi-mode problem is considered.

3.2 Maximal Time Lags

Similarly to minimal time lags, we denote a maximal time lag between the completion time of activity i and the start time of successsor activity j as $\bar{d}_{ij}^{FS}$, which means that $C_i + \bar{d}_{ij}^{FS} \geq S_j$ must hold. In other words, activity j may not start later than $\bar{d}_{ij}^{FS}$ periods after the completion time of activity i. Analogously, maximal start-start, start-finish, and finish-finish time lags can be given. The RCPSP with minimal and maximal time lags is often referred to as RCPSP/max. Note that maximal time lags might lead to infeasible project instances, and the associated feasibility problem is NP-complete. Like minimal time lags, maximal time lags are covered by *temp* for β in the classification scheme of Brucker et al. [31].

Problems with minimal and maximal time lags have been discussed by, e.g., Bartusch et al. [16], Cesta et al. [36], Dorndorf et al. [58], Neumann and Zimmermann [135, 137], Neumann et al. [138], Schwindt [161], and Schwindt and Trautmann [162]). The concepts of minimal and maximal time lags has also been considered within the multi-mode RCPSP, see Barrios et al. [14], Brucker and Knust [30], Heilmann [82, 83], de Reyck and Herroelen [47] and Calhoun et al. [34]. Sabzehparvar and Seyed-Hosseini [156] consider minimal and maximal mode-dependent time lags, i.e. any time lag between two activities i and j depends on the modes m_i and m_j (thus, a maximal finish-start time lag can be written as $\bar{d}_{im_ijm_j}^{FS}$).

It should be noted that maximal time lags typically lead to cyclic network structures, see Franck and Neumann [70] for a discussion of this issue. A survey of exact and heuristic algorithms for problems with maximal time lags has been given by Neumann and Zimmermann [136].

3.3 Release Dates and Deadlines

A release date r_j is the earliest time at which activity j may be started. Likewise, a deadline d_j denotes the latest time instant at which activity j must be finished. In the notation of Brucker et al. [31], release dates and due dates are represented in the β field by r_j and d_j, respectively. Recent problems with release dates and deadlines include Drezet and Billaut [65], Kis [99, 100], and Klein and Scholl [103, 104]. Baptiste et al. [13] consider what they call the cumulative scheduling problem which is an RCPSP with a single renewable resource and without precedence relations, but with release dates and deadlines for the activities (the objective is to find a feasible schedule).

Note that a release date r_j is equivalent to a minimal finish-start time lag between the dummy source activity and activity j. Analogously, a deadline d_j can be imposed by a maximal finish-finish time lag between the dummy source activity and activity j.

Whereas a deadline as defined above may not be violated, Ballestín et al. [10], Brânzei et al. [26], Chiu and Tsai [40], Neumann et al. [138], and Özdamar et al. [147] consider due dates which can be exceeded at some penalty cost. Objective functions considering due dates are discussed in Section 5.1. Due dates are also important when multiple projects are planned simultaneously, see Section 6.2.

3.4 Time-Switch Constraints

Time-switch constraints have been introduced by Yang and Chen [196]. The planning horizon is divided into cycles of work and rest time windows. An activity can start only in a work window. This approach allows to capture working times (e.g., working is allowed from Monday till Friday but not on weekends). Note that there is a certain relationship to the concept of forbidden periods (see Section 2.6), but a main difference is that the forbidden periods depend on the activities.

Brucker and Knust [30] mention such time windows as well and point out that they can be taken into account by adding a renewable resource with a capacity of one during each work window and 0 otherwise. Vanhoucke et al. [184] employ time-switch constraints within a discrete time-cost tradeoff problem (see 2.5).

3.5 Further Temporal Constraints

Bartels and Zimmermann [15] introduce so-called partially ordered destructive relations for projects that consist of engineering and testing activities in the automotive industry. Such a relation between activities i and j implies that either i must be completed before j starts, or these activities must be executed in different modes (i.e., use different resources). This is relevant for the case that activity j destroys a resource (e.g., a test vehicle which would be destroyed in a crash test) that could also be used by activity i which would not destroy the resource. See also Section 4.3 on the resource types used in this problem setting.

In a case study on a real-world project, Hartmann [80] notes that many medical research projects require that certain activities may not finish in the same period. In order to avoid systematic errors, replications of the same experiment (which correspond to activities) are not allowed to end at the same time. Thereby, it is not prescribed which activity shall finish first. It is shown that this type of temporal constraint is, from a mathematical point of view, a special case of time-dependent requests for renewable resources (see Section 2.2).

Brucker and Knust [30] introduce parallelity constraints which force two activities to be processed in parallel for at least one period. They also discuss the opposite case in which two activities may not be processed in parallel at all. The latter is extended by minimal separating times between two activities which may not be carried out in parallel. Note that a minimal separating time is different from a minimal time lag since the former is not associated with a precedence relation between the activities. Finally, Brucker and Knust [30] introduce so-called bounding constraints which require that a set of activities is carried out for at least l and at most u time units within a given set of periods.

Kis [100] describes so-called feeding precedence constraints. Each precedence relation between two activities i and j is associated with a percentage g_{ij} which indicates that j may not start before g_{ij} percent of i are completed. Moreover, at any point in time the percentage of j that is completed may not be greater than the percentage of i that is completed.

Bomsdorf and Derigs [23] consider a class of precedence relations where no activity k may be scheduled between two precedence-related activities i and j.

Nonobe and Ibaraki [141] use a similar concept which adds a dependency on renewable resources. An immediate precedence constraint $i \prec\prec_r j$ between activities i and j implies that i must be finished before j is started and that no other activity k that requires resource r is allowed to be started between the completion time of i and the start time of j. Nonobe and Ibaraki [141] use this to ensure that no other activity is started between a setup activity and the activity that requires the setup. They also mention that these immediate precedence constraints can be used to enforce equal start times for activities or to enforce that two activities overlap.

Krüger and Scholl [115] introduce temporal constraints caused by resource transfers. Transferring resource r from the location where activity i is carried out to the location of activity j requires Δ_{ijr} time units. In an extended formulation, executing a resource transfer may require another resource.

3.6 Logical Dependencies

Several researchers, e.g., Elmaghraby [67] and Belhe and Kusiak [17], have extended the precedence network concept by adding logical nodes. In the RCPSP, an activity node which has several successor nodes implies that this activity must be finished before any of the successors may be started. This controls the temporal arrangement of the activities. As *all* successor activities must be executed, the activity nodes can be viewed as logical “AND” nodes. Extending this concept, some activity nodes might be declared as logical “OR” nodes, implying that at least one successor activity of an “OR” node has to be performed. Analogously, “XOR” (exclusive-or) nodes induce that exactly one successor activity must be executed. These network extensions are, for example, useful to model research and development projects: After an “XOR”-activity “test prototype” has been performed, either activity “modify prototype” or activity “start production” is executed, depending on the outcome of the test. In the project management literature, networks with logical nodes are mostly treated in connection with stochastic concepts, see Neumann [133].

Another type of logical dependencies has been proposed by Kuster and Jannach [119]. A so-called mutual exclusion relationship between activities i and j implies that if i is executed then j shall not be executed. Analogously, a mutual inclusion relationship between activities i and j means that if i is executed then j shall be executed as well. The resulting problem is used for scheduling ground processes on airports such as fueling, cleaning and catering.

3.7 Network Representations

Finally, we briefly address the way the activities are represented within the network. In the RCPSP and many of its extensions, the so-called *activity-on-node* representation is used, that is, each activity corresponds to a node in the network, while the precedence relations are given by arcs between the nodes. In the literature, also an alternative representation can be found, namely the *actvity-on-arc* network. There, each activity corresponds to an arc in the network, while the nodes represent events. This representation has been used by, e.g., Achuthan and Hardjawidjaja [2], Brânzei et al. [26], Elmaghraby [66], Smith-Daniels et al. [167], and Tareghian and Taheri [171]. For a brief discussion of the similarities and differences between both representations, we refer to Kolisch and Padman [111].

4 Generalized Resource Constraints

4.1 Nonrenewable and Doubly Constrained Resources

The basic RCPSP features only one type of resources which is called renewable because it is avialable in each period with its full capacity. In project scheduling problems with multiple modes ($MPS|prec|C_{\max}$, see Section 2.4), often three different kinds of resources are considered, namely renewable, nonrenewable, and doubly constrained resources. This resource categorization has been developed by Słowiński [165] and Węglarz [191].

Renewable resources such as manpower and machines are limited on a per-period basis. In contrast, nonrenewable resources have a limited capacity for the entire project, an example would be a budget for the project. While renewable resources are already part of the basic (single-mode) RCPSP, nonrenewable ones need only be considered in a multi-mode environment. In the classification scheme of Brucker et al. [31], we propose to reflect a multi-mode problem with both renewable and nonrenewable resource with $\alpha = MPS;R;N$. This slightly modifies the original notation of Brucker et al. [31], but it enables us to integrate further resource categories throughout the following sections.

Doubly constrained resources are limited both for each period and for the whole project (money is an example if both the budget and the per-period cashflow of the project are limited). As a doubly constrained resource can be incorporated by a renewable and a nonrenewable resource, doubly constrained resources do not enhance the expressive power of the problem setting and need not be considered explicitly.

4.2 Partially Renewable Resources

Partially renewable resources (which generalize both renewable and nonrenewable resources) have been introduced by Böttcher et al. [24]; see also Alvarez-Valdes et al. [7, 8], Drexl et al. [63, 64] and Schirmer and Drexl [159]. For each partially renewable resource k, we have a set Π_k of so-called period subsets. Resource k has a total availability of $R_k^{\pi}(Q_{ki})$ during the time given by period subset $Q_{ki} \subseteq \{1, \ldots, T\}$, where $Q_{ki} \in \Pi_k$ and T is the planning horizon. The request of activity j for partially renewable resource k is again denoted as r_{jk}. In an equivalent normalized formulation, a separate partially renewable resource is defined for each period subset. Considering the notation of Brucker et al. [31], we propose to use $\alpha = PS;PR$ to reflect project scheduling with partially renewable resources.

For illustration of this concept, consider an employee who may work on every day from Monday through Friday and either on Saturday or on Sunday but not both. Obviously, each day (Monday through Sunday) would be one period. A partially renewable resource would then be defined as follows: Each day from Monday through Friday makes up one separate period subset with an associated availability of 1. The weekend restriction is imposed by defining another period subset containing both Saturday and Sunday with a capacity of 1.

Zhu et al. [198] integrate partially renewable resources into the multi-mode RCPSP. Nudtasomboon and Randhawa [144] employ a special case of partially renewable resources where each period subset is an interval $[t_1, t_2]$.

4.3 Cumulative Resources

Cumulative resources have been introduced by Neumann and Schwindt [134] who incorporate them into an RCPSP with minimal and maximal time lags. If a project represents a production process, then an activity may require some intermediate product which is withdrawn from the storage area, or it may manufacture a product which is put into the storage area. A cumulative resource k is given by the capacity $\overline{R}_k$ and the minimum inventory level (or safety stock) $\underline{R}_k$ of the storage area. An activity j may produce r_{jk} items (then we have $r_{jk} > 0$) or withdraw r_{jk} items (then we have $r_{jk} < 0$); otherwise we have $r_{jk} = 0$ (note that events rather than activities are used to describe production and withdrawal formally). Neumann and Schwindt [134] point out that cumulative resources subsume both renewable and nonrenewable resources. We propose to use $\alpha = PS;Cu$ to specify the RCPSP with cumulative resources in the notation of Brucker et al. [31].

Bartels and Zimmermann [15] employ cumulative resources within a multi-mode RCPSP with minimal and maximal time lags. They consider a project of engineering and testing activities in the automotive industry. A test vehicle is modeled as a cumulative resource since it can be built, used (which temporarily withdraws its capacity) and destroyed in a crash test (which withdraws its capacity until the project end).

Neumann et al. [139] use cumulative resources in an RCPSP incorporating minimal and maximal time lags to capture a scheduling problem related to batch production in the process industry. Schwindt and Trautmann [162] further extend this setting.

4.4 Continuous Resources

In the problem settings discussed up to now, the considered resources were available in discrete quantities only, as in the cases of, e.g., manpower and machines. Węglarz et al. [194] generalized the concept of renewable resources by allowing continuously divisible resources. This is useful if the resources correspond to, e.g., energy or raw material like liquids. An overview is provided in Blazewicz et al. [22]. Considering the notation of Brucker et al. [31], we suggest to use $\alpha = PS;Co$ to reflect the RCPSP with continuous resources.

Further problems with continuous resources have been considered by Józefowska et al. [96], Kis [99, 100] and Waligóra [190]. Węglarz [192] presents an extension which is essentially a continuous variant of the doubly constrained resource concept.

4.5 Dedicated Resources

Dedicated resources are resources that can be assigned to only one activity at a time. They can be expressed by a renewable resources with an availability of one unit per period, $R_k = 1$. Consequently, they are included in the RCPSP as a special case. Project scheduling problems with dedicated resources are studied in Bianco et al. [19, 20]. Dorndorf et al. [57] refer to the RCPSP with dedicated resources as *disjunctive scheduling problem*, see also Dorndorf et al. [59]. In the notation of Brucker et al. [31], dedicated resources can be covered by setting $\alpha = PS; Rm, 1, 1$ which means that we have m renewable resources with a capacity of one unit and a request of at most one unit.

4.6 Resource Capacities Varying with Time

Up to this point, the availability of the renewable resources has been assumed to be constant over time, i.e., the capacity is the same in each period. This assumption might be too strict in some cases such as changing availability of workers due to vacation or varying equipment capacities due to maintenance. To capture resource availabilities varying with time, we denote the capacity of renewable resource k in period t as R_{kt}. Time-dependent resource capacities have been discussed by, e.g., Bomsdorf and Derigs [23], Klein [101, 102], Klein and Scholl [103], Nonobe and Ibaraki [141], Pesch [149] and Schwindt and Trautmann [162]. Hartmann [80] uses time-dependent capacities to capture the availability of researchers and laboratory equipment in a medical research project. Resource capacities varying with time are sometimes discussed together with preemptive scheduling, see Section 2.1 and the references cited therein. In the notation of Brucker et al. [31], time-varying capacities of renewable resources can be captured using $\alpha = PS; Rm, \cdot, \cdot$ which indicates that we have m renewable resources with a time-dependent capacity and an arbitrary request.

Brucker and Knust [30] consider so-called disjunctive resources which have time-dependent capacities of up to 1, that is, $R_{kt} \in \{0, 1\}$. Observe that these resources are a time-dependent version of the dedicated resources described in Section 4.5.

Icmeli and Rom [92] deal with time-dependent capacities in a problem with a continuous time-line where changes in the resource availability occur at certain points in time called milestones.

Bartusch et al. [16] show that resource capacities varying with time can be transformed into constant capacities if minimal and maximal time lags are available. The constant capacity would be defined as the maximum of the time-dependent capacity over time, and for each time interval with a smaller capacity, an artificial activity is defined to reduce the capacity appropriately. Each artificial activity is fixed to the desired time interval using a minimal and a maximal time lag.

Also note that the RCPSP with time-varying resource capacities is a special case of the RCPSP with partially renewable resources, since we can define a period subset for each period with an individual capacity.

5 Alternative Objectives

5.1 Time-Based Objectives

While the minimization of the makespan is among the most popular objectives (see, e.g., the references in Kolisch and Hartmann [108]), there are various other time-based objectives. Objectives based on lateness, tardiness, and earliness are of particular importance. The lateness L_j of an activity j is the deviation of the completion time C_j from a given due date d_j, hence $L_j = C_j - d_j$. The tardiness T_j is similar but it cannot be negative; it is defined as $T_j = \max\{0, C_j - d_j\}$. The earliness E_j is defined analogously as $E_j = \max\{0, d_j - C_j\}$.

Ballestín et al. [10], Kolisch [106], Nudtasomboon and Randhawa [144], and Viana and de Sousa [188] consider the minimization of the weighted tardiness. Note that this generalizes the makespan objective. Neumann et al. [138] describe the minimization of the maximum lateness and of the weighted total tardiness. Vanhoucke et al. [182] discuss a just-in-time objective which is achieved by minimizing the weighted sum of all earliness and tardiness values. Franck and Schwindt [71] mention a multi-mode RCPSP with the objective to minimize the sum of the earliness and tardiness values. An alternative objective minimizes the maximum value of all earliness and tardiness values. Lorenzoni et al. [122] propose a variant where earliness and tardiness are measured w.r.t. a given time window in which an activity should be carried out.

Vanhoucke [179] defines a set of time windows for each activity. Carrying out an activity within one of its time windows is desired due to quality considerations. The objective function minimizes penalties that are caused by executing activities outside their time windows. The approach is motivated by a bio-technology project. Vanhoucke [179] notes that this concept is similar to the time-switch constraints (see Section 3.4) where time windows are reflected by hard constraints.

Nudtasomboon and Randhawa [144] propose to minimize the sum of all activity completion times, while Rom et al. [154] minimize the weighted sum of the completion times. Similarly, Nazareth et al. [132] suggest to minimize the mean flow time, which is the average of all activity completion times. Note that minimization of total completion time and minimization of average completion time is equivalent.

In the $\alpha|\beta|\gamma$ notation of Brucker et al. [31], these objectives can be captured in the γ field by symbols that are well known from machine scheduling: We can use $\gamma = C_{\max}$ for the makespan, $\gamma = L_{\max}$ for the maximal lateness, $\gamma = \sum w_j C_j$ for total weighted completion time, $\gamma = \sum w_j T_j$ for total weighted tardiness, $\gamma = \sum w_j U_j$ for weighted number of tardy jobs, or $\gamma = \sum (w_j^e E_j + w_j^t T_j)$ for sum of total weighted earliness and total weighted tardiness (w_j is the weight of activity j in the objective and $U_j \in \{0,1\}$ indicates whether activity j is late or not).

5.2 Robustness-Based Objectives

During the execution of a project, delays may occur that could not be foreseen when the schedule was determined. Therefore, a project manager might be interested in a robust schedule, i.e. a schedules in which a delay has only a limited effect. This approach is often referred to as proactive scheduling. (While there are several robustness-oriented approaches that make use of stochastic concepts to capture the uncertainty explicitly, we restrict ourselves to deterministic problems.)

Al-Fawzan and Haouari [5] define the free slack sl_j of activity j as the amount of time the completion of j can be delayed without affecting any other activity. They propose a bi-objective problem incorporating makespan minimization and maximization of total free slack which is used as a measure of robustness. This problem setting is also treated by Abbasi et al. [1]. Kobylanski and Kuchta [105] provide a discussion of this approach and propose to use maximization of the minimum free slack $\min_i sl_i$ as sole objective. Quality according to the makespan is assured using an additional project deadline constraint.

Chtourou and Haouari [41] follow the approach of Al-Fawzan and Haouari [5] and add further measures of robustness. They suggest to weight the free slack of an activity with the number of its immediate successors and/or the sum of its resource requests. Moreover, they outline alternative measures in which the free slack sl_j of activity j is replaced with a binary value $\alpha_j = 0$ if $sl_j = 0$ and $\alpha_j = 1$ if $sl_j > 0$. The idea behind the latter approach is to avoid a bias caused by a very large free slack values. Such large slacks might not be particularly important, given that a smaller slack is already a sufficient buffer for typical delays.

Icmeli-Tukel and Rom [93] assume that activities need to be reworked. RT_{jt} is the expected rework time (i.e., delay) of activity j when completed at time t, while RC_{jt} is the expected rework cost. It is assumed that RT_{jt} and RC_{jt} increase with time. The objective is to find a schedule with minimal sum of rework times and costs.

5.3 Objectives for Rescheduling

Rescheduling is necessary if the project is already in progress, but due to unexpected events (e.g., delays) the schedule that has been calculated before the start of the project is no longer valid. In such a situation, the problem's characteristics may have changed: Some activities might already be finished and can be ignored, some activities might be in progress and must be considered fixed (as long as no preemption is allowed), and the resource availability may have changed and might even have switched from time-independent to time-dependent. In contrast to proactive scheduling which anticipates disruptions by building robust schedules, we now consider the case that some disruption has occured and a new schedule has to be determined. This case is often referred to as reactive scheduling.

Calhoun et al. [34] propose to minimize the perturbation of the original schedule by minimizing the number of activities that receive a different start time in the new schedule. Van de Vonder et al. [177] propose to reschedule the remaining activities such that the sum of deviations of the new finishing times from the original ones is minimized. They refer to a just in time problem developed by Vanhoucke et al. [182] where the weighted sum of earliness and tardiness of each activity is minized. They suggest to use the original finishing times as due dates in order to obtain a just-in-time problem. Sakkout and Wallace [157] measure the perturbation to be minimized as the sum of deviations of starting and finishing times of all activities.

Zhu et al. [197] propose to penalize changes in resource utilization and in the selected modes in a multi-mode RCPSP. Moreover, it is considered to fix some activities by additional constraints. This can be useful if the activities have been scheduled that close to the current point of time that no change is possible, or if the schedule should catch up with the original one at a specific time in the future.

5.4 Objectives Based on Renewable Resources

Various objectives related to renewable resources have been discussed in the literature. A comprehensive overview is provided by Neumann et al. [140].

In the basic RCPSP, the makespan is to be minimized while a given capacity level of each renewable resource has to be observed. A "dual" version of this is the *resource investment problem* where the costs for providing a certain capacity level are to be minimized while a deadline for the project has to be observed. The objective is to minimize the sum of availability costs of all resources, i.e. $\sum_k C_k(R_k)$, where C_k is a discrete non-decreasing cost function of resource k and the capacity levels R_k are variables. A special case is obtained from the linear cost function $C_k(R_k) = c_k \cdot R_k$, where c_k is the per-unit cost of resource k. The resource investment problem has recently been tackled by Drexl and Kimms [61], Neumann and Zimmermann [137], Neumann et al. [138], Ranjbar et al. [152], and Yamashita et al. [195].

Shadrokh and Kianfar [163] present an extension of the resource investment problem. Instead of a deadline, a due date is used, i.e., delayed completion of the project is allowed. The objective is to minimize the sum of resource availability costs and tardiness penalty. The latter results from a constant cost factor for each period the project is delayed.

Nübel [143] suggests the *resource renting problem* in which the renewable resources have to be rented. Renting resource k is associated with fixed costs c_k^f per unit and variable costs c_k^v per unit and period. Thus renting a units of resource k over t periods leads to costs of $a \cdot (c_k^f + t \cdot c_k^v)$. The fixed costs could be the delivery costs of the resource to the project's location while the variable costs represent the actual rent. The objective is to minimize the total renting costs. Observe that $c_k^v = 0$ leads to the resource investment problem. The resource renting problem has been picked up by Ballestín [9].

Another resource-based objective is to achieve a smooth resource profile, which leads to the *resource leveling problem*. The objective there is to minimize the changes in the level of resource usage from period to period in the schedule while a deadline has to be observed. This might be measured as the maximum change or the sum of all changes, see Neumann and Zimmermann [137], Neumann et al. [138] and Nudtasomboon and Randhawa [144]. Bandelloni et al. [12] propose to minimize the sum of all squared changes.

Davis et al. [44], Neumann and Zimmermann [137] and Viana and de Sousa [188] propose to minimize the utilized renewable resource units of each resource that exceed a given level. Nudtasomboon and Randhawa [144] minimize the cumulated deviations of the resource utilization from a given level, i.e., they consider both over- and underutilization. Bomsdorf and Derigs [23] discuss the minimization of both the number and the length of gaps in the resource profile. Kis [99] distinguishes between internal and external capacities of the renewable resources. The objective is to minimize the expenses for using external capacities.

In the notation of Brucker et al. [31], these objectives are represented by $\sum c_k f(r_k(S,t))$ in the γ field. Here c_k represent the cost per unit of resource k and f is a function of resource utilization profile $r_k(S,t)$. Note that the resource investment problem is obtained by setting $f = \max$.

5.5 Objectives Based on Nonrenewable Resources

The multi-mode RCPSP (see Section 2.4) requires that the resource capacities are not exceeded while the makespan is minimized. Analogously to the resource investment problem of Section 5.4, an alternative setting would be to impose a deadline on the project while minimizing the consumption of the nonrenewable resources.

Such a nonrenewable resource-based objective has been considered by Akkan et al. [4] and Demeulemeester et al. [52] within the discrete time-cost tradeoff problem, where the only nonrenewable resource is interpreted as money. Similarly, Nudtasomboon and Randhawa [144] and Tareghian and Taheri [171] minimize the consumption of a nonrenewable resource which again represents money. Nudtasomboon and Randhawa [144] and Viana and de Sousa [188] suggest to minimize the consumed nonrenewable resource units that exceed the capacities. These objectives can be specified by $\gamma = \sum c_k f(r_k(S))$ in the notation of Brucker et al. [31], where f is a function of the consumption $r_k(S)$ of nonrenewable resource k in schedule S.

5.6 Objectives Based on Costs

Maniezzo and Mingozzi [124] and Möhring et al. [128, 129] consider costs c_{jt} for each activity j that depend on the start time t of j. The objective is to minimize the sum of these costs (the underlying problem is an RCPSP without resources). As pointed out by Möhring et al. [129], this objective includes various other well-known objectives as special cases, such as makespan minimization, earliness-tardiness-based objectives and maximization of the net present value (for the latter see Section 5.7).

Achuthan and Hardjawidjaja [2] minimize total project costs which consist of earliness and tardiness costs with regard to due dates as well as costs related to the durations of the activities (the durations can be shortened at additional costs). The project must be finished not later than a given deadline, and renewable resources are not considered.

Dodin and Elimam [55] aim at minimizing costs which includes costs for activity crashing (i.e., shortening the duration), material costs and inventory holding costs. Based on a given due date, a bonus for early project completion or a penalty for late completion is added.

Nonobe and Ibaraki [142] consider activity durations which have to be between a lower and an upper limit. This leads to an objective that minimizes the costs related to the durations, where the cost function for each activity is convex.

Rummel et al. [155] consider a cost-based objective that consists of two components. The project duration costs are proportional to the makespan. Activities can be combined (consolidated) in order to reduce the makespan, but this causes consolidation costs. The objective is used in a problem without resource constraints.

5.7 Objectives Based on the Net Present Value

Another important type of objective emerges if cash flows occur while the project is carried out. Cash outflows are induced by the execution of activities and the usage of resources. On the other hand, cash inflows result from payments due to the completion of specified parts of the project. Typically, discount rates are also included. Note that cash flows related to activity j might occur at several points in time during execution of j. However, they can easily be compounded to a single cash flow at the beginning or the end of j. These considerations result in problems with the objective to maximize the net present value (NPV) of the project subject to the standard RCPSP constraints. Recent papers include Kimms [98], Mika et al. [125], Padman and Zhu [148], and Vanhoucke et al. [181]. In the notation of Brucker et al. [31], the objective to optimize the NPV is represented by $\sum c_j^F \beta^{C_j}$ for γ, where c_j^F is the cash flow of activity j and β is the discount factor.

Furthermore, NPV objectives have been investigated for the multi-mode RCPSP (see Ulusoy et al. [176], Varma et al. [187], and Waligóra [190]), the resource investment problem (see Najafi and Niaki [131]), and the RCPSP with minimal and maximal time lags (see Neumann and Zimmermann [135, 137]). Icmeli and Rom [92] employ the NPV objective in a problem with continuous activity durations and time-dependent resource capacities. An overview of problem settings with NPV objectives is given by Herroelen et al. [87]. A comparison of problems with cash flow based objectives is provided by Dayanand and Padman [46].

Several variants of the NPV objective mentioned above arise in literature. Instead of fixed cash flows, Etgar et al. [69] consider individual cash flow functions of the completion time for each activity. In Vanhoucke et al. [185], at specific points of time a cash inflow related to activity j occurs which amounts to the fraction of j being finished at this moment. Vanhoucke et al. [183] assume the cash flow of an activity to be a linear non-increasing function of the activity's completion time. Najafi and Niaki [131] consider a cash flow related to a subset of activities, that is, a cash flow is initiated when the last activity of the subset is finished. Furthermore, the cash flow associated with activity j might depend on the mode chosen for j as lined out in Icmeli and Erenguc [91].

Some authors have extended the NPV objective by additional payments upon project completion. Icmeli and Erenguc [90] as well as Özdamar et al. [147] consider a penalty being charged for each period the project is finished after a given due date. Chiu and Tsai [40], Doersch and Patterson [56], and Sung and Lim [170] extend this by including also bonus payments for early completion.

Smith-Daniels et al. [167] and Sung and Lim [170] propose the objective to maximize the discounted amount of cash available in each period. This amount is influenced by cash flows associated with each activity. In Smith-Daniels et al. [167] this is extended by a constraint: An activity j may not be started at a specific point in time if the cash outflow at the beginning of j exceeds the available cash. Cash can be seen as a renewable resource if the outflow equals the inflow for each activity. However, if this does not hold, the capacity of this resource depends on the point in time and the activities being already started (note that there is a similarity to the concept of cumulative resources, see Section 4.3). In Sung and Lim [170] there is no limit for the amount of cash available. In particular, the amount of available cash might be negative which can be interpreted as being indebted.

Ulusoy and Cebelli [175] investigate the negotiation process to find the timing of payments and the amount of each specific payment between a client and a contractor. Obviously, the client seeks to minimize the NPV while the contractor aims at maximizing it. The objective in Ulusoy and Cebelli [175] is to find the payment structure which minimizes each party's loss in comparison to the respective ideal payment structure. Dayanand and Padman [45] treat a similar problem but restrict themselves to the client's point of view. The client might associate a specific value with each event (start or completion of a job). Cash outflows can be assigned to each event having a positive value. The problem is to find a project schedule and decide cash outflows to happen at a given number of events. The total outflow might exceed the total value of finished activities (minus a percentual retention) at no point of time. The objective is to find a solution such that discounted cash inflow (associated with finishing the project) minus total discounted cash outflow is maximized.

5.8 Multiple Objectives

The problems discussed above have a single objective function (e.g., makespan minimization) while all other properties of a schedule are controlled by means of constraints (e.g., resource usages and costs). However, several authors have employed multiple performance measures into their project scheduling problems.

A widely used approach to cope with multiple objectives is to define one overall objective as the weighted sum of all performance measures considered. This is done by Nudtasomboon and Randhawa [144] who include various objectives into the multi-mode RCPSP such as makespan, weighted tardiness, resource leveling and nonrenewable resource consumption. Voß and Witt [189] employ the multi-mode RCPSP with an objective that contains makespan, weighted tardiness and setup costs. The inclusion of setup costs supports batching of activities. The problem setting is motivated by a production planning problem at a steel manufacturer. Bomsdorf and Derigs [23] employ an objective for movie shooting projects that consists of several components which are allowed to be squared. The components include specific criteria such as the minimization of location changes over time (each activity is associated with a location). Al-Fawzan and Haouari [5] combine makespan minimization and maximization of total free slack into one objective.

Another way to deal with multiple objectives is the generation of Pareto-optimal schedules. This approach is followed by several authors. Davis et al. [44] minimize the makespan as well as the overutilization of each renewable resource. Viana and de Sousa [188] add to these the minimization of overutilization of each nonrenewable resource and the mean weighted tardiness. Hapke et al. [77] propose a multi-criteria approach which allows to simultaneously consider several objectives, namely time based, resource based, and financial ones. Słowiński et al. [166] consider the multi-mode RCPSP with various objectives including makespan, mean weighted lateness, total number of tardy activities, smoothness of the resource profile, total and weighted resource consumption and net present value. Dörner et al. [60] employ three objectives within a variant of the time-cost tradeoff problem. The first objective is a function of the project makespan, while the second and the third are functions of the monetary and nonmonetary costs for crashing the activities, respectively. The case of Pareto-optimal schedules can be represented in the classification scheme of Brucker et al. [31] by setting γ to a vector containing all symbols of the objectives under consideration.

Nabrzynski and Węglarz [130] present a knowledge-based approach to a project scheduling problem with multiple modes and a set of time and cost criteria. The latter includes the makespan, the smoothness of the resource profile, mean weighted lateness, mean weighted flow time, total and weighted resource consumption as well as the net present value.

6 Multiple Projects

6.1 Networks for Multiple Projects

In practice, often not only one but several dependent projects have to be scheduled simultaneously. This is important if two or more projects which may be processed in parallel share at least one resource. Herroelen [85] stresses the importance of frameworks for scheduling multiple projects.

The most common way to deal with multiple projects is to comprise their networks into a "super-network" by adding a "super-source" and a "super-sink," while a common pool of resources is considered. This approach has been proposed by Pritsker et al. [150]. Confessore et al. [42] consider a set of projects in which each project has its own resources while one additional resource is shared by the projects. The advantage of integrating multiple projects in a single network is that this provides a formal basis for the application of scheduling methods for single projects also to the case of multiple projects.

As for the single-project case, further constraints can be added. Pritsker et al. [150] add due dates and deadlines for the sink activities of the single projects. Franck et al. [72] consider a network of multiple projects with minimal and maximal time lags. Krüger and Scholl [115] discuss a multi-project problem that includes resource transfers. Kumanan et al. [116] consider multiple projects the activities of which can be performed in one of several modes.

6.2 Objectives for Multi-Project Scheduling

Several different objectives for scheduling multiple projects have been discussed. In many multi-project problems, each project is associated with a due date, and the tardiness appears to be among the most widely used performance measures for multi-project scheduling.

Chen [39] employs the minimization of weighted tardiness as well as the minimization of costs that exceed the budget of each individual project as well as the total budget. Chiu and Tsai [40] deal with an NPV objective that includes earliness and tardiness payments with regard to the project due dates. Franck et al. [72] consider resource leveling as well as resource investment objectives. Goncalves et al. [74] aim at minimizing the weighted sum of earliness and tardiness as well as the flow time of each project. Homberger [89] discusses the minimization of the average makespan of the projects, where the makespan of each individual project is defined as its completion time minus its release date. Lawrence and Morton [120] deal with the minimization of weighted tardiness. Lova et al. [123] consider various objectives such as mean tardiness, makespan of the "super-project," resource leveling and project splitting. The latter is defined as the number of periods a project is interrupted, i.e., no activity of this project is carried out after the project has been started and before it is finished (but probably activities related to other projects).

Kurtulus and Davis [118] consider the total delay, where the delay of each project is measured as the difference between completion time in the actual schedule and completion time in the resource-unconstrained case. Kurtulus [117] extend this approach by defining several functions that assign different delay penalties to the projects. Browning and Yassine [27] follow this definition of delays and discuss objectives including average delay per project, average percentage delay per project (measured as percentage of the critical path length) as well as percentage delay of the "super-network."

6.3 Project Selection and Scheduling

Chen and Askin [38] develop a multi-project problem which contains two types of decisions: First, a set of candidate projects is given, and the projects to be carried out are selected (this allows to decide, e.g., which product development projects should be executed). Second, the selected projects are scheduled subject to the usual precedence and renewable resource constraints. The project selection and scheduling decisions are made simultaneously with the objective to maximize the net present value.

Kolisch and Meyer [110] integrate project selection and scheduling into a problem for pharmaceutical research projects. The selection of projects is captured by modes reflecting that an activity is not executed, while the mode identity concept ensures that either all activities of a project are carried out or none. The problem further includes time-dependent resource requests and an objective that is based on the net present value.

6.4 Further Specific Multi-Project Problems

Heimerl and Kolisch [84] propose a multi-project problem where the schedule of each project is fixed. Consequently, each project can be viewed as an activity with time-dependent resource requirements. The resources are the internal and external workforce where different skills and efficiencies are taken into account. The objective is to minimize costs for regular and overtime work as well as for external resources.

Kolisch [106] presents a problem for scheduling multiple assembly projects. It includes due dates for the individual projects and tardiness minimization. Only a limited number of individual projects can be in progress at the same time because space in the assembly area is limited. Moreover, an activity can only start when all parts required for the assembly are available (note that this is similar to the concept of cumulative resources, see Section 4.3).

Shtub et al. [164] develop multi-project problems which assume that a project is repeated several times, as it may occur in small batch production. Their problem settings consider learning effects that can be exploited if a (human) resource performs several identical projects, that is, the time needed for one project may decrease.

Tukel and Wasti [174] discuss problems for product development projects which include outsourcing of sub-projects to suppliers. The "contractual approach" assumes that the suppliers schedule their sub-projects in the first step. The second step is the scheduling of the main project, where each sub-project of a supplier is treated as a single activity with the duration corresponding to the sub-project's makespan. The "partnership approach" is a multi-project problem setting that integrates the sub-project's activities into the overall project network.

Vanhoucke [180] considers a project which is repeated several times. The resulting projects are represented in a super-network as described above, and the super-sink is associated with a deadline. An activity requires the same resource in each project (which is referred to as work continuity), and the objective is to minimize the resource idle time.

7 Test Instance Generators

When testing exact or heuristic methods for project scheduling, test instances are necessary. In recent years, several parameter-driven instance generators for the RCPSP and many of its extensions have been developed. This section gives a brief overview of the problem variants that have been covered (a discussion of the control parameters is beyond the scope of this work).

Kolisch et al. [113] develop a generator called ProGen for the classical RCPSP as well as the multi-mode extension. It has been applied to generate several sets with test instances for these problem classes. These sets which are comprised in the internet-based library PSPLIB (see Kolisch and Sprecher [112] and Kolisch et al. [114]) have been used in a huge number of studies.

ProGen has been extended to cover further extensions of the RCPSP. Schwindt [161] develops a version called ProGen/max in order to include minimal and maximal time lags. This generator can also produce activities with multiple modes as well as instances for the resource leveling and the resource investment problem. Drexl et al. [64] introduce a version called ProGen/πx which incorporates partially renewable resources, mode identity, mode-dependent and sequence-dependent setup times (called changeover times here), mode-dependent minimum time lags, and forbidden periods for activities.

Agrawal et al. [3] suggest a generator called DAGEN for activity-on-arc project networks. Demeulemeester et al. [54] develop RanGen, a generator for single-mode and multi-mode project instances which is based on different control parameters than ProGen. Vanhoucke et al. [186] provide RanGen2, which extends RanGen by incorporating further topological network measures. Browning and Yassine [28] propose a generator for problems consisting of multiple projects where the activities are associated with due dates.

8 Conclusions

We have summarized and classified publications on various variants and generalizations of the well-known resource-constrained project scheduling problem (RCPSP). Among the most popular extensions are multiple modes, generalized time lags, and objectives based on the net present value. Beyond these well-researched areas, many other concepts have been developed in recent years. This paper can serve as a guide through these developments.

Several recent problem settings have been motivated by specific industries. This includes production and engineering where Bartels and Zimmermann [15] consider the automotive industry, Schwindt and Trautmann [162] deal with the process industry, Kolisch [106] discusses the production of palletizing systems, and Voß and Witt [189] consider a steel manufacturer. Specific requirements of research projects have been discussed for medical research (Hartmann [80]), pharmaceutical research (Kolisch and Meyer [110]), and bio-technology (Vanhoucke [179]). Further examples include training in the telecommunication industry (Tiwari et al. [173]) and aviation industry (Haase et al. [76]), arrivals of ships in ports (Lorenzoni et al. [122]), IT projects (Heimerl and Kolisch [84]) and movie projects (Bomsdorf and Derigs [23]). Also note that some of the above papers consider ongoing processes rather than projects in the strict sense (i.e., a set of activities which are carried out to achieve a predefined goal). Apparently, project scheduling concepts are powerful enough to capture such problems as well.

The field of project scheduling has also attracted researchers who examined the underlying mathematical structures and their relationships to other optimization problems. For example, Sprecher [168] shows that the flow shop, open shop, job shop, and assembly line balancing problems are special cases of project scheduling problems. Hartmann [78] outlines how bin packing, strip packing, and knapsack problems can be modeled as project scheduling problems. Stadtler [169] compares the multi-level capacitated lotsizing problem and the RCPSP and presents an integrated model. Brucker and Knust [30] and Drexl and Salewski [62] show how school and university timetabling problems can be captured as project scheduling problems.

These studies show that the use of the RCPSP with all its extensions is not limited to applications in its original field, the scheduling of projects. The RCPSP and its generalizations are also perceived as a collection of powerful tools that allow to describe many highly complex optimization problems from other areas. If some problem is captured as a resource-constrained project scheduling problem, it is possible to employ the solution methods that have been developed for the project scheduling problem. Hence, algorithms originally designed for project scheduling can be transferred to problems from other fields. We believe that this underscores the importance of research on models and methods for project scheduling.

References

[1] B. Abbasi, S. Shadrokh, and J. Arkat. Bi-objective resource-constrained project scheduling with robustness and makespan criteria. *Applied Mathematics and Computation*, 180(1):146–152, 2006.

[2] N. Achuthan and A. Hardjawidjaja. Project scheduling under time dependent costs – A branch and bound algorithm. *Annals of Operations Research*, 108(1–4):55–74, 2001.

[3] M. K. Agrawal, S. E. Elmaghraby, and W. S. Herroelen. DAGEN: A generator of testsets for project activity nets. *European Journal of Operational Research*, 90:376–382, 1996.

[4] C. Akkan, A. Drexl, and A. Kimms. Network decomposition-based benchmark results for the discrete time-cost tradeoff problem. *European Journal of Operational Research*, 165:339–358, 2005.

[5] M. Al-Fawzan and M. Haouari. A bi-objective model for robust resource-constrained project scheduling. *International Journal of Production Economics*, 96:175–187, 2005.

[6] J. Alcaraz, C. Maroto, and R. Ruiz. Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms. *Journal of the Operational Research Society*, 54:614–626, 2003.

[7] R. Alvarez-Valdes, E. Crespo, J. M. Tamarit, and F. Villa. A scatter search algorithm for project scheduling under partially renewable resources. *Journal of Heuristics*, 12(1-2):95–113, 2006.

[8] R. Alvarez-Valdes, E. Crespo, J. M. Tamarit, and F. Villa. GRASP and path relinking for project scheduling under partially renewable resources. *European Journal of Operational Research*, 189 (3):1153–1170, 2008.

[9] F. Ballestín. A genetic algorithm for the resource renting problem with minimum and maximum time lags. *Lecture Notes in Computer Science*, 4446:25–35, 2007.

[10] F. Ballestín, V. Valls, and S. Quintanilla. Due dates and RCPSP. In Józefowska and Węglarz [95], pages 131–163.

[11] F. Ballestin, V. Valls, and S. Quintanilla. Preemption in resource-constrained project scheduling. *European Journal of Operational Research*, 189(3):1136–1152, 2008.

[12] M. Bandelloni, M. Tucci, and R. Rinaldi. Optimal resource leveling using non-serial dyanamic programming. *European Journal of Operational Research*, 78(2):162–177, 1994.

[13] P. Baptiste, C. L. Pape, and W. Nuijten. Satisfiability tests and time-bound adjustments for cumulative scheduling problems. *Annals of Operations Research*, 92:305333, 1999.

[14] A. Barrios, F. Ballestin, and V. Valls. A double genetic algorithm for the MRCPSP/max. *Computers & Operations Research*, 2009. Forthcoming.

[15] J.-H. Bartels and J. Zimmermann. Scheduling tests in automotive R&D projects. *European Journal of Operational Research*, 193(3):805–819, 2009.

[16] M. Bartusch, R. H. Möhring, and F. J. Radermacher. Scheduling project networks with resource constraints and time windows. *Annals of Operations Research*, 16:201–240, 1988.

[17] U. Belhe and A. Kusiak. Resource-constrained scheduling of hierarchically structured design activity networks. *IEEE Transactions on Engineering Management*, 52:150–158, 1995.

[18] O. Bellenguez and E. Néron. Lower bounds for the multi-skill project scheduling problem with hierarchical levels of skills. *Lecture Notes in Computer Science*, 3616:229–243, 2005.

[19] L. Bianco, P. Dell'Olmo, and M. G. Speranza. Heuristics for multimode scheduling problems with dedicated resources. *European Journal of Operational Research*, 107:260–271, 1998.

[20] L. Bianco, M. Caramia, and P. Dell'Olmo. Solving a preemptive project scheduling problem with coloring techniques. In Węglarz [193], pages 135–146.

[21] J. Blazewicz, J. K. Lenstra, and A. H. G. Rinnooy Kan. Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics*, 5:11–24, 1983.

[22] J. Blazewicz, K. Ecker, E. Pesch, G. Schmidt, and J. Weglarz. *Handbook on Scheduling*. Springer, Berlin, Germany, 2007.

[23] F. Bomsdorf and U. Derigs. A model, heuristic procedure and decision support system for solving the movie shoot scheduling problem. *OR Spectrum*, 30 (4):751–772, 2008.

[24] J. Böttcher, A. Drexl, R. Kolisch, and F. Salewski. Project scheduling under partially renewable resource constraints. *Management Science*, 45:543–559, 1999.

[25] K. Bouleimen and H. Lecocq. A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research*, 149(2):268–281, 2003.

[26] R. Brânzei, G. Ferrari, V. Fragnelli, and S. Tijs. Two approaches to the problem of sharing delay costs in joint projects. *Annals of Operations Research*, 109(1-4):359–374, 2002.

[27] T. R. Browning and A. A. Yassine. Resource-constrained multi-project scheduling: Priority rule performance revisited. Technical report, Texas Christian University, M. J. Neeley School of Business, 2006.

[28] T. R. Browning and A. A. Yassine. A random generator of resource-constrained multi-project scheduling problems. Technical report, Texas Christian University, M. J. Neeley School of Business, 2007.

[29] P. Brucker. Scheduling and constraint propagation. *Discrete Applied Mathematics*, 123(1-3):227–256, 2002.

[30] P. Brucker and S. Knust. Resource-constrained project scheduling and timetabling. *Lecture Notes in Computer Science*, 2079:277–293, 2001.

[31] P. Brucker, A. Drexl, R. Möhring, K. Neumann, and E. Pesch. Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112:3–41, 1999.

[32] J. Buddhakulsomsiria and D. S. Kim. Properties of multi-mode resource-constrained project scheduling problems with resource vacations and activity splitting. *European Journal of Operational Research*, 175:279–295, 2006.

[33] J. Buddhakulsomsiria and D. S. Kim. Priority rule-based heuristic for multi-mode resource-constrained project scheduling problems with resource vacations and activity splitting. *European Journal of Operational Research*, 178:374–390, 2007.

[34] K. M. Calhoun, R. F. Deckro, J. T. Moore, J. W. Chrissis, and J. C. V. Hove. Planning and replanning in project and production scheduling. *Omega – The international Journal of Management Science*, 30(3):155–170, 2002.

[35] C. C. B. Cavalcante, C. C. de Souza, M. W. P. Savelsbergh, Y. Wang, and L. A. Wolsey. Scheduling projects with labor constraints. *Discrete Applied Mathematics*, 112(1-3):27–52, 2001.

[36] A. Cesta, A. Oddi, and S. F. Smith. A constraint-based method for project scheduling with time windows. *Journal of Heuristics*, 8(1):109–136, 2002.

[37] A. P. Chassiakos and S. P. Sakellaropoulos. Time-cost optimization of construction projects with generalized activity constraints. *Journal of Construction Engineering and Management*, 131:1115–1124, 2005.

[38] J. Chen and R. G. Askin. Project selection, scheduling and resource allocation with time dependent returns. *European Journal of Operational Research*, 193(1):23–34, 2009.

[39] V. Y. X. Chen. A 0-1 goal programming model for scheduling multiple maintenance projects at a copper mine. *European Journal of Operational Research*, 76(1):176–191, 1994.

[40] H. N. Chiu and D. M. Tsai. An efficient search procedure for the resource-constrained multi-project scheduling problem with discounted cash flows. *Construction Management and Economics*, 20: 5566, 2002.

[41] H. Chtourou and M. Haouari. A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling. *Computers & Industrial Engineering*, 55(1):183–194, 2008.

[42] G. Confessore, S. Giordani, and S. Rismondo. A market-based multi-agent system model for decentralized multi-project scheduling. *Annals of Operations Research*, 150(1):115–135, 2007.

[43] J. Damay, A. Quilliot, and E. Sanlaville. Linear programming based algorithms for preemptive and non-preemptive RCPSP. *European Journal of Operational Research*, 182(3):1012–1022, 2007.

[44] K. R. Davis, A. Stam, and R. A. Grzybowski. Resource constrained project scheduling with multiple objectives: A decision support approach. *Computers & Operations Research*, 19(7):657–669, 1992.

[45] N. Dayanand and R. Padman. Project contracts and payment schedules: The client's problem. *Management Science*, 47:1654–1667, 2001.

[46] N. Dayanand and R. Padman. On modelling payments in projects. *Journal of the Operational Research Society*, 48:906–918, 1997.

[47] B. de Reyck and W. S. Herroelen. The multi-mode resource-constrained project scheduling problem with generalized precedence relations. *European Journal of Operational Research*, 119(2):538–556, 1999.

[48] D. Debels and M. Vanhoucke. The impact of various activity assumptions on the lead time and resource utilization of resource-constrained projects. *Computers & Industrial Engineering*, 54:140–154, 2008.

[49] R. F. Deckro, J. E. Hebert, W. A. Verdini, P. H. Grimsrud, and S. Venkateshwar. Nonlinear time/cost tradeoff models in project management. *Computers & Industrial Engineering*, 28(2):219–229, 1995.

[50] E. L. Demeulemeester and W. S. Herroelen. Modelling setup times, process batches and transfer batches using activity network logic. *European Journal of Operational Research*, 89:355–365, 1996.

[51] E. L. Demeulemeester and W. S. Herroelen. An efficient optimal solution procedure for the preemptive resource-constrained project scheduling problem. *European Journal of Operational Research*, 90(2):334–348, 1996.

[52] E. L. Demeulemeester, B. de Reyck, B. Foubert, W. S. Herroelen, and M. Vanhoucke. New computational results on the discrete time/cost trade-off problem in project networks. *Journal of the Operational Research Society*, 49:614–626, 1998.

[53] E. L. Demeulemeester, B. de Reyck, and W. S. Herroelen. The discrete time/resource trade-off problem in project networks – A branch-and-bound approach. *IIE Transactions*, 32:1059–1069, 2000.

[54] E. L. Demeulemeester, M. Vanhoucke, and W. S. Herroelen. A random network generator for activity-on-the-node networks. *Journal of Scheduling*, 6:17–38, 2003.

[55] B. Dodin and A. A. Elimam. Integrated project scheduling and mateiral planning with variable activity duration and rewards. *IIE Transactions*, 33: 1005–1018, 2001.

[56] R. H. Doersch and J. H. Patterson. Scheduling a project to maximize its present value: A zero-one programming approach. *Management Science*, 23: 882–889, 1977.

[57] U. Dorndorf, T. Phan Huy, and E. Pesch. A survey of interval capacity consistency tests for time- and resource constrained scheduling. In Węglarz [193], pages 213–238.

[58] U. Dorndorf, E. Pesch, and T. Phan Huy. A time-oriented branch-and-bound algorithm for resource-constrained project scheduling with generalized precedence constraints. *Management Science*, 46: 1365–1384, 2000.

[59] U. Dorndorf, E. Pesch, and T. Phan Huy. Constraint propagation techniques for the disjunctive scheduling problem. *Artificial Intelligence*, 122:189–240, 2000.

[60] K. F. Dörner, W. J. Gutjahr, R. F. Hartl, C. Strauss, and C. Stummer. Nature-inspired metaheuristics for multiobjective activity crashing. *Omega*, 36: 1019–1037, 2008.

[61] A. Drexl and A. Kimms. Optimization guided lower and upper bounds for the resource investment problem. *Journal of the Operational Research Society*, 52:340–351, 2001.

[62] A. Drexl and F. Salewski. Distribution requirements and compactness constraints in school timetabling. *European Journal of Operational Research*, 102:193–214, 1997.

[63] A. Drexl, J. Juretzka, F. Salewski, and A. Schirmer. New modelling concepts and their impact on resource-constrained project scheduling. In Węglarz [193], pages 413–432.

[64] A. Drexl, R. Nissen, J. H. Patterson, and F. Salewski. Progen/πx - an instance generator for resource-constrained project scheduling problems with partially renewable resources and further extensions. *European Journal of Operational Research*, 125(1):59–72, 2000.

[65] L.-E. Drezet and J.-C. Billaut. A project scheduling problem with labour constraints and time-dependent activities requirements. *European Journal of Operational Research*, 112(1):217–225, 2008.

[66] S. E. Elmaghraby. *Activity networks: Project planning and control by network models*. Wiley, New York, 1977.

[67] S. E. Elmaghraby. An algebra for the analysis of generalized networks. *Management Science*, 10: 419–514, 1964.

[68] S. S. Erenguc, T. Ahn, and D. G. Conway. The resource constrained project scheduling problem with multiple crashable modes: An exact solution method. *Naval Research Logistics*, 48:107–127, 2001.

[69] R. Etgar, A. Shtub, and L. J. LeBlanc. Scheduling projects to maximize net present value - the case of time-dependent, contingent cash flows. *European Journal of Operational Research*, 96(1):90–96, 1997.

[70] B. Franck and K. Neumann. Resource-constrained project scheduling with time windows: Structural questions and priority rule methods. Technical Report WIOR-492, Universität Karlsruhe, Germany, 1997.

[71] B. Franck and C. Schwindt. Different resource-constrained project scheduling models with minimal and maximal time-lags. Technical Report WIOR-450, Universität Karlsruhe, Germany, 1995.

[72] B. Franck, K. Neumann, and C. Schwindt. A capacity-oriented hierarchical approach to single-item and small-batch production using project scheduling methods. *OR Spektrum*, 19:77–85, 1997.

[73] B. Franck, K. Neumann, and C. Schwindt. Project scheduling with calendars. *OR Spektrum*, 23:325–334, 2001.

[74] J. F. Goncalves, J. J. M. Mendes, and M. G. C. Resende. A genetic algorithm for the resource constrained multi-project scheduling problem. *European Journal of Operational Research*, 189(3): 1171–1190, 2008.

[75] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Optimisation and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5:236–287, 1979.

[76] K. Haase, J. Latteier, and A. Schirmer. The course scheduling problem at Lufthansa Technical Training. *European Journal of Operational Research*, 110:441–456, 1998.

[77] M. Hapke, A. Jaszkiewicz, and R. Słowiński. Interactive analysis of multiple-criteria project scheduling problems. *European Journal of Operational Research*, 107:315–324, 1998.

[78] S. Hartmann. Packing problems and project scheduling models: An integrating perspective. *Journal of the Operational Research Society*, 51: 1083–1092, 2000.

[79] S. Hartmann. Project scheduling with multiple modes: A genetic algorithm. *Annals of Operations Research*, 102:111–135, 2001.

[80] S. Hartmann. *Project scheduling under limited resources: Models, methods, and applications*. Number 478 in Lecture Notes in Economics and Mathematical Systems. Springer, Berlin, Germany, 1999.

[81] S. Hartmann and R. Kolisch. Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 127:394–407, 2000.

[82] R. Heilmann. Resource-constrained project scheduling: a heuristic for the multi-mode case. *OR Spektrum*, 23:335–357, 2001.

[83] R. Heilmann. A branch-and-bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags. *European Journal of Operational Research*, 144:348–365, 2003.

[84] C. Heimerl and R. Kolisch. Scheduling and staffing multiple projects with a multi-skilled workforce. *OR Spektrum*, 2009. Forthcoming.

[85] W. S. Herroelen. Project scheduling — Theory and practice. *Production and Operations Management*, 14:413–432, 2005.

[86] W. S. Herroelen and R. Leus. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165 (2):289–306, 2005.

[87] W. S. Herroelen, P. van Dommelen, and E. L. Demeulemeester. Project network models with discounted cash flows: A guided tour through recent developments. *European Journal of Operational Research*, 100:97–121, 1997.

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

[89] J. Homberger. A multi-agent system for the decentralized resource-constrained multi-project scheduling problem. *International Transactions in Operational Research*, 14:565589, 2007.

[90] O. Icmeli and S. S. Erenguc. A branch and bound procedure for the resource constrained project scheduling problem with discounted cash-flows. *Management Science*, 42:1395–1408, 1996.

[91] O. Icmeli and S. S. Erenguc. The resource constrained time/cost tradeoff project scheduling problem with discounted cash flows. *Journal of Operations Management*, 14(3):255–275, 1996.

[92] O. Icmeli and W. O. Rom. Solving the resource constrained project scheduling problem with optimization subroutine library. *Computers & Operations Research*, 23(8):801–817, 1996.

[93] O. Icmeli-Tukel and W. O. Rom. Ensuring quality in resource constrained project scheduling. *European Journal of Operational Research*, 103:483–496, 1997.

[94] B. Jarboui, N. Damak, P. Siarry, and A. Rebai. A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems. *Applied Mathematics and Computation*, 195:299–308, 2008.

[95] J. Józefowska and J. Węglarz, editors. *Perspectives in Modern Project Scheduling*. Springer, Berlin, Germany, 2006.

[96] J. Józefowska, M. Mika, R. Rózycki, and G. W. und Jan Węglarz. Solving the discrete-continuous project scheduling problem via its discretization. *Mathematical Methods of Operations Research*, 52 (3):489–499, 2000.

[97] J. Józefowska, M. Mika, R. Rozycki, G. Waligora, and J. Węglarz. Simulated annealing for multi-mode resource-constrained project scheduling. *Annals of Operations Research*, 102:137–155, 2001.

[98] A. Kimms. Maximizing the net present value of a project under resource constraints using a lagrangian relaxation based heuristic with tight upper bounds. *Annals of Operations Research*, 102(1-4): 221–236, 2001.

[99] T. Kis. A branch-and-cut algorithm for scheduling of projects with variable-intensity activities. *Mathematical Programming A*, 103:515–539, 2005.

[100] T. Kis. RCPS with variable intensity activities and feeding precedence constraints. In Józefowska and Węglarz [95], pages 105–129.

[101] R. Klein. Computing lower bounds by destructive improvement — An application to resource-constrained project scheduling. *International Journal of Production Research*, 38:3937–3952, 2000.

[102] R. Klein. Project scheduling with time-varying resource constraints. *International Journal of Production Research*, 38:3937–3952, 2000.

[103] R. Klein and A. Scholl. Scattered branch and bound — An adaptive search strategy applied to resource-constrained project scheduling. *Central European Journal of Operations Research*, 7:177–201, 2000.

[104] R. Klein and A. Scholl. PROGRESS: optimally solving the generalized resource-constrained project scheduling problem. *Mathematical Methods of Operations Research*, 52(3):467–488, 2000.

[105] P. Kobylanski and D. Kuchta. A note on the paper by m. a. al-fawzan and m. haouari about a bi-objective problem for robust resource-constrained project scheduling. *International Journal of Production Economics*, 107:496–501, 2007.

[106] R. Kolisch. Integrated scheduling, assembly area- and part-assignment for large-scale, make-to-order assemblies. *International Journal of Production Economics*, 64(1-3):127–141, 2000.

[107] R. Kolisch and A. Drexl. Local search for nonpreemptive multi-mode resource-constrained project scheduling. *IIE Transactions*, 29:987–999, 1997.

[108] R. Kolisch and S. Hartmann. Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research*, 174:23–37, 2006.

[109] R. Kolisch and S. Hartmann. Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis. In Węglarz [193], pages 147–178.

[110] R. Kolisch and K. Meyer. Selection and scheduling of pharmaceutical research projects. In J. Jozefowska and J. Weglarz, editors, *Perspectives in Modern Project Scheduling*, pages 321–344. Springer, Berlin, Germany, 2006.

[111] R. Kolisch and R. Padman. An integrated survey of deterministic project scheduling. *Omega*, 29:249–272, 2001.

[112] R. Kolisch and A. Sprecher. PSPLIB – a project scheduling problem library. *European Journal of Operational Research*, 96:205–216, 1996.

[113] R. Kolisch, A. Sprecher, and A. Drexl. Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science*, 41:1693–1703, 1995.

[114] R. Kolisch, A. Sprecher, and C. Schwindt. Benchmark instances for project scheduling problems. In Węglarz [193], pages 197–212.

[115] D. Krüger and A. Scholl. Managing and modelling general resource transfers in (multi-)project scheduling. *OR Spektrum*, 2009. Forthcoming.

[116] S. Kumanan, G. J. Jose, and K. Raja. Multi-project scheduling using an heuristic and a genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, 31(3–4):360–366, 2006.

[117] I. S. Kurtulus. Multiproject scheduling: Analysis of scheduling strategies under unequal delay penalties. *Journal of Operations Management*, 5:291–307, 1985.

[118] I. S. Kurtulus and E. W. Davis. Multi-project scheduling: Categorization of heuristic rules performance. *Management Science*, 28:161–172, 1982.

[119] J. Kuster and D. Jannach. Handling airport ground processes based on resource-constrained project scheduling. *Lecture Notes in Computer Science*, 4031:166–176, 2006.

[120] S. R. Lawrence and T. E. Morton. Resource-constrained multi-project scheduling with tardy costs: Comparing myopic, bottleneck, and resource pricing heuristics. *European Journal of Operational Research*, 64(2):168–187, 1993.

[121] H. Li and K. Womer. Modeling the supply chain configuration problem with resource constraints. *International Journal of Project Management*, 26 (6):646–654, 2008.

[122] L. L. Lorenzoni, H. Ahonen, and A. G. de Alvarenga. A multi-mode resource-constrained scheduling problem in the context of port operations. *Computers & Industrial Engineering*, 50:55–65, 2006.

[123] A. Lova, C. Maroto, and P. Tormos. A multicriteria heuristic method to improve resource allocation in multiproject scheduling. *European Journal of Operational Research*, 127(2):408–424, 2000.

[124] V. Maniezzo and A. Mingozzi. The project scheduling problem with irregular starting time costs. *Operations Research Letters*, 25(4):175–182, 1999.

[125] M. Mika, G. Waligóra, and J. Węglarz. Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models. *European Journal of Operational Research*, 164 (3):639–668, 2005.

[126] M. Mika, G. Waligóra, and J. Węglarz. Modelling setup times in project scheduling. In Józefowska and Węglarz [95], pages 131–165.

[127] M. Mika, G. Waligóra, and J. W eglarz. Tabu search for multi-mode resource-constrained project scheduling with schedule-dependent setup times. *European Journal of Operational Research*, 187 (3):1238–1250, 2008.

[128] R. H. Möhring, A. S. Schulz, F. Stork, and M. Uetz. On project scheduling with irregular starting time costs. *Operations Research Letters*, 28(4):149–154, 2001.

[129] R. H. Möhring, A. S. Schulz, F. Stork, and M. Uetz. Solving project scheduling problems by minimum cut computations. *Management Science*, 49:330–350, 2003.

[130] J. Nabrzynski and J. Węglarz. Knowledge-based multiobjective project scheduling problems. In Węglarz [193], pages 383–411.

[131] A. A. Najafi and S. T. A. Niaki. A genetic algorithm for resource investment problem with discounted cash flows. *Applied Mathematics and Computation*, 183(2):1057–1070, 2006.

[132] T. Nazareth, S. Verma, S. Bhattacharya, and A. Bagchi. The multiple resource constrained project scheduling problem: A breadth-first approach. *European Journal of Operational Research*, 112(2):347–366, 1999.

[133] K. Neumann. *Stochastic project networks: Temporal analysis, scheduling and cost minimization*. Number 344 in Lecture Notes in Economics and Mathematical Systems. Springer, Berlin, Germany, 1990.

[134] K. Neumann and C. Schwindt. Project scheduling with inventory constraints. *Mathematical Methods of Operations Research*, 56:513–533, 2002.

[135] K. Neumann and J. Zimmermann. Exact and truncated branch-and-bound procedures for resource-constrained project scheduling with discounted cash flows and general temporal constraints. *Central European Journal of Operations Research*, 10: 357–380, 2002.

[136] K. Neumann and J. Zimmermann. Methods for resource-constrained project scheduling problem with regular and nonregular objective functions and schedule-dependent time windows. In Węglarz [193], pages 261–288.

[137] K. Neumann and J. Zimmermann. Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints. *European Journal of Operational Research*, 127(2):425–443, 2000.

[138] K. Neumann, C. Schwindt, and J. Zimmermann. Recent results on resource-constrained project scheduling with time windows: Models, solution methods, and applications. *Central European Journal of Operations Research*, 10:113–148, 2002.

[139] K. Neumann, C. Schwindt, and N. Trautmann. Scheduling of continuous and discontinuous material flows with intermediate storage restrictions. *European Journal of Operational Research*, 165: 495–509, 2005.

[140] K. Neumann, C. Schwindt, and J. Zimmermann. Resource-constrained project scheduling with time windows: Recent developments and new applications. In Józefowska and Węglarz [95], pages 375–407.

[141] K. Nonobe and T. Ibaraki. Formulation and tabu search algorithm for the resource constrained project scheduling problem. In C. C. Ribeiro and P. Hansen, editors, *Essays and Surveys in Metaheuristics*, pages 557–588. Kluwer Academic Publishers, 2002.

[142] K. Nonobe and T. Ibaraki. A metaheuristic approach to the resource constrained project scheduling with variable activity durations and convex cost functions. In Józefowska and Węglarz [95], pages 225–248.

[143] H. Nübel. The resource renting problem subject to temporal constraints. *OR Spektrum*, 23:359–381, 2001.

[144] N. Nudtasomboon and S. U. Randhawa. Resource-constrained project scheduling with renewable and non-renewable resources and time-resource trade-offs. *Computers & Industrial Engineering*, 32(1): 227–242, 1997.

[145] L. Özdamar. A genetic algorithm approach to a general category project scheduling problem. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 29:44–59, 1999.

[146] L. Özdamar and G. Ulusoy. A survey on the resource-constrained project scheduling problem. *IIE Transactions*, 27:574–586, 1995.

[147] L. Özdamar, G. Ulusoy, and M. Bayyigit. A heuristic treatment of tardiness and net present value criteria in resource constrained project scheduling. *International Journal of Physical Distribution & Logistics Management*, 28:805–824, 1998.

[148] R. Padman and D. Zhu. Knowledge integration using problem spaces: A study in resource-constrained project scheduling. *Journal of Scheduling*, 9(2):133–152, 2006.

[149] E. Pesch. Lower bounds in different problem classes of project schedules with resource constraints. In Węglarz [193], pages 53–76.

[150] A. A. B. Pritsker, L. J. Watters, and P. M. Wolfe. Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science*, 16:93–107, 1969.

[151] M. Ranjbar and F. Kianfar. Solving the discrete time/resource trade-off problem in project scheduling with genetic algorithms. *Applied Mathematics and Computation*, 191:451–456, 2007.

[152] M. Ranjbar, F. Kianfar, and S. Shadrokh. Solving the resource availability cost problem in project scheduling by path relinking and genetic algorithm. *Applied Mathematics and Computation*, 196:879–888, 2008.

[153] M. Ranjbar, B. de Reyck, and F. Kianfar. A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling. *European Journal of Operational Research*, 193:35–48, 2009.

[154] W. O. Rom, O. I. Tukel, and J. R. Muscatello. MRP in a job shop environment using a resource constrained project scheduling model. *Omega*, 30(4): 275–286, 2002.

[155] J. L. Rummel, Z. Walter, R. Dewan, and A. Seidmann. Activity consolidation to improve responsiveness. *European Journal of Operational Research*, 161(3):683–703, 2005.

[156] M. Sabzehparvar and S. M. Seyed-Hosseini. A mathematical model for the multi-mode resource-constrained project scheduling problem with mode dependent time lags. *Journal of Supercomputing*, 44(3):257–273, 2008.

[157] H. Sakkout and M. Wallace. Probe backtrack search for minimal perturbation in dynamic scheduling. *Constraints*, 5(4):359–388, 2000.

[158] F. Salewski, A. Schirmer, and A. Drexl. Project scheduling under resource and mode identity constraints: Model, complexity, methods, and application. *European Journal of Operational Research*, 102:88–110, 1997.

[159] A. Schirmer and A. Drexl. Allocation of partially renewable resources — Concepts, capabilities, and applications. *Networks*, 37:21–34, 2001.

[160] F. Schultmann and O. Rentz. Environment-oriented project scheduling for the dismantling of buildings. *OR Spectrum*, 23(1):51–78, 2001.

[161] C. Schwindt. Generation of resource-constrained project scheduling problems subject to temporal constraints. Technical Report WIOR-543, Universität Karlsruhe, Germany, 1998.

[162] C. Schwindt and N. Trautmann. Batch scheduling in process industries: An application of resource-constrained project scheduling. *OR Spectrum*, 22 (4):501–524, 2000.

[163] S. Shadrokh and F. Kianfar. A genetic algorithm for resource investment project scheduling problem, tardiness permitted with penalty. *European Journal of Operational Research*, 181(1):86–101, 2007.

[164] A. Shtub, L. J. LeBlanc, and Z. Cai. Scheduling programs with repetitive projects: A comparison of a simulated annealing, a genetic and a pair-wise swap algorithm. *European Journal of Operational Research*, 88:124–138, 1996.

[165] R. Słowiński. Multiobjective network scheduling with efficient use of renewable and nonrenewable resources. *European Journal of Operational Research*, 7:265–273, 1981.

[166] R. Słowiński, B. Soniewicki, and J. Węglarz. DSS for multiobjective project scheduling. *European Journal of Operational Research*, 79(2):220–229, 1994.

[167] D. E. Smith-Daniels, R. Padman, and V. L. Smith-Daniels. Heuristic scheduling of capital constrained projects. *Journal of Operations Management*, 14(3):241–254, 1996.

[168] A. Sprecher. *Resource-constrained project scheduling: Exact methods for the multi-mode case*. Number 409 in Lecture Notes in Economics and Mathematical Systems. Springer, Berlin, Germany, 1994.

[169] H. Stadtler. Multilevel capacitated lot-sizing and resource-constrained project scheduling: An integrating perspective. *International Journal of Production Research*, 43:5253–5270, 2005.

[170] C. S. Sung and S. K. Lim. A project activity scheduling problem with net present value measure. *International Journal of Production Economics*, 37(2-3):177–187, 1994.

[171] H. R. Tareghian and S. H. Taheri. A solution procedure for the discrete time, cost and quality tradeoff problem using electromagnetic scatter search. *Applied Mathematics and Computation*, 190(2):1136–1145, 2007.

[172] L. V. Tavares. A review of the contribution of operational research to project management. *European Journal of Operational Research*, 136(1):1–18, 2002.

[173] V. Tiwari, J. H. Patterson, and V. A. Mabert. Scheduling projects with heterogeneous resources to meet time and quality objectives. *European Journal of Operational Research*, 193(3):780–790, 2009.

[174] O. I. Tukel and S. N. Wasti. Analysis of supplier buyer relationships using resource constrained project scheduling strategies. *European Journal of Operational Research*, 129(2):271–276, 2001.

[175] G. Ulusoy and S. Cebelli. An equitable approach to the payment scheduling problem in project management. *European Journal of Operational Research*, 127(2):262–278, 2000.

[176] G. Ulusoy, F. Sivrikaya-Şerifoğlu, and Şule Şahin. Four payment models for the multi-mode resource constrained project scheduling problem with discounted cash flows. *Annals of Operations Research*, 102(1–4):237–261, 2001.

[177] S. Van de Vonder, E. L. Demeulemeester, and W. S. Herroelen. A classification of predictive-reactive project scheduling procedures. *Journal of Scheduling*, 10(3):195–207, 2007.

[178] M. Vanhoucke. Setup times and fast tracking in resource-constrained project scheduling. *Computers & Industrial Engineering*, 54(4):1062–1070, 2008.

[179] M. Vanhoucke. Scheduling an R&D project with quality-dependent time slots. *Lecture Notes in Computer Science*, 3982:621–630, 2006.

[180] M. Vanhoucke. Work continuity constraints in project scheduling. *Journal of Construction Engineering and Management*, 132:14–25, 2006.

[181] M. Vanhoucke, E. L. Demeulemeester, and W. S. Herroelen. On maximizing the net present value of a project under renewable resource constraints. *Management Science*, 47:1113–1121, 2001.

[182] M. Vanhoucke, E. L. Demeulemeester, and W. S. Herroelen. An exact procedure for the resource-constrained weighted earliness-tardiness project scheduling problem. *Annals of Operations Research*, 102:179–196, 2001.

[183] M. Vanhoucke, E. L. Demeulemeester, and W. S. Herroelen. Maximizing the net present value of a project with linear time-dependent cash flows. *International Journal of Production Research*, 39: 3159–3181, 2001.

[184] M. Vanhoucke, E. L. Demeulemeester, and W. S. Herroelen. Discrete time/cost trade-offs in project scheduling with time-switch constraints. *Journal of the Operational Research Society*, 53:741–751, 2002.

[185] M. Vanhoucke, E. L. Demeulemeester, and W. S. Herroelen. Progress payments in project scheduling problems. *European Journal of Operational Research*, 148(3):604–620, 2003.

[186] M. Vanhoucke, J. Coelho, D. Debels, B. Maenhout, and L. V. Tavares. An evaluation of the adequacy of project network generators with systematically sampled networks. *European Journal of Operational Research*, 187:511–524, 2008.

[187] V. A. Varma, R. Uzsoy, J. Pekny, and G. Blau. Lagrangian heuristics for scheduling new product development projects in the pharmaceutical industry. *Journal of Heuristics*, 13(5):403–433, 2007.

[188] A. Viana and J. P. de Sousa. Using metaheuristics in multiobjective resource constrained project scheduling. *European Journal of Operational Research*, 120(2):359–374, 2000.

[189] S. Voß and A. Witt. Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-world application. *International Journal of Production Economics*, 105(2):445–458, 2007.

[190] G. Waligóra. Discrete-continuous project scheduling with discounted cash flows – a tabu search approach. *Computers & Operations Research*, 35(7): 2141–2153, 2008.

[191] J. Węglarz. On certain models of resource allocation problems. *Kybernetics*, 9:61–66, 1981.

[192] J. Węglarz. Project scheduling with continuously divisible, doubly-constrained resources. *Management Science*, 27:1040–1057, 1981.

[193] J. Węglarz, editor. *Project scheduling: Recent models, algorithms and applications*. Kluwer Academic Publishers, 1999.

[194] J. Węglarz, J. Blazewicz, W. Cellary, and R. Słowiński. Algorithm 520: An automatic revised simplex method for constrained resource network scheduling. *ACM Transactions on Mathematical Software*, 3:295–300, 1977.

[195] D. S. Yamashita, V. A. Armentano, and M. Laguna. Robust optimization models for project scheduling with resource availability cost. *Journal of Scheduling*, 10(1):67–76, 2007.

[196] H. H. Yang and Y. L. Chen. Finding the critical path in an activity network with time-switch constraints. *European Journal of Operational Research*, 120: 603–613, 2000.

[197] G. Zhu, J. F. Bard, and G. Yu. Disruption management for resource-constrained project scheduling. *Journal of the Operational Research Society*, 56: 365–381, 2005.

[198] G. Zhu, J. F. Bard, and G. Yu. A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem. *INFORMS Journal on Computing*, 18:377–390, 2006.