

Salewski, Frank; Schirmer, Andreas; Drexl, Andreas

Working Paper — Digitized Version

Project Scheduling under Resource and Mode Identity Constraints. Part I: Model, Complexity Status, and Methods

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 387

Provided in Cooperation with:

Christian-Albrechts-University of Kiel, Institute of Business Administration

Suggested Citation: Salewski, Frank; Schirmer, Andreas; Drexl, Andreas (1996) : Project Scheduling under Resource and Mode Identity Constraints. Part I: Model, Complexity Status, and Methods, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 387, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/181066>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

No. 387

**Project Scheduling under Resource
and Mode Identity Constraints.**

**Part I: Model, Complexity Status,
and Methods**

Salewski / Schirmer / Drexl

January 1996

Frank Salewski, Andreas Schirmer, Andreas Drexl, Institut für Betriebswirtschaftslehre,
Christian-Albrechts-Universität zu Kiel, Olshausenstr. 40, D-24118 Kiel, Germany

Phone, Fax +49-431-880-15 31

E-mail salewski@bwl.uni-kiel.de, schirmer@bwl.uni-kiel.de, drexl@bwl.uni-kiel.de

Contents

1. Introduction.....	1
2. Problem Setting	2
2.1 Background and Motivation	2
2.2 Model Formulation	4
2.3 Classification of Precedence and Temporal Relations	7
2.4 Modeling Capabilities and Complexity Status	9
3. Applications of Mode Identity Constraints.....	12
4. Algorithmic Scheme	13
4.1 Mode Selection (Stage 1)	14
4.2 Job Scheduling (Stage 2)	16
5. Summary and Preview	19
References.....	20

Abstract: A recurring problem in project management involves the allocation of scarce resources to the individual jobs comprising the project. In many situations such as audit-staff scheduling, timetabling, and course scheduling, the resources correspond to individuals (skilled labour). This naturally leads to an assignment-type project scheduling problem, i.e. a project has to be performed by assigning one or more of several individuals (resources) to each job. In this paper we consider the nonpreemptive variant of a resource-constrained project scheduling problem with mode identity. Mode identity refers to a generalization of the multi-mode case where the set of all jobs is partitioned into disjoint subsets while all jobs forming one subset have to be processed in the same mode. Both time and cost incurred by processing a subset of jobs depend on the resources assigned to it. This problem is a substantial and non-trivial generalization of the well-known multi-mode case. Regarding precedence and temporal relations as well as release dates and deadlines, the question arises to which jobs resources should be assigned in order to minimize overall costs. For solving this time-resource-cost-tradeoff problem we present a tailored *parallel randomized solution approach* into which both static and dynamic priority rules can be incorporated.

Keywords: PROJECT MANAGEMENT / SCHEDULING; RESOURCE CONSTRAINTS; PRECEDENCE / TEMPORAL CONSTRAINTS; MODE IDENTITY CONSTRAINTS; REGRET-BASED BIASED RANDOM SAMPLING

1. Introduction

A recurring problem in project management involves the allocation of scarce resources to the individual jobs comprising the project. In many situations the resources correspond to individuals (skilled labour). In this paper we consider a project consisting of J jobs to be scheduled over a planning horizon T subject to certain precedence and temporal constraints. Each job may be assigned to one or more of R renewable and N nonrenewable resources. Furthermore, the set of all jobs is partitioned into U disjoint subsets, and all jobs forming one subset have to be performed by the same resources. Both time and cost incurred by processing a subset of jobs depend on the resources assigned to it. Our objective is to find a least expensive schedule under which the project is completed by the given planning horizon. This model is highly suitable to audit-staff scheduling (Bolenz and Frank 1977, Chan and Dodin 1986, Drexl 1990, Dodin and Chan 1991, Salewski 1995), timetabling (de Werra 1985, Abramson 1991), course scheduling (Drexl, Juretzka, and Salewski 1993), and other assignment type project scheduling problems.

Traditional resource-constrained project scheduling approaches (Davis and Patterson 1975, Talbot and Patterson 1978, Stinson, Davis, and Khumawala 1978, Demeulemeester and Herroelen 1992) have been restricted to the case where each job may be performed in only one predefined way. More recently efforts have been made to formulate and solve the more general preemptive project scheduling problem where job durations are functions of consumed resources (Blazewicz, Cellary, Slowinski, and Weglarz 1986). Meanwhile efforts have been documented (Elmaghraby 1977, Talbot 1982, Patterson, Slowinski, Talbot and Weglarz 1989 and 1990, Drexl 1991, Sprecher 1994, Kolisch 1995) which address the formulation and solution of a variety of nonpreemptive project scheduling problems where job durations are discrete functions of job performance modes (Multi-Mode Resource-Constrained Project

Scheduling Problem / MRCPSP). However, the case of *mode identity* where the set of all jobs is partitioned into a number of disjoint subsets where all jobs belonging to the same subset have to be processed in the same mode has not been investigated up to now (Mode Identity Resource-Constrained Project Scheduling Problem / MIRCPSP).

The paper is organized as follows. In Section 2 we introduce the MIRCPS P and investigate where it is positioned in the context of project scheduling. In Section 3 several possible applications of mode identity constraints are outlined. In Section 4 we present the algorithmic scheme of a randomized heuristic solution approach which has been designed specifically for the MIRCPS P. Finally, Section 5 provides a brief summary of Part I and a preview of Part II of this paper, where a problem from the field of audit-staff scheduling is formulated in terms of the MIRCPS P.

2. Problem Setting

As indicated by the name, the Mode Identity Resource-Constrained Project Scheduling Problem (MIRCPS P) belongs to the field of resource-constrained project scheduling, to which it introduces a new mode concept that is more general than currently known mode concepts.

In Section 2.1, the underlying ideas of the concept of mode identity are explained, while a mathematical formulation of the model is presented in Section 2.2. To account for the wide range of precedence and temporal constraints covered by the model, we propose in Section 2.3 a general classification scheme allowing to differentiate between different types of precedence and temporal constraints. Finally, in Section 2.4 we show that the MIRCPS P is a generalization of the MRCPS P and that the feasibility (optimization) variant of the MIRCPS P is NP-complete (NP-hard).

2.1 Background and Motivation

In the well-known multi-mode case of project scheduling all mode-job-assignments are mutually independent in the sense that assigning a mode to one job j of a project consisting of a set J of nonpreemptable jobs does not necessarily force any other job to be processed in a specific mode.

In order to illustrate this, consider the following example. Let denote MM_j ($1 \leq j \leq J$) the set of modes in which job j may be performed, and mm_j the specific mode in which it is actually processed. (Note that we introduce mm_j for the sake of simplicity; we could also employ the binary variable x_{jmt} defined below. Also, the example disregards any resource or temporal constraints.) In addition, let be given mode-job-assignments as shown in Table 1 where each job j has to be processed in a specific mode mm_j from a set of modes MM_j , independently of all other mode-job-assignments.

Table 1*Exemplary Mode-Job-Assignments*

j	MM_j	mm_j
1	{1, 2, 3, 4}	3
2	{1, 2, 3, 4}	2
3	{1, 2, 3}	2
4	{1, 2, 3, 4}	2
5	{1, 2, 3, 4}	4
6	{1, 2, 3}	3
7	{1}	1

Imagine, however, a situation in which certain jobs belong together in the sense that they must be executed in the same way. Consider e.g. the jobs 1, 2, and 5 on one hand as well as the jobs 3 and 6 on the other hand, which were assigned originally to different modes (cp. Table 1). Now, all jobs belonging to the same subset must be processed in the same mode. In order to formulate this situation in a scheduling model we partition the set of all jobs J into U disjoint subsets of jobs H_u ($1 \leq u \leq U$) as shown in Table 2. Now, let MI_u denote the set of modes in which the jobs of H_u may be performed; w.l.o.g. all jobs $j \in H_u$ are assumed to have the same number of modes. Finally, let mi_u denote the specific mode in which all jobs $j \in H_u$ are actually processed. Then the so-called mode identity - explicitly formulated in constraints (4) below - guarantees that all jobs belonging to the same subset H_u are processed in the same mode mi_u from a set of modes MI_u .

Table 2*Partitioning the Set of all Jobs in the Mode Identity Case*

u	H_u	MI_u	mi_u	mm_j
1	{1, 2, 5}	{1, 2, 3, 4}	3	$mm_1 = mm_2 = mm_5 = 3$
2	{3, 6}	{1, 2, 3}	1	$mm_3 = mm_6 = 1$
3	{4}	{1, 2, 3, 4}	2	$mm_4 = 2$
4	{7}	{1}	1	$mm_7 = 1$

2.2 Model Formulation

In the following we give a formal description of the MIRCPS. The problem parameters are summarized (in alphabetical order) in Table 3. Note that - in addition to the "standard" parameters of project scheduling models - the parameters H_u ($1 \leq u \leq U$) and M_u ($1 \leq u \leq U$) will be used.

Table 3

Problem Parameters of the MIRCPS

Problem Parameter	Definition
c_{um}	Cost of processing subset u in mode m
d_{jm}	Time required to perform job j in mode m
δ_j	Deadline of job j
H_u	Specific nonempty subset of jobs
J	Number of jobs, indexed by j
k_{jmn}^v	Total consumption of nonrenewable resource n required to execute job j in mode m
k_{jmr}^p	Per-period usage of renewable resource r required to execute job j in mode m
K_n^v	Total availability of nonrenewable resource n
K_{rt}^p	Availability of renewable resource r in period t
λ_j	Release date of job j
M_u	Number of modes of subset u , indexed by m
N	Number of nonrenewable resources, indexed by n
$q_{j'jm'm}$	Mode-dependent time-lags (finish-to-start) for $j' \in V_j$, if j' (j) is processed in mode m' (m).
R	Number of renewable resources, indexed by r
T	Number of periods (planning horizon), indexed by t
U	Number of disjoint subsets of jobs, indexed by u
V_j	Set of all immediate predecessors of job j

W.l.o.g. the parameters N and R as well as all c_{um} , d_{jm} , δ_j , k_{jmn}^v , k_{jmr}^p , K_n^v , K_{rt}^p , and λ_j are assumed to be nonnegative integers, while J , M_u , T , and U are assumed to be positive integers. All parameters $q_{j'jm'm}$ may take arbitrary, i.e. positive, zero, or negative, integer values. These restrictions entail no loss of generality since they are equivalent to allowing

rational numbers, i.e. fractions, and multiplying them with the smallest common multiple of their denominators.

In addition, we assume for all jobs j and j' ($1 \leq j \leq J$; $1 \leq j' \leq J$) that $j' \in V_j$ implies $j' \neq j$. This means, loosely speaking, that no job can be a *direct* (immediate) predecessor of itself. Note, however, that this assumption does not exclude the case where $j' \in V_j$ and $j \in V_{j'}$ simultaneously hold, such that a job may be its own *indirect* predecessor. An example of this relationship is given in Part II of this paper.

In order to reduce the number of binary variables needed in the following formulation of the MIRCSP, we compute from the time-lags $q_{j'm}$, the release dates λ_j , and the deadlines δ_j for each job j the earliest start time ES_j and the latest finish time LF_j . The job j must be completed within the interval $[ES_j + d_{jm}, LF_j]$ or at least one of the temporal constraints will be violated. Further, let denote f_u the job with the smallest index of subset H_u , i.e. $f_u = \min \{j \mid j \in H_u\}$.

Table 4 summarizes the derived parameters of the MIRCSP (in alphabetical order).

Table 4
Derived Parameters of the MIRCSP

Derived Parameter	Definition
ES_j	Earliest starting time of activity j
f_u	Job with the smallest index of subset H_u , $f_u = \min \{j \mid j \in H_u\}$
LF_j	Latest finishing time of activity j

Defining variables

$$x_{jmt} = \begin{cases} 1, & \text{if job } j \text{ is performed in mode } m \text{ and completed in period } t, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

allows to formulate a binary program - using the general framework given in Pritsker, Waters, and Wolfe (1969) - as follows:

$$\text{Minimize } Z(x) = \sum_{u=1}^U \sum_{m=1}^{M_u} c_{um} \sum_{t=ES_{f_u}+d_{f_{um}}}^{LF_{f_u}} x_{f_{um}t} \quad (2)$$

subject to

$$\sum_{m=1}^{M_u} \sum_{t=ES_{f_u}+d_{f_{um}}}^{LF_{f_u}} x_{f_{um}t} = 1 \quad (1 \leq u \leq U) \quad (3)$$

$$\sum_{t=ES_{f_u}+d_{f_{um}}}^{LF_{f_u}} x_{f_{um}t} = \sum_{t=ES_j+d_{jm}}^{LF_j} x_{jmt} \quad (1 \leq u \leq U; \forall j \in H_u \setminus \{f_u\}; 1 \leq m \leq M_u) \quad (4)$$

$$\sum_{t=ES_{j'}+d_{j'm}}^{LF_{j'}} (t+q_{j'jm'm}) x_{j'mt} \leq \sum_{t=ES_j+d_{jm}}^{LF_j} (t-d_{jm}) x_{jmt} \quad (1 \leq u' \leq U; \forall j' \in H_{u'}; 1 \leq u \leq U; \forall j \in H_u; j' \in V_j; 1 \leq m' \leq M_{u'}; 1 \leq m \leq M_u) \quad (5)$$

$$\sum_{u=1}^U \sum_{m=1}^{M_u} \sum_{j \in H_u} k_{jmr}^{\rho} \sum_{q=t}^{t+d_{jm}-1} x_{jmq} \leq K_{rt}^{\rho} \quad (1 \leq r \leq R; 1 \leq t \leq T) \quad (6)$$

$$q \in \{ES_j+d_{jm}, \dots, LF_j\}$$

$$\sum_{u=1}^U \sum_{m=1}^{M_u} \sum_{j \in H_u} k_{jmn}^{\nu} \sum_{t=ES_j+d_{jm}}^{LF_j} x_{jmt} \leq K_n^{\nu} \quad (1 \leq n \leq N) \quad (7)$$

$$x_{jmt} \in \{0, 1\} \quad (1 \leq u \leq U; \forall j \in H_u; 1 \leq m \leq M_u; ES_j+d_{jm} \leq t \leq LF_j) \quad (8)$$

In (2), x represents the vector of all binary variables; $Z(x)$ is the objective function value for any feasible vector x with respect to (3) - (8). By virtue of the mode identity constraints (4), only the job with the smallest index of each subset of jobs needs to be regarded for maintaining the corresponding mode.

The job completion constraints (3) stipulate that the job with the smallest index of each subset of jobs is completed exactly once in one of its modes. The mode identity constraints (4) guarantee for each subset of jobs that if the job with the smallest index is completed then all the other jobs will be completed as well, and in the same mode as said job. Thus, (3) and (4) combine to ensure that all jobs of each subset of jobs will be processed in the same mode, and that all of them will be completed.

The temporal constraints (5) enforce respectation of the time-lags - and thus the precedence relations - between the jobs. The temporal constraints will be dealt with in more detail in the following section.

The capacity constraints (6) and (7) represent per-period and total availability restrictions of renewable and nonrenewable resources, respectively.

2.3 Classification of Precedence and Temporal Relations

Note from Section 2.2 that the formulation of the temporal constraints (5) is general enough to cover not only the standard type of time-lags, namely minimum time-lags, but also the less common maximum time-lags, since these can easily be converted into minimum time-lags using the transformation introduced in Bartusch, Möhring, and Radermacher 1988. It is quite obvious that the presence of maximum lags tends to render scheduling problems even more difficult. In fact, to our knowledge there are only very few results so far on the performance of solution procedures for this kind of problems (cf. Brinkmann, Neumann 1994; Neumann, Zhan 1995).

In order to establish a terminology as a starting point for the analysis of such problems, we propose three criteria which allow to distinguish between different cases of precedence and temporal relations. These criteria are *connectivity*, *cyclicity*, and *domain*. In particular, the first two criteria pertain to both precedence and temporal relations, while the last applies to temporal relations only.

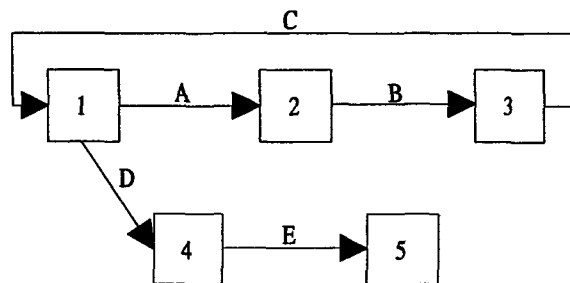
For notational brevity in what follows, let \angle denote the precedence relation induced by all the time-lags $q_j'j_m'm$, i.e. $\angle = \{(j', j) \mid 1 \leq j \leq J \wedge 1 \leq j' \leq J \wedge j' \in V_j\}$. Accordingly, we are free to write simply $j' \angle j$ whenever $j' \in V_j$ holds. From the assumptions in Section 2.2 it is clear that the relation \angle is irreflexive, asymmetric, and transitive. Let denote j and j' ($1 \leq j \leq J$; $1 \leq j' \leq J$) jobs belonging to subsets H_u and $H_{u'}$ ($1 \leq u \leq U$; $1 \leq u' \leq U$) while m and m' ($1 \leq m \leq M_u$; $1 \leq m' \leq M_{u'}$) will denote arbitrary modes in which j and j' may be processed. Also, let $J = \{1, \dots, J\}$. Finally, all maximum time-lags are assumed to have been transformed into minimum ones. Note that in the definitions to come the more restricted case is always given first.

- Definition 1** (Connectivity) A precedence relation is called
- isolating iff only jobs belonging to the same subset of jobs H_u are related, i.e. for all jobs $j \in H_u, j' \in H_{u'}, j \angle j'$ implies $u = u'$,
 - connecting otherwise, i.e. iff at least two jobs from different subsets of jobs H_u are related, i.e. there exist jobs $j \in H_u, j' \in H_{u'}, j \angle j'$ such that $u \neq u'$. ■

- Definition 2** (Cyclicity) A precedence relation is called
- acyclic iff it forms no cycle, i.e. there exists no subset of jobs $\{j_1, \dots, j_n\} \subseteq J$, where $j_1 \prec j_2, j_2 \prec j_3, \dots, j_{n-1} \prec j_n$, and $j_1 = j_n$.
- cyclic otherwise, i.e. iff it forms at least one cycle, i.e. there exists a subset of jobs $\{j_1, \dots, j_n\} \subseteq J$, where $j_1 \prec j_2, j_2 \prec j_3, \dots, j_{n-1} \prec j_n$, and $j_1 = j_n$. ■
- Definition 3** (Domain) A temporal relation is called
- non-negative iff each time-lag has a non-negative value, i.e. for all jobs j' ($1 \leq j' \leq J$) and j ($1 \leq j \leq J$) with $j' \prec j$ the inequality $q_{j'jm'm} \geq 0$ holds,
- arbitrarily-valued otherwise, i.e. iff at least one time-lag has a negative value, i.e. there exist jobs j' ($1 \leq j' \leq J$) and j ($1 \leq j \leq J$) with $j' \prec j$ where $q_{j'jm'm} < 0$. ■

These definitions are best illustrated in terms of an example. This is provided by Figure 1, where each arc (denoted by a letter) from a job j' to another job j (denoted by integer numbers, respectively) is understood to represent a time-lag $q_{j'jm'm}$. The set of all jobs is assumed to be divided into two subsets $H_1 = \{1, 2, 3\}$ and $H_2 = \{4, 5\}$.

Figure 1
Some Exemplary Time-Lags



Consider now the arcs A, B, C, and E. Loosely speaking, i.e. identifying each arc with the time-lag it represents, we may say that these four arcs taken together induce an isolating precedence relation. Still, adding the arc D renders the induced relation a connecting one. The arcs A, B, and C taken together form a cycle, such that the induced relation qualifies as being cyclic, whereas removing either one of them would render the relation acyclic. Finally, the reader may note that in the case of cyclic temporal relations the length of the longest path formed by the arcs must be nonpositive; otherwise there would be no feasible solution since not all of the temporal relations could possibly be fulfilled. This implies that given a cycle at least one of the contributing time-lags must be allowed to be arbitrarily-valued. If the time-

lags corresponding to A and B are 2 and 3, respectively, then the time-lag corresponding to C must not be greater than -5 minus the total duration of jobs 1, 2, and 3 in order to allow for a feasible schedule (negative time-lags may be interpreted as the result of transforming positive maximum lags into negative minimum ones). We will take up again these classification criteria in Section 4 as well as in Part II of this paper.

2.4 Modeling Capabilities and Complexity Status

This section deals with two theoretical aspects of the MIRCPS as a whole. First, Theorem 1 allows to position the MIRCPS in the field of project scheduling.

Theorem 1 The MIRCPS is a generalization of the MRCPS.

Proof: Consider the MIRCPS where $|H_u| = 1$ for all u . In this case the MIRCPS is equivalent to the MRCPS with $J = U$. ■

To illustrate this insight, let us reconsider the mode-job-assignments from Table 1. The corresponding transformations are shown in Table 5 where by setting $H_u = \{u\} = \{j\}$ all job-mode-assignments of the MRCPS can be represented in terms of the MIRCPS.

Table 5

Mode Identity Is a Generalization of the Multi-Mode Case

Multi-Mode Case			Mode Identity Case			
j	MM_j	mm_j	u	H_u	MI_u	mi_u
1	{1, 2, 3, 4}	3	1	{1}	{1, 2, 3, 4}	3
2	{1, 2, 3, 4}	2	2	{2}	{1, 2, 3, 4}	2
3	{1, 2, 3}	2	3	{3}	{1, 2, 3}	2
4	{1, 2, 3, 4}	2	4	{4}	{1, 2, 3, 4}	2
5	{1, 2, 3, 4}	4	5	{5}	{1, 2, 3, 4}	4
6	{1, 2, 3}	3	6	{6}	{1, 2, 3}	3
7	{1}	1	7	{7}	{1}	1

Second, we show the MIRCPS to belong to the class of strongly NP-hard problems by proving the corresponding feasibility problem $MIRCPS^{feas}$ to be strongly NP-complete. (We assume the reader to be familiar with the issues of complexity theory. For introductory texts on this subject cf. Garey and Johnson 1979, Papadimitriou and Steiglitz 1982, Papadimitriou 1994.)

Note that in order to do so, we have to assume for each instance that T is polynomially boundable in the instance length. On the formal level, this assumption is necessary since otherwise the MIRCSP is only solvable in exponential time. On the practical level, though, this assumption does not restrict the practical relevance of the MIRCSP since if the planning horizon is part of the instance it will usually not become arbitrarily large, and thus be polynomially boundable. (For a more detailed discussion cf. Schirmer 1996a and 1996b.) The proof relies upon the *3-partition problem* (3PP) which is well-known to be strongly NP-complete (Garey and Johnson 1975):

(3PP) Given $L \in \mathbb{N}$, $S = \{1, \dots, 3 \cdot L\}$, $B \in \mathbb{N}$, and $\mu_j \in \mathbb{N}$ ($1 \leq j \leq 3 \cdot L$) such that

$$B/4 < \mu_j < B/2 \quad (1 \leq j \leq 3 \cdot L) \quad (9)$$

$$\text{and } \sum_{j \in S} \mu_j = L \cdot B \quad (10)$$

hold, is there a partition of S into L disjoint subsets S_i ($1 \leq i \leq L$) such that

$$\sum_{j \in S_i} \mu_j = B \quad (1 \leq i \leq L) \quad (11)$$

holds true?

Note that the conditions on the μ_j ($1 \leq j \leq 3 \cdot L$) imply that $|S_i| = 3$ ($1 \leq i \leq L$). Obviously, $\text{MAX}_{\text{STD}}(3\text{PP}) = B$ and $\text{LNG}_{\text{STD}}(3\text{PP}) \approx O(L \cdot \log B)$ hold.

Theorem 2 The $\text{MIRCSP}^{\text{feas}}$ is strongly NP-complete.

Proof:

- (i) ($\text{MIRCSP}^{\text{feas}} \in \text{NP}$) Letting M denote $\max \{M_u \mid 1 \leq u \leq U\}$, the length of an instance I of $\text{MIRCSP}^{\text{feas}}$ is $\text{LNG}_{\text{STD}}(I) \approx O(J^2 \cdot M^2 \cdot N \cdot R \cdot T \cdot U \cdot \log \text{MAX}_{\text{STD}}(I))$. A certificate C of $\text{MIRCSP}^{\text{feas}}$ consists of one value for each of the $J \cdot M \cdot T$ decision variables; due to their binarity $\text{MAX}_{\text{STD}}(C) = 1$ and thus $\text{LNG}_{\text{STD}}(C) = J \cdot M \cdot T \cdot (\log(1) + 1) = J \cdot M \cdot T$. Therefore, reading a certificate has a time complexity of $O(J \cdot M \cdot T)$ which is polynomial in the input length. Evaluating the constraints will require $O(J \cdot M \cdot N \cdot R \cdot T^2 \cdot U)$ additions and multiplications; assuming that each operation takes constant time, the time complexity of the total evaluation is also $O(J \cdot M \cdot N \cdot R \cdot T^2 \cdot U)$, which is polynomial as well. So, any certificate can be read and verified in polynomial time, hence $\text{MIRCSP}^{\text{feas}}$ is element of NP.
- (ii) ($3\text{PP} \propto_p \text{MIRCSP}^{\text{feas}}$) Let $L \in \mathbb{N}$, $S = \{1, \dots, 3 \cdot L\}$, $B \in \mathbb{N}$, $\mu_j \in \mathbb{N}$ ($1 \leq j \leq 3 \cdot L$) constitute an arbitrary instance of 3PP. Then construct an instance of $\text{MIRCSP}^{\text{feas}}$ as follows: $T = L$, $J = U = 3 \cdot L$, $N = 0$, $H_u = \{u\}$, $M_u = 1$, $d_{j1} = 1$ ($1 \leq j \leq J$), $R = 1$, $k_j = \mu_j$

($1 \leq j \leq J$), $K = B$, $\delta_j = 0$, $\lambda_j = T$, and V_j ($1 \leq j \leq J$) empty, where due to $R = 1$ and $N = 0$ k_j stands for k_{j11}^P , and K for the K_{1t}^P ($1 \leq t \leq T$) which are assumed to be all equal (cf. Figure 2).

Figure 2

3PP Transforms to MIRCSPfeas

$K = B$	$k_1 = \mu_1$	\vdots	\vdots	\vdots	\vdots	
	$k_2 = \mu_2$	\vdots	$k_j = \mu_j$	\vdots	$k_{J-1} = \mu_{3L-1}$	$K/4 < k_j < K/2$
	\vdots	\vdots	\vdots	\vdots	$k_J = \mu_{3L}$	$B/4 < \mu_j < B/2$
	$\underbrace{\hspace{2em}}$	\dots	$\underbrace{\hspace{2em}}$	\dots	$\underbrace{\hspace{2em}}$	
	S_1		S_i		S_L	

In order to show that this transformation is indeed pseudo-polynomial, let us first argue that there exists a feasible schedule for the constructed instance of $MIRCSPfeas$ iff there is a partition of S as described above. Assuming that there exists such a partition of S , we can design a feasible schedule by scheduling all activities in S_i ($1 \leq i \leq L$) to period i . As the S_i form a partition of S , this procedure covers all activities, hence constraints (3) are met. Due to $M_u = 1$ ($1 \leq u \leq U$) there are no mode identity constraints (4), due to \angle empty no precedence constraints (5), due to $N = 0$ no nonrenewable resource constraints (7). Finally, (11) translates to

$$\sum_{j \in S_i} k_j = K \quad (12)$$

such that the schedule also satisfies the resource constraints (6). Conversely, if such a feasible schedule exists, a partition of S as described above can be obtained by putting together in S_i ($1 \leq i \leq L$) all elements j ($1 \leq j \leq 3 \cdot L$) scheduled to period i . Due to constraints (3) this yields a partition, and due to constraints (6) also the desired property (11) holds.

Second, the above transformation can be performed in pseudo-polynomial, even in polynomial time since all $O(L \cdot T)$ assignments can be done in $O(\log B)$ time. As all values of the 3PP-instance form part of the derived instance, the third condition, viz. that the transformed instance will not be of categorially smaller length, will be met even with the identical polynomial. Finally, the maximum number occurring in the constructed

instance, B , is also the maximum number in the 3PP-instance, such that it can be bounded in length and magnitude of the 3PP-instance by the polynomial $p(x,y) = 0 \cdot x + y$. So, the above transformation from 3PP to $\text{MIRCPSP}^{\text{feas}}$ is indeed pseudo-polynomial. ■

Corollary 1 The MIRCPSP is strongly NP-hard.

Actually, one can even show a stronger result, namely that the MIRCPSP is strongly NP-equivalent. The general idea of such proofs is to exhibit pseudo-polynomial reductions from $\text{MIRCPSP}^{\text{feas}}$ to MIRCPSP and vice versa. Details of appropriate simplified proof techniques are described in Garey and Johnson (1979) and Schirmer (1995).

3. Applications of Mode Identity Constraints

In this section we outline how mode identity constraints can be employed, giving examples from the fields of audit-staff scheduling, timetabling, and production planning.

A common problem in audit-staff scheduling involves assigning teams of auditors to audit engagements and scheduling the individual jobs of each engagement. In doing so, one has to guarantee that the assignment of a team to an engagement, once made, is never changed, i.e. that all jobs of an engagement are processed by the same team of auditors. Interpreting a specific team composition as one mode of processing and representing each auditor by one renewable resource, mode identity constraints are appropriate for modeling this kind of restrictions (for more details on this problem cf. Part II of this paper).

A similar problem arises in the field of timetabling where assignments between teachers, classes, rooms, lessons, and periods have to be constructed. Assuming that an assignment of teachers to lessons has already been made, the next step may consist of splitting the lessons into blocks and scheduling them. For example, a lesson of five hours length could be split into one block of two and one block of three hours or, alternatively, into one block of one hour and two blocks of two hours each. If both possibilities are feasible, one has to ensure that the scheduled blocks of that lesson combine to a total duration of five hours. Letting each mode represent one alternative splitting of a lesson into blocks, this can be done easily using mode identity constraints. Obviously, similar considerations apply to course scheduling problems.

Finally, we point out an application to production planning. Consider items which are manufactured or assembled in a number of production activities (each such activity corresponding to one job). To avoid unduly setup costs - incurred e.g. by transportation or shipment - it may be desirable to combine those jobs required for producing one specific item into sets of jobs which all have to be performed in the same facility (each facility in which a job may be executed corresponding to one of its modes). Once a mode has been assigned to one job, the mode identity constraints guarantee that all other jobs of the same set are performed in the

same mode, implying the corresponding item to be produced completely in one specific facility.

4. Algorithmic Scheme

This section discusses a solution approach, named *RAMSES* (Randomized Mode Selection and Scheduling), which has been developed specifically for the MIRCPS. Three reasons led to this development.

First, standard methods of integer programming seem to lend themselves to solving the MIRCPS. However, even for modestly sized problems the formulation (2) - (8) translates into very large integer programs which in turn result in prohibitive running times. In fact, the NP-hardness of the model allows to conjecture that - except for the most simple cases - no exact scheduling algorithm will generally produce optimal solutions in a reasonable amount of time.

Second, apart from exact methods, a multitude of deterministic heuristics for approximately solving scheduling problems have been proposed (Davis and Patterson 1975, Kurtulus and Davis 1982, Talbot 1982, Kurtulus and Narula 1985, Ulusoy and Özdamar 1989, Alvarez-Valdés and Tamarit 1989a and 1989b, Boctor 1990, Valls, Perez, and Quintanilla 1992, Kolisch 1995). They all use priority values to resolve conflicts between jobs competing for the allocation of scarce resources. Unfortunately, heuristics are not guaranteed to find an optimal schedule. In fact, heuristics may - especially in the presence of scarce resources - not even be able to find a feasible schedule (cp. Drexl and Grünewald 1993). Furthermore, deterministic heuristics yield only one solution for an instance, even if applied several times. Considering that this solution may be arbitrarily bad or even infeasible, determinism seems to be a major deficiency for heuristic methods.

Third, biased random sampling methods try to overcome the shortcoming of determinism by performing the selection process randomly, but according to probabilities which are proportional to priority values. In this way, in each scheduling step every job *may* be chosen, though those sharing higher priorities will have a higher probability of being selected. Due to their nondeterminism, repeated application of randomized heuristics will produce a set of solutions rather than one sole solution. Usually some of these solutions will be better than the one found with the deterministic version of the same method (cf. Cooper 1976; Kolisch 1995). Moreover, no tie-breaking rules need to be specified for randomized methods: since they operate randomly ties cannot occur. However, even if *RAMSES* is borrowing from some of the ideas incorporated in the randomized scheduling method *STOCOM* (Drexl and Grünewald 1993), common heuristic scheduling methods cannot be utilized directly since these would not take the mode identity constraints (4) into account.

RAMSES may be characterized as a parallel regret-based biased random sampling approach. As indicated by its name, its proceeding can be divided into two successive stages: It is performing the assignment of modes to subsets of jobs in the first stage while the actual scheduling of the jobs is done in the second stage. Since both stages use priority values, different priority rules can be used alternatively in each stage.

Recall from Section 2.2 that the model (2) - (8) is general enough to cover both types of time-lags, viz. minimum as well as maximum lags. Obviously, the presence of arbitrary maximum lags tends to render already nontrivial scheduling problems even more difficult. We have no results as yet on the performance of *RAMSES* in the presence of cyclic, arbitrary-valued, and connecting lags. In Part II of this paper we present, however, an application where restricting the temporal relation to the special case of isolating, even though cyclic and arbitrary-valued, lags allows to apply *RAMSES* very successfully.

4.1 Mode Selection (Stage 1)

For each subset of jobs one of its modes is selected. This is accomplished by first choosing a subset of jobs H_u to be considered, based upon priority values $\bar{\omega}_u$ ($1 \leq u \leq U$), and second selecting one of its modes, based upon priority values $\bar{\pi}_m$ ($1 \leq m \leq M_u$). To compute these values, a variety of different priority rules (in the following denoted by ω und π , respectively) can be imagined (e.g. for $\bar{\pi}_m$ the cost of processing a subset H_u of jobs in a certain mode).

The priority values are modified to obtain the actual selection priorities as

$$\bar{\bar{\omega}}_u = (\bar{\omega}_u + \varepsilon)^\alpha \quad (1 \leq u \leq U) \quad (13)$$

$$\bar{\bar{\pi}}_m = (\bar{\pi}_m + \varepsilon)^\beta \quad (1 \leq m \leq M_u) \quad (14)$$

where $\varepsilon \in \mathbb{R}_{>0}$ and $\alpha, \beta \in \mathbb{R}_{\geq 0}$. $\varepsilon > 0$ guarantees $\bar{\bar{\omega}}_u$ and $\bar{\bar{\pi}}_m$ to be nonzero and thus gives also a subset or mode with a priority value of zero a chance of being chosen; the control parameters α and β allow to diminish or enforce the differences between the rule-dependent values for $\alpha, \beta < 1$ or $\alpha, \beta > 1$, respectively.

The probability $\tilde{\omega}_u$ ($1 \leq u \leq U$) that subset H_u of jobs will be selected is given by

$$\tilde{\omega}_u = \bar{\bar{\omega}}_u / \left(\sum_{u'=1}^U \bar{\bar{\omega}}_{u'} \right) \quad (1 \leq u \leq U) \quad (15)$$

The probability $\tilde{\pi}_m$ ($1 \leq m \leq M_u$) that mode m of the considered subset H_u will be selected is

$$\tilde{\pi}_m = \bar{\pi}_m / \left(\sum_{m'=1}^{M_u} \bar{\pi}_{m'} \right) \quad (1 \leq m \leq M_u) \quad (16)$$

Formally, the first stage of *RAMSES* can now be described as follows:

Initialization

SET := {1,...,U};

Execution

```

while SET ≠ ∅
{
  for all u ∈ SET
  {
    compute  $\bar{\omega}_u$ ;
    compute  $\bar{\pi}_u$ ;
  }

  for all u ∈ SET
    compute  $\tilde{\omega}_u$ ;

  select u ∈ SET with probability proportional to  $\tilde{\omega}_u$ ;
  SET := SET \ {u};

  for all m ∈ {1,...,Mu}
  {
    compute  $\bar{\pi}_m$ ;
    compute  $\bar{\pi}_m$ ;
  }

  for all m ∈ {1,...,Mu}
    compute  $\tilde{\pi}_m$ ;

  select m ∈ {1,...,Mu} with probability proportional to  $\tilde{\pi}_m$ ;
  modeu := m;
}

```

Output

For each subset H_u of jobs, $mode_u$ denotes the mode in which the jobs of subset H_u are to be processed.

Note that in the presence of static priority rules ω and/or π some minor modifications can be applied in order to speed up computations. E.g., the priority values $\bar{\omega}_u$ ($1 \leq u \leq U$) need not

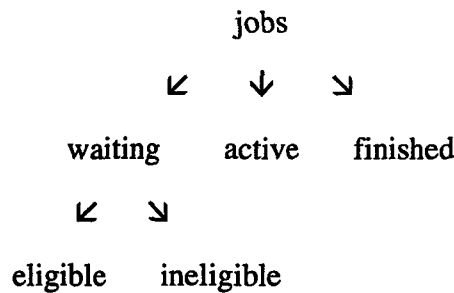
be determined if π is a static rule since then the order in which the subsets are considered for mode assignment is irrelevant.

4.2 Job Scheduling (Stage 2)

Heuristic construction methods generally classify jobs into four disjoint sets of states: *eligible*, *ineligible*, *active*, and *finished* (cp. e.g. Kurtulus and Narula 1985). A job that is neither active nor finished is termed *eligible* if it may be scheduled currently, *ineligible* otherwise. For the sake of computational efficiency, however, it is useful to introduce an additional state, viz. *waiting*. If a job is neither active nor finished, it is defined to be waiting. This means that the state waiting consists of eligible and ineligible jobs, simultaneously (cp. Figure 3).

Figure 3

Classification of Scheduling States



A schedule is built, following a parallel scheduling scheme (cp. Kelley 1963, Kolisch 1995), by proceeding chronologically over all periods of the planning horizon. In each period t the set EJ of eligible jobs, i.e. jobs which may be started without violating the temporal constraints (5), is determined from the set WJ of waiting jobs. From EJ one job is scheduled for starting execution in period t . In order to update EJ with respect to the set AJ of active jobs, EJ has to be recomputed after each job selection. This process of augmenting partial schedules is repeated as long as no restriction is violated. Then the next period is considered etc., until eventually the planning horizon is reached. The resulting schedule is feasible if all jobs are scheduled, and infeasible otherwise.

Let for each subset of jobs H_u ($1 \leq u \leq U$) denote m ($1 \leq m \leq M_u$) the mode in which H_u is to be processed. Let for each job $j \in H_u$ specify $finish_j$ the period in which the processing of j is finished. Let t denote the current period. Further, let denote FJ the set of finished jobs. Then the set WJ can be derived from:

$$WJ = \{1, \dots, J\} \setminus (FJ \cup AJ) \quad (17)$$

Next, let denote RC_{rt}^P ($1 \leq r \leq R$; $1 \leq t \leq T$) the remaining capacity of renewable resource r in period t and RC_n^V ($1 \leq n \leq N$) the total remaining capacity of nonrenewable resource n . Note that now instead of $[ES_j, LF_j]$ tighter time windows $[ES_{jm}, LF_{jm}]$ can be calculated by using $q_{j'jm'm}$, since the mode assignment has already been accomplished in the first stage. Using \angle as introduced in Section 2.3, EJ can be computed for each period considered as

$$\begin{aligned}
EJ &= \{ j \in WJ \mid ES_{jm+1} \leq t \leq LF_{jm} - d_{jm} + 1 \\
&\wedge (\forall j' \angle j) ((j' \in AJ \cup FJ \wedge finish_{j'} + q_{j'jm'm} + 1 \leq t) \vee (j' \notin AJ \cup FJ \wedge q_{j'jm'm} < 0)) \\
&\wedge (\forall j \angle j'') ((j'' \in AJ \cup FJ \wedge t \leq finish_{j''} - d_{j''m} - d_{jm} - q_{jj''mm} + 1) \vee (j'' \notin AJ \cup FJ)) \\
&\wedge (\forall r \in \{1, \dots, R\}) (\forall t' \in \{t, \dots, t + d_{jm} - 1\}) k_{jmr}^P \leq RC_{rt}^P \\
&\wedge (\forall n \in \{1, \dots, N\}) k_{jmn}^V \leq RC_n^V \} \quad (18)
\end{aligned}$$

The essence of (18) is that a job is eligible if the following conditions are met. First, scheduling the job to the currently considered period does not violate precedence constraints. Second, for each of its predecessors the following holds: If it is active or finished then the appropriate time-lags are respected, if it is waiting then future iterations will still be able to meet the appropriate time-lags. Third, for each of its successors the following holds: If it is active or finished then the appropriate time-lags are respected (The second condition will take care of the other case where the job is still waiting). Fourth, capacity limitations of both renewable and nonrenewable resources are respected. We should emphasize that the first condition is dominated by the second one, due to the fact that precedence relations are merely a special case of temporal ones. Although being redundant, it allows to speed up the process of determining EJ by identifying some of the ineligible jobs faster.

Starting from these definitions, the selection probabilities can easily be computed. Assuming priorities $\bar{\rho}_j$ ($j \in EJ$) to be calculated according to a rule (in the sequel denoted by ρ), modified priorities can be derived from them as

$$\bar{\bar{\rho}}_j = (\bar{\rho}_j + \varepsilon)^\gamma \quad (\forall j \in EJ) \quad (19)$$

where $\varepsilon \in \mathbf{R}_{\geq 0}$ and $\gamma \in \mathbf{R}_{\geq 0}$. The probability $\tilde{\rho}_j$ ($\forall j \in EJ$) that job j will be selected is

$$\tilde{\rho}_j = \bar{\bar{\rho}}_j / \left(\sum_{j' \in EJ} \bar{\bar{\rho}}_{j'} \right) \quad (\forall j \in EJ) \quad (20)$$

Now, the second stage of *RAMSES* may be described as follows:

Initialization

```

FJ := ∅;
AJ := ∅;
WJ := {1, ..., J};

```

```

for all j ∈ WJ
{
  compute ESjm;
  compute LFjm;
}

```

Execution

```

for t := 1 to T
{
  compute EJ;
  while (EJ ≠ ∅)
  {
    for all j ∈ EJ
    {
      compute  $\bar{\rho}_j$ ;
      compute  $\bar{\rho}_j$ ;
    }

    for all j ∈ EJ
      compute  $\tilde{\rho}_j$ ;

    select j ∈ EJ according to  $\tilde{\rho}_j$ ;
    finishj := t + djm - 1;
    AJ := AJ ∪ {j};
    WJ := WJ \ {j};
    compute EJ;
  }

  for all j ∈ AJ
  if (finishj = t)
  {
    FJ := FJ ∪ {j};
    AJ := AJ \ {j};
  }
}

```

Output

If $FJ = \{1, \dots, J\}$, then for each job *j* *finish*_{*j*} specifies the period in which *j* is finished; otherwise, the constructed schedule is infeasible.

Note that in the presence of static priority rules again several modifications can be used to speed up the processing of the algorithm.

To summarize, *RAMSES* is a two-stage heuristic solution approach for the MIRCPS. Utilizing randomized priority rules, its first stage constructs an assignment of modes to subsets of jobs, while its second stage is performing the actual scheduling of the jobs. The algorithmic scheme of *RAMSES* may be seen as a framework into which different priority rules can be incorporated.

We refrain, however, from proposing specific rules at this point and claiming their general applicability. Since the MIRCPS is a further generalization of the already general MRCPS, a variety of practical scheduling situations could be represented in terms of the MIRCPS. We believe that the performance of different rules and different values of the control parameters depends strongly on the structure of the problem setting at hand. Thus it seems unwise to evaluate them on test instances generated in a purely random way. Rather than that, we advocate assessing their performance on instances which reflect real-life scheduling situations.

5. Summary and Preview

In this paper, we introduced the MIRCPS, an assignment-type, resource-constrained project scheduling model which comprises many features being important with respect to project management in practice. Moreover, it extends the well-known multi-mode case to the so-called mode identity case. For dealing with this model, a regret-based biased random sampling approach, *RAMSES*, has been developed into which different priority rules can be incorporated.

In regard of the broad applicability of the MIRCPS we refrain from evaluating *RAMSES* on test instances generated in a purely random way. Rather than that, we advocate assessing its performance on instances which reflect specific real-life scheduling situations.

To illustrate this, Part II of this paper will present an application from the field of audit-staff scheduling. We will address several topics: We will demonstrate how to represent the problem setting in terms of the MIRCPS and how to adapt *RAMSES* adequately. Then, we will cover the design of a specifically tailored instance generator, the statistical model, the experimental design, and the performance measures used in an extensive experimental study serving to assess the performance of *RAMSES*. Finally, we will provide the results of the study.

References

- Abramson, D., "Constructing School Timetables using Simulated Annealing: Sequential and Parallel Algorithms", *Management Science*, 37 (1991), 98-113.
- Alvarez-Valdés, R. and J.M. Tamarit, "Heuristic Algorithms for Resource-Constrained Project Scheduling: A Review and an Empirical Analysis", in: Slowinski, R. and J. Weglarz (eds.): *Advances in Project Scheduling*, Elsevier, Amsterdam, 1989a, 113-134.
- Alvarez-Valdés, R. and J.M. Tamarit, "Algoritmos Heurísticos Deterministas y Aleatorios en Secuenciación de Proyectos con Recursos Limitados", *Qüestiió*, 13 (1989b), 173-191.
- Bartusch, M., R.H. Möhring, and F.J. Radermacher, "Scheduling Project Networks with Resource Constraints and Time Windows", *Annals of Operations Research*, 16 (1988), 201-240.
- Blazewicz, J., W. Cellary, R. Slowinski and J. Weglarz, *Scheduling under Resource Constraints - Deterministic Models*, Baltzer, Basel, 1986 (*Annals of Operations Research*, 7).
- Boctor, F.F., "Some Efficient Multi-Heuristic Procedures for Resource-Constrained Project Scheduling", *European Journal of Operational Research*, 49 (1990), 3-13.
- Bolenz, G. and R. Frank, "Das Zuordnungsproblem von Prüfern zu Prüffeldern unter Berücksichtigung von Reihenfolgebedingungen - Ein Lösungsansatz der binären Optimierung", *Zeitschrift für betriebswirtschaftliche Forschung*, 29 (1977), 427-447.
- Brinkmann, K. and K. Neumann, "Heuristic Procedures for Resource-Constrained Project Scheduling with Minimal and Maximal Time Lags: The Minimum Project-Duration and Resource-Levelling Problems", Report WIOR-443 (1994), Universität Karlsruhe.
- Chan, K.H. and B. Dodin, "A Decision Support System for Audit-Staff Scheduling with Precedence Constraints and Due Dates", *The Accounting Review*, 61 (1986), 726-733.
- Cooper, D.F., "Heuristics for Scheduling Resource-Constrained Projects: An Experimental Investigation", *Management Science*, 22 (1976), 1186-1194.
- Davis, E.W. and J.H. Patterson, "A Comparison of Heuristic and Optimum Solutions in Resource-Constrained Project Scheduling", *Management Science*, 21 (1975), 944-955.
- Demeulemeester, E. and W. Herroelen, "A Branch-and-Bound Procedure for the Multiple Resource-Constrained Project Scheduling Problem", *Management Science*, 38 (1992), 1803-1818.

- Dodin, B. and K.H. Chan, "Application of Production Scheduling Methods to External and Internal Audit Scheduling", *European Journal of Operational Research*, 52 (1991), 267 - 279.
- Drexl, A., *Planung des Ablaufs von Unternehmensprüfungen*, Poeschel, Stuttgart, 1990.
- Drexl, A., "Scheduling of Project Networks by Job Assignment", *Management Science*, 37 (1991), 1590-1602.
- Drexl, A. and J. Grünewald, "Nonpreemptive Multi-Mode Resource-Constrained Project Scheduling", *IIE Transactions*, 25 (1993), 74-81.
- Drexl, A, J. Juretzka, and F. Salewski, "Academic Course Scheduling under Workload and Changeover Constraints", *Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel* 337 (1993).
- Elmaghraby, S.E., *Activity Networks: Project Planning and Control by Network Models*, Wiley, New York, 1977.
- Garey, M.R. and D.S. Johnson, *Computers and Intractability - A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, 1979.
- Kelley, J.E., "The Critical-Path Method: Resources Planning and Scheduling", in: *Industrial Scheduling*, Muth, J.F. and G.L. Thompson (eds.), Prentice-Hall, Englewood Cliffs (N.J.), 1963, pp. 347-365.
- Kolisch, R., *Project Scheduling under Resource Constraints - Efficient Heuristics for Several Problem Classes*, Physica, Heidelberg, 1995.
- Kurtulus, I.S. and E.W. Davis, "Multi-Project Scheduling: Categorization of Heuristic Rules Performance", *Management Science*, 28 (1982), 161-172.
- Kurtulus, I.S. and S.C. Narula, "Multi-Project Scheduling: Analysis of Project Performance", *IIE Transactions*, 17 (1985), 58-66.
- Neumann, K. and J. Zhan, "Heuristics for the Minimum Project-Duration Problem with Minimal and Maximal Time Lags under Fixed Resource Constraints", *Journal of Intelligent Manufacturing* 6 (1995), 145-154.
- Papadimitriou, C.H., *Computational Complexity*, Addison-Wesley, Reading, Mass, 1994.
- Papadimitriou, C.H. and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs (N.J.), 1982.

- Patterson, J., R. Slowinski, B. Talbot and J. Weglarz, "An Algorithm for a General Class of Precedence and Resource Constrained Scheduling Problems", in: Slowinski, R. and J. Weglarz (eds.): *Advances in Project Scheduling*, Elsevier, Amsterdam, 1989, 3-28.
- Patterson, J., R. Slowinski, B. Talbot and J. Weglarz, "Computational Experience with a Backtracking Algorithm for Solving a General Class of Resource Constrained Scheduling Problems", *European Journal of Operational Research*, 49 (1990), 68-79.
- Pritsker, A.A.B., W.D. Watters and P.M. Wolfe, "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach", *Management Science*, 16 (1969), 93 - 108.
- Salewski, F., *Hierarchische Personaleinsatzplanung in Wirtschaftsprüfungsgesellschaften*, Physica, Heidelberg, 1995.
- Schirmer, A., "A Guide to Complexity Theory in Operations Research", *Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel*, 381 (1995).
- Schirmer, A., "New Insights on the Complexity of Resource-Constrained Project Scheduling - A Case of Single-Mode Scheduling", *Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel*, 390 (1996a).
- Schirmer, A., "New Insights on the Complexity of Resource-Constrained Project Scheduling - Two Cases of Multi-Mode Scheduling", *Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel*, 391 (1996b).
- Sprecher, A., *Resource-Constrained Project Scheduling - Exact Methods for the Multi-Mode Case*, Springer, Berlin, 1994.
- Stinson, J.P., E.W. Davis and B.M. Khumawala, "Multiple Resource-Constrained Scheduling Using Branch and Bound", *AIIE Transactions*, 10 (1978), 252-259.
- Talbot, R.B., "Resource-Constrained Project Scheduling with Time-Resource Tradeoffs: The Nonpreemptive Case", *Management Science*, 28 (1982), 1197-1210.
- Talbot, R.B. and J.H. Patterson, "An Efficient Integer Programming Algorithm with Network Cuts for Solving Resource-Constrained Scheduling Problems", *Management Science*, 24 (1978), 1163-1174.
- Ulusoy, G. and L. Özdamar, "Heuristic Performance and Network/Resource Characteristics in Resource-Constrained Project Scheduling", *Journal of the Operational Research Society*, 40 (1989), 1145-1152.

Valls, V., M.A. Perez, and M.S. Quintanilla, "Heuristic Performance in Large Resource-Constrained Projects", Working Paper, Departamento de Estadística e Investigación Operativa, Universitat de València (1992).

de Werra, D., "An Introduction to Timetabling", *European Journal of Operational Research*, 19 (1985), 151-162.