

Salewski, Frank; Schirmer, Andreas; Drexel, Andreas

Working Paper — Digitized Version

Project Scheduling under Resource and Mode Identity Constraints. Part II: An Application to Audit-Staff Scheduling

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 388

Provided in Cooperation with:

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

Suggested Citation: Salewski, Frank; Schirmer, Andreas; Drexel, Andreas (1996) : Project Scheduling under Resource and Mode Identity Constraints. Part II: An Application to Audit-Staff Scheduling, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 388, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/181067>

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. 388

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

**Part II: An Application to
Audit-Staff Scheduling**

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. Formulation of the Medium-Term Audit-Staff Scheduling Problem Using Mode Identity Constraints.....	2
3. Classification and Complexity Status.....	7
4. Algorithmic Scheme and Priority Rules	9
4.1 Priority Rules for Mode Assigning (Stage 1)	9
4.1.1 Engagement Selection.....	9
4.1.2 Mode Selection	10
4.2 Priority Rules for Subphase Scheduling (Stage 2).....	12
4.2.1 Precedence-Based Rule.....	12
4.2.2 Critical-Path-Based Rules.....	12
4.2.3 Resource-Based Rules	13
5. Experimental Performance Analysis	15
5.1 Generation of Test Instances	15
5.2 Statistical Model	17
5.3 Experimental Design	18
5.4 Performance Measures.....	19
6. Computational Results.....	20
7. Summary and Conclusions	26
References.....	27

Abstract: When scheduling its audit-staff, the management of an auditing firm encompasses a number of different decisions which differ markedly in terms of organizational echelon involved, length of the planning horizon and the planning periods, degree of aggregation of the audit tasks, degree of detail of the required information, and decision objective. We outline an hierarchical planning approach for the audit-staff scheduling problem and demonstrate how the first level can be expressed in terms of the MIRCPSP, which has been presented in Part I of this paper. For solving this problem, we also demonstrate how the solution approach *RAMSES*, also presented in Part I of this paper, can be adapted to it. We provide a number of different priority rules which can be employed alternatively. The results of an extensive computational study indicate that *RAMSES* is capable of producing "good" solutions in neglectable amounts of time.

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

1. Introduction

In Part I of this paper we presented a general time- and resource-constrained project scheduling problem with mode identity constraints, together with a general parallel randomized solution approach. In Part II we will show how to use these concepts for specific real-world scheduling problems by giving an example from the auditing environment.

When scheduling its audit-staff, the management of an auditing firm encompasses a number of different decisions. These decisions may be grouped into several categories which differ markedly in terms of organizational echelon involved, length of the planning horizon and the planning periods, degree of aggregation of the audit tasks, degree of detail of the required information, and decision objective.

Traditional audit-staff scheduling models (Balachandran and Zoltners 1981, Chan and Dodin 1986, Drexel 1990 and 1991, Dodin and Chan 1991) are single-level models which try to construct a direct assignment of auditors to tasks and periods. To facilitate algorithmic treatment, all these models are more or less gross simplifications of practical planning situations. In contrast, the above observations led us to conduct a survey among the 200 biggest certified public accountant (CPA) firms in Germany. Based upon its findings we formulated an hierarchical modelling framework (Salewski and Drexel 1993, Salewski 1995) comprising three levels, viz. a medium-term (tactical), a medium-to-short-term (tactical-operational), and a short-term (operational) level:

- The tactical planning assigns teams of auditors to the engagements. It constructs a schedule by determining the workload per auditor and week over a planning horizon of between three and twelve months.
- The tactical-operational planning disaggregates the results of the tactical level for one week and all auditors. The outcome is a schedule for each auditor that covers - on the basis of periods of four hours - all engagements in which he is involved in the considered week.

- The operational planning is based upon the results of the tactical-operational level for one week and one engagement. It assigns the auditors involved in the auditing of that engagement to the corresponding audit tasks and schedules these tasks. This is done for a period length of one hour.

In Part II of this paper we will focus on the first level of the approach. Section 2 briefly outlines the corresponding problem setting and provides a mathematical formalization (in terms of the MIRCPSP) while Section 3 establishes its classification and its complexity status. In Section 4 we demonstrate how *RAMSES* can be adapted to this specific problem and provide a number of alternative priority rules. Section 5 outlines the design of a problem specific instance generator and defines the statistical model, the experimental design, and the performance measures used in an extensive computational study whose results are covered in Section 6. Finally, Section 7 provides a brief summary of Part II of this paper, along with our conclusions.

2. Formulation of the Medium-Term Audit-Staff Scheduling Problem Using Mode Identity Constraints

The *Medium-Term Audit-Staff Scheduling Problem* (MASSP) may be characterized by the following assumptions:

- A firm employs one or more *auditors*, which have to audit one or more *engagements* within a given *planning horizon* of normally 13, 26, or 52 weeks.
- Each engagement is made up of one or more *phases*, e.g. preliminary, intermediate, and final audit, which in turn can be decomposed into one or more *subphases*. Some subphases may not be executed before the completion of certain other subphases (*predecessors*). More exact: the phases of each engagement as well as the subphases of each phase must be processed in a strictly linear order which implies that each subphase except of the first subphase of the first phase of each engagement possesses exactly one predecessor.
- Work on some phases may not commence before a specific *release time*, as well as it is to be completed by a certain *deadline*.
- The *availability* of some auditors may be restricted in certain periods, e.g. due to holidays or vacations.
- For some periods a client may want to confine the time during which the auditing takes place (*maximum processing time*), e.g. due to vacation periods or stock-taking activities.
- Often an engagement could be audited by several alternative audit teams (modes). Different team compositions will result in different auditor processing times. Usually some modes will be preferable to others: Factors influencing the suitability of an auditor for a specific engagement are e.g. qualification level, industry experience, familiarity with the clients business, and degree of difficulty of the audit tasks. The preferability of a

mode as a whole may e.g. be linked to the total processing time needed (Drexel 1990). Hence, a preference value will be assigned to each mode.

- Between subsequent subphases (which thus belong to the same engagement), mode-dependent *minimum and maximum time-lags* are given.

Table 1
Problem Parameters of the MASSP

Problem Parameter	Definition
A	Number of auditors
a	Specific auditor, $a = 1, \dots, A$
b_{um}	Preference value corresponding to the processing of engagement u in mode m
C_{at}	Capacity of auditor a in period t
D_{ut}	Maximum processing time of engagement u in period t
δ_{up}	Deadline of phase (u, p)
k_{upsma}	Time auditor a needs to process subphase (u, p, s) in mode m (capacity usage)
λ_{up}	Release time of phase (u, p)
M_u	Number of modes of engagement u
m	Specific mode, $m = 1, \dots, M_u$
P_u	Number of phases of engagement u
p	Specific phase, $p = 1, \dots, P_u$
$q_{up's'psm}$	Minimum (finish-to-start) time-lag between subsequent subphases (u, p', s') and (u, p, s) when executing u in mode m
$\hat{q}_{up's'psm}$	Maximum (finish-to-start) time-lag between subsequent subphases (u, p', s') and (u, p, s) when executing u in mode m
S_{up}	Number of subphases of phase (u, p) . W.l.o.g. each subphase has a duration of one period.
s	Specific subphase, $s = 1, \dots, S_{up}$
T	Number of periods
t	Specific period, $t = 1, \dots, T$
U	Number of engagements
u	Specific engagement, $u = 1, \dots, U$
V_{ups}	Set of all immediate predecessors of subphase (u, p, s)

The objective is to assign the overall best-suited teams to the engagements (*mode assignment*), and to determine when the individual subphases are to be executed (*subphase scheduling*).

The definitions and the notation used in the following are summarized (in alphabetical order) in Table 1. W.l.o.g. T, A, U, as well as all M_u , P_u , S_{up} , and δ_{up} are assumed to be positive integers, while all other problem parameters are assumed to be nonnegative integers.

To simplify the mathematical formulation of the problem, we perform some preliminary computations. First, the indices u , p , s are replaced by

$$j = g(u, p, s) = \sum_{u'=1}^{u-1} \sum_{p'=1}^{P_u} S_{u'p'} + \sum_{p'=1}^{p-1} S_{up'} + s \quad (1)$$

Thus, the parameters V_{ups} , $q_{up's'psm}$, $\hat{q}_{up's'psm}$, and k_{upsma} become V_j , $q_{jj'm}$, $\hat{q}_{jj'm}$, and k_{jma} .

Then, let denote $f_u = g(u, 1, 1)$ the first and $l_u = g(u, P_u, S_u P_u)$ the last subphase of each engagement u ($1 \leq u \leq U$).

Further, the maximum lags $\hat{q}_{jj'm}$ can be transformed into minimum lags $q_{jj'm}$ (with a corresponding update of V_j) (cf. Bartusch, Möhring, and Radermacher 1988), and from the time-lags, the release times, and the deadlines earliest and latest finish times EF_j and LF_j can be computed, respectively.

Table 2 summarizes the derived parameters of the MASSP (in alphabetical order).

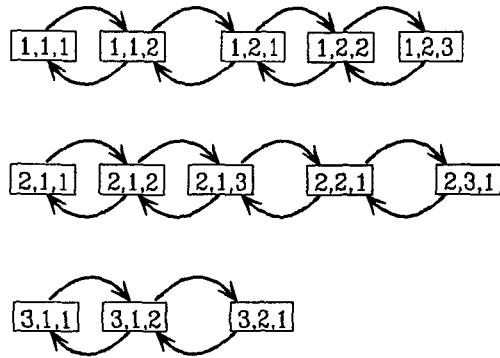
Table 2
Derived Parameters of the MASSP

Derived Parameter	Definition
EF_j	Earliest finishing time of activity j
f_u	First subphase of engagement u ($1 \leq u \leq U$)
l_u	Last subphase of engagement u ($1 \leq u \leq U$)
LF_j	Latest finishing time of activity j

After these transformations, let us consider for a moment the above assumptions. The presence of minimum and (transformed) maximum time-lags between subsequent subphases, along with their special sequence, which arises from the above mentioned decomposition process, imply for each engagement a "chain structure" of the time lags as illustrated in Figure 1, where

each entry has the format u,p,s (here we briefly return to the former notation to emphasize the engagement decomposition). Each structure may be seen as being composed of arc-disjoint cycles of length 2, one between first and second subphase, one between second and third one, and so forth up to the last cycle between last but one and last subphase. With respect to the classification scheme introduced in Part I, the temporal relation can be characterized as cyclic, arbitrarily-valued, and isolating.

Figure 1
"Chain Structure" of Time Lags



Now the assignment of subphases to modes and periods can be represented by binary variables x_{jmt} defined as

$$x_{jmt} = \begin{cases} 1, & \text{if subphase } j \text{ is performed in mode } m \text{ and completed in period } t \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

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

$$\text{Maximize } Z(x) = \sum_{u=1}^U \sum_{m=1}^{M_u} b_{um} \sum_{t=EF_{fu}}^{LF_{fu}} x_{fumt} \quad (3)$$

subject to

$$\sum_{m=1}^{M_u} \sum_{t=EF_{f_u}}^{LF_{f_u}} x_{f_u m t} = 1 \quad (1 \leq u \leq U) \quad (4)$$

$$\sum_{t=EF_{f_u}}^{LF_{f_u}} x_{f_u m t} = \sum_{t=EF_j}^{LF_j} x_{j m t} \quad (1 \leq u \leq U; f_u + 1 \leq j \leq l_u; 1 \leq m \leq M_u) \quad (5)$$

$$\sum_{m=1}^{M_u} \sum_{t=EF_{j'}}^{LF_{j'}} (t + q_{j'} j_m) x_{j' m t} \leq \sum_{m=1}^{M_u} \sum_{t=EF_j}^{LF_j} (t - 1) x_{j m t} \quad (1 \leq u \leq U; f_u \leq j \leq l_u; j' \in V_j) \quad (6)$$

$$\sum_{u=1}^U \sum_{m=1}^{M_u} \sum_{\substack{j=1 \\ t \in \{EF_j, \dots, LF_j\}}}^{l_u} k_{j m a} x_{j m t} \leq C_{a t} \quad (1 \leq a \leq A; 1 \leq t \leq T) \quad (7)$$

$$\sum_{m=1}^{M_u} \sum_{\substack{j=1 \\ t \in \{EF_j, \dots, LF_j\}}}^{l_u} k_{j m a} x_{j m t} \leq D_{u t} \quad (1 \leq u \leq U; 1 \leq a \leq A; 1 \leq t \leq T) \quad (8)$$

$$x_{j m t} \in \{0, 1\} \quad (1 \leq u \leq U; f_u \leq j \leq l_u; 1 \leq m \leq M_u; EF_j \leq t \leq LF_j) \quad (9)$$

The objective function (3) maximizes the total team preference over all engagements. Due to (4), it suffices to include only one subphase of each engagement in (3). The choice of the first one is arbitrary. The subphase completion constraints (4) stipulate that the first subphase of each engagement is completed exactly once in one of its modes. The mode identity constraints (5) guarantee for each engagement that if the first subphase is completed then the other subphases will be completed as well, and in the same mode as the first one. Thus, (4) and (5) combine to ensure that all subphases of each engagement will be processed in the same mode, and that all of them will be completed. The temporal constraints (6) represent the precedence order on the subphases and enforce respect of the time-lags between them. The auditor capacity constraints (7) assure that for no auditor his per-period workload exceeds his capacity. In this regard, each auditor is treated as a renewable resource. The engagement capacity constraints (8) guarantee that for no engagement and no auditor the maximum per-period processing time is exceeded. In this regard, each engagement is treated as a renewable resource.

3. Classification and Complexity Status

This section provides two important theoretical insights. First, a classification of the MASSP is given by the following theorem.

Theorem 1 The MASSP is a special case of the MIRCPSP. ■

Proof: The MIRCPSP can be restricted to the MASSP as shown in Table 3.

Table 3

Restriction of the MIRCPSP to the MASSP

Parameter of the MIRCPSP	Restricted to
c_{um}	$= \max \{b_{u'm'} \mid 1 \leq u' \leq U \wedge 1 \leq m' \leq M_{u'}\} - b_{um}$
d_{jm}	$= 1$
δ_j	$= \begin{cases} \delta_{up}, & \text{if } (\exists u \in \{1, \dots, U\}) (\exists p \in \{1, \dots, P_u\}) (j = g(u, p, S_{up})) \\ T, & \text{otherwise} \end{cases}$
H_u	$= \{f_u, \dots, l_u\}$
K_n^v	undefined, since $N = 0$
k_{jmn}^v	undefined, since $N = 0$
K_{rt}^p	$= \begin{cases} C_{rt}, & \text{if } r \leq A \\ D_{(r-1) \text{ div } A, t}, & \text{otherwise} \end{cases}$
k_{jmr}^p	$= \begin{cases} k_{jm(1+(r-1) \text{ mod } A)}, & \text{if } (r \leq A) \vee (j \in \{f_{(r-1) \text{ div } A}, \dots, l_{(r-1) \text{ div } A}\}) \\ 0, & \text{otherwise} \end{cases}$
λ_j	$= \begin{cases} \lambda_{up}, & \text{if } (\exists u \in \{1, \dots, U\}) (\exists p \in \{1, \dots, P_u\}) (j = g(u, p, 1)) \\ 0, & \text{otherwise} \end{cases}$
N	$= 0$
$q_{j'jm'm}$	$= \begin{cases} q_{j'jm}, & \text{if } m' = m \\ \text{undefined}, & \text{otherwise} \end{cases}$
R	$= (U + 1) A$

Noting that $EF_j - d_{jm}$ equals ES_j , the transformations of (3) - (6) and (9) into constraints of the MIRCPSP are mostly obvious; however, some additional explanations for the ones of (7) and (8) seem in place. As indicated above, each auditor a ($1 \leq a \leq A$) corresponds to a resource r

($1 \leq r \leq A$). However, due to the fact that (8) restricts the processing time for each auditor-engagement *combination*, a similar summation over all U engagements is not possible. Hence, we introduce artificial resources, treating each combination of an engagement and an auditor as a separate resource. Accordingly, each engagement u ($1 \leq u \leq U$) corresponds to resources r ($u \cdot A + 1 \leq r \leq (u+1)A$). Finally, note that for the MASSP the summation over $q = t, \dots, t+d_{jm}-1$ (cf. (6) in Part I of the paper) is redundant since $d_{jm} = 1$ ($f_u \leq j \leq l_u; 1 \leq m \leq M_u$).

Second, we show the MASSP to belong to the class of strongly NP-hard problems by proving the corresponding feasibility problem $\text{MASSP}^{\text{feas}}$ 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 MASSP is only solvable in exponential time. On the practical level, though, this assumption does not restrict the practical relevance of the MASSP as the results of our survey indicate that staff scheduling is always done for a fixed planning horizon (at most 52 weeks), and any fixed value of T is clearly polynomially boundable. (For a more detailed discussion cf. Schirmer 1996a and 1996b.)

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

Proof:

- (i) ($\text{MASSP}^{\text{feas}} \in \text{NP}$) Using M to abbreviate $\max \{M_u \mid 1 \leq u \leq U\}$, the length of an instance I of the $\text{MASSP}^{\text{feas}}$ is $\text{LNG}_{\text{STD}}(I) \approx O(A \cdot J \cdot M \cdot T \cdot \log \text{MAX}_{\text{STD}}(I))$. A certificate C of the $\text{MASSP}^{\text{feas}}$ 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 $O(A \cdot J \cdot M \cdot T)$ additions and multiplications; assuming that each operation takes constant time, the time complexity of the total evaluation is $O(A \cdot J \cdot M \cdot T)$ as well. So, any certificate can be read and verified in polynomial time, hence $\text{MASSP}^{\text{feas}}$ is element of NP .
- (ii) ($\text{MIRCPSP}^{\text{feas}} \in_p \text{MASSP}^{\text{feas}}$) Due to Theorem 2 (of Part I of this paper), $\text{MIRCPSP}^{\text{feas}}$ is strongly NP-complete. Due to Theorem 1, $\text{MIRCPSP}^{\text{feas}}$ can be pseudo-polynomially transformed to $\text{MASSP}^{\text{feas}}$. ■

Corollary 1 The MASSP is strongly NP-hard.

Actually, one can even show a stronger result, namely that the MASSP is strongly NP-equivalent. The general idea of such proofs is to exhibit pseudo-polynomial reductions from

MASSP^{feas} to MASSP and vice versa. Details of appropriate simplified proof techniques are described in Garey and Johnson (1979) and Schirmer (1995).

4. Algorithmic Scheme and Priority Rules

We now demonstrate how *RAMSES* can be adapted to the MASSP. Again, the first stage assigns one mode to each subphase while the scheduling of the subphases is done in the second stage. In fact, the algorithmic scheme described in Part I can be used without modifications.

We describe a number of priority rules for both stages which may be used alternatively. Together with the corresponding algorithmic scheme, each combination (ω, π, ρ) of priority rules defines one specific algorithm for the MASSP.

4.1 Priority Rules for Mode Assigning (Stage 1)

Since the mode assignment is done in two steps, i.e. selection of an engagement and selection of an appropriate mode, we introduce the priority rules following this sequence. Let denote EE the set of all eligible engagements, i.e. engagements to which no mode has been assigned yet.

4.1.1 Engagement Selection

Two rules will be described in the sequel.

- PRS. Pure Random Selection

This (static) rule accords to all engagements the same preference value, thus giving all of them the same probability of being selected:

$$\bar{\omega}_u = 1 \quad (u \in EE) \quad (10)$$

- MIN TRU MAX PRF. Minimum Total Resource Usage Maximum Preference

This (static) rule prefers those engagements where in the most preferred mode the demand for auditor capacities is small.

Let for each eligible engagement u denote m_u^{max} the most preferred mode, i.e. the mode m with the highest preference value b_{um} . Ties are broken by selecting the mode with the smaller total capacity demand, if possible; arbitrarily otherwise.

Let further denote AE_{um} the set of all auditors actually participating in the processing of engagement u , which can be derived from

$$AE_{um} = \{a \mid \sum_{j=f_u}^{l_u} k_{jma} > 0\} \quad (u \in EE; 1 \leq m \leq M_u) \quad (11)$$

Also, let denote TRU_{um} the total capacity usage incurred by processing engagement u in mode m , i.e.

$$TRU_{um} = \sum_{j=f_u}^{l_u} \sum_{a \in AE_{um}} k_{jma} \quad (u \in EE; 1 \leq m \leq M_u) \quad (12)$$

and TRU^{max} the maximum resource usage $TRU_{um_u^{max}}$ of all currently eligible engagements when processing them in their most preferred mode:

$$TRU^{max} = \max \{ TRU_{um_u^{max}} \mid u \in EE \} \quad (13)$$

Then the priority value can be computed as

$$\bar{\omega}_u = TRU^{max} - TRU_{um_u^{max}} \quad (u \in EE) \quad (14)$$

4.1.2 Mode Selection

Let in the following denote u the engagement under consideration, which has been selected before.

- MAX PRF. Maximum Preference

This (static) rule uses generalized opportunity costs, reflecting the effect of the considered mode assignment on the objective function value. For the engagement u under consideration, let denote b_u^{min} the preference value of the least preferred mode:

$$b_u^{min} = \min \{ b_{um} \mid 1 \leq m \leq M_u \} \quad (15)$$

In order to maximize the objective function (3), the priority values $\bar{\pi}_m$ are calculated as the maximum possible deterioration of the objective function value that could arise from *not* processing engagement u in mode m :

$$\bar{\pi}_m = b_{um} - b_u^{min} \quad (1 \leq m \leq M_u) \quad (16)$$

- MAX PRC. Maximum Preference Remaining Capacity

This (dynamic) rule tends - as well as MAX PRF - to select modes having a high preference value. Based upon the total capacity of an auditor a , computed as the sum of all his period

capacities C_{at} , the maximum feasible demand TCA_a for auditor a within the planning horizon is calculated by multiplying his total capacity by a "safety factor" SF:

$$TCA_a = SF \sum_{t=1}^T C_{at} \quad (1 \leq a \leq A) \quad (17)$$

Relying on the results of preliminary testing, SF was set to 0.95. Due to (17), only modes will be considered for selection where the remaining capacity RCA_a of each involved auditor a, computed as the difference between TCA_a and the capacity demands incurred by the mode assignments established so far, is not exceeded. (If no mode has a sufficient remaining capacity, an artificial mode is selected where all capacity demands equal zero. This reflects the practical situation where certain engagements have to be rejected due to insufficient resource availabilities.)

Let denote u' ($1 \leq u' \leq U$) an arbitrary engagement and m' the corresponding mode which has already been assigned. Then the remaining capacity RCA_a of auditor a can be derived from

$$RCA_a = TCA_a - \sum_{\substack{u'=1 \\ u' \notin EE}}^U \sum_{j=f_{u'}}^{l_{u'}} k_{jm'a} \quad (1 \leq a \leq A) \quad (18)$$

and the priority values are calculated as

$$\bar{\pi}_m = \begin{cases} b_{um} - b_u^{min}, & \text{if } (\forall a \in AE_{um}) RCA_a \geq \sum_{j=f_u}^{l_u} k_{jma} \\ -\epsilon, & \text{otherwise} \end{cases} \quad (1 \leq m \leq M_u) \quad (19)$$

If a mode m is to be excluded from being selected because of insufficient resource availabilities, we let $\bar{\pi}_m = -\epsilon$. In the case of $\beta \neq 0$ then $\bar{\pi}_m = 0$ and thus $\tilde{\pi}_m = 0$ hold, in the case of $\beta = 0$ we set $\bar{\pi}_m = 0$.

- MAX RCP. Maximum Remaining Capacity

This (dynamic) rule accords the highest priority values to those modes whose assigning leaves the highest average remaining capacity to the involved auditors. Again, only modes are considered where the remaining capacity RCA_a of no involved auditor a is exceeded. (In the case of insufficient remaining capacities the proceeding is as described above.)

Let denote RCE_m the average remaining capacity of the involved auditors after mode m has been assigned to engagement u :

$$RCE_m = \begin{cases} \left(\sum_{a \in AE_{um}} RCA_a - \sum_{j=f_u}^{l_u} \sum_{a \in AE_{um}} k_{jma} \right) / |AE_{um}|, & \text{if } |AE_{um}| > 0 \\ 0, & \text{otherwise} \end{cases} \quad (1 \leq m \leq M_u) \quad (20)$$

Now the priority values can be computed as

$$\bar{\pi}_m = \begin{cases} RCE_m, & \text{if } (\forall a \in AE_{um}) RCA_a \geq \sum_{j=f_u}^{l_u} k_{jma} \\ -\varepsilon, & \text{otherwise} \end{cases} \quad (1 \leq m \leq M_u) \quad (21)$$

In the case of $\beta = 0$ and $\bar{\pi}_m = -\varepsilon$, we set $\bar{\pi}_m = 0$ (as done for MAX PRC, above).

4.2 Priority Rules for Subphase Scheduling (Stage 2)

For the subphase scheduling, we distinguish between *precedence-based*, *critical-path-based*, and *resource-based* priority rules. Let in the sequel for each engagement u denote m the mode assigned to u and EJ the set of all eligible subphases, i.e. those subphases which could be scheduled in the current period without violating any of the restrictions.

4.2.1 Precedence-Based Rule

Rules of this type determine priority values according to criteria which are related to the precedence relations between the subphases. Only one rule of this type is used here.

- MTS. Most Total Successors

This (static) rule (cf. Alvarez-Valdés and Tamarit 1989) ranks the subphases according to their number of successors. The idea is to schedule first the subphases on whom the most successors are waiting, because delaying such a subphase would also delay all its successors.

Let for each subphase j denote S_j the set of all its - direct as well as indirect - successors. Then the priorities are determined from

$$\bar{\rho}_j = |S_j| \quad (j \in EJ) \quad (22)$$

4.2.2 Critical-Path-Based Rules

These (static) rules (cf. Alvarez-Valdés and Tamarit 1989) refer to the results of the critical-path analysis (cf. e.g. Levy, Thompson, and Wiest 1963).

- MIN EFT. Minimum Earliest Finishing Time

This rule prefers subphases with small earliest finishing times, in order to get them "out of the way" early. Now, instead of EF_j and LF_j the more exact values EF_{jm} and LF_{jm} can be used since the mode assignment has already taken place in the first stage (cp. Section 4.2 of Part I). In order to assign high priorities to subphases with small values of EF_{jm} , the priorities are computed as

$$\bar{p}_j = T - EF_{jm} \quad (j \in EJ) \quad (23)$$

- MIN LFT. Minimum Latest Finishing Time

The latest finishing time of a subphase yields a measure of its urgency, because its processing must be completed by this date. Thus, MIN LFT tries to schedule urgent subphases first.

$$\bar{p}_j = T - LF_{jm} \quad (j \in EJ) \quad (24)$$

- MIN SLK. Minimum Slack

The slack of a subphase is the interval between its earliest and its latest finishing time. This rule prefers subphases with a small slack since these are most critical with respect to delays.

$$\bar{p}_j = T - (LF_{jm} - EF_{jm}) \quad (j \in EJ) \quad (25)$$

4.2.3 Resource-Based Rules

These rules prioritize the eligible subphases j in terms of either their resource usage, the scarcity of the resources they use (e.g. the resource "auditor"), or a combination of both.

Let for each j denote AJ_j the set of all auditors involved in processing j in mode m , i.e.

$$AJ_j = \{a \mid 1 \leq a \leq A \wedge k_{jma} > 0\} \quad (j \in EJ) \quad (26)$$

Further, let denote RUJ_j the total capacity usage incurred by the processing of j in mode m , i.e.

$$RUJ_j = \sum_{a \in AJ_j} k_{jma} \quad (j \in EJ) \quad (27)$$

and RUJ_{max} the maximum of the resource usages RUJ_j of all currently eligible subphases:

$$RUJ_{max} = \max \{RUJ_j \mid j \in EJ\} \quad (28)$$

Finally let for each eligible subphase j denote $RC_{j|}$ the total remaining capacity over all assigned auditors

$$RC_{j|} = \sum_{a \in AJ_j} RC_{at} \quad (j \in EJ) \quad (29)$$

where RC_{at} denotes the remaining capacity of auditor a in period t , considering the mode assignments already made in stage 1.

- **MAX TRU. Maximum Total Resource Usage**

This (static) rule (cp. the rule GRES in Kurtulus, Davis 1982 and Kurtulus, Narula 1985, as well as GRD in Davis, Patterson 1975 for the case of nonrenewable resources) prefers those subphases whose processing will use large amounts of the auditor capacities.

$$\bar{\rho}_j = RU_{j|} \quad (j \in EJ) \quad (30)$$

- **MIN TRU. Minimum Total Resource Usage**

This (static) rule prefers subphases whose processing will use little of the auditor capacities.

$$\bar{\rho}_j = RU_{j|}^{max} - RU_{j|} \quad (j \in EJ) \quad (31)$$

- **MAX TRC. Maximum Total Remaining Capacity**

This (dynamic) rule quantifies the importance of subphases in terms of the scarcity of the resources used. The highest priority values are assigned to those subphases where the involved auditors have the largest remaining capacity in the considered period.

$$\bar{\rho}_j = RC_{j|} \quad (j \in EJ) \quad (32)$$

- **MIN TRC. Minimum Total Remaining Capacity**

This (dynamic) rule is based upon the same idea as MAX TRC. However, the highest priorities are assigned to the subphases with the smallest total remaining capacities, i.e.

$$\bar{\rho}_j = B - RC_{j|} \quad (j \in EJ) \quad (33)$$

where $B \in \mathbb{R}_{\geq 0}$ is large enough to guarantee that $\bar{\rho}_j$ is nonnegative.

- MAX RRU. Maximum Relative Resource Usage

The (dynamic) rule MAX RRU compares the resource usage of a given subphase with the highest resource usage RUJ^{max} of all currently eligible subphases. The priority values accorded are the higher, the higher the relative usages are.

$$\bar{p}_j = RUJ_j / RUJ^{max} \quad (j \in EJ) \quad (34)$$

- MIN RRU. Minimum Relative Resource Usage

The (dynamic) rule MIN RRU is based upon the same comparison as MAX RRU. However, the priority values accorded are the higher, the lower the relative usages are.

$$\bar{p}_j = 1 - (RUJ_j / RUJ^{max}) \quad (j \in EJ) \quad (35)$$

5. Experimental Performance Analysis

In the following subsections we outline the procedure followed for generating a sample of test instances of practical relevance. We also describe the statistical model and the design of an extensive experimental study conducted, along with the definitions of the performance measures used to evaluate the results of the study.

5.1 Generation of Test Instances

Even in current literature, the systematic generation of test instances does not receive much attention. For the well-researched field of project scheduling, Kolisch, Sprecher, and Drexl (1996) report that "very little research concerned with the systematic generation of benchmark instances has been published. [...] most efforts are only briefly described."

Generally, two possible approaches can be found adopted in literature when having to come up with test instances. First, practical cases. Their strength is their high practical relevance while the obvious drawback is the absence of any systematic structure allowing to infer any general properties. Even the 110 instances of the widely known Patterson-set (Patterson 1984), which have become a quasi benchmark in project scheduling, have been collated from different sources, rather than being generated from a systematic design. Thus, even if an algorithm performs good on some practice cases, it is not guaranteed that it will continue to do so on other instances as well. Second, artificial instances. Since they are generated randomly according to predefined specifications, their plus lies in the fact that fitting them to certain requirements such as given probability distributions poses no problems. A detailed such procedure for generating project scheduling instances has been proposed by Kolisch, Sprecher, Drexl (1996). However, they may reflect situations with little or no resemblance to any

problem setting of practical interest. Hence, an algorithm performing well on several such artificial instances may or may not perform satisfactorily in practice.

Therefore, we decided to devise a combination of both approaches, thereby attempting to keep the strengths of both approaches while avoiding their drawbacks. For a start, we conducted a survey among the 200 biggest CPA firms in Germany, asking the respective official in charge of staff planning (if existent) or one of the firm's partners to provide details about length of planning horizon, number of auditors, number and structure of audit engagements, auditor working capacities (working hours per day or week) and possible variations therein (e.g. due to vacations, training), etc. In addition, we carried out interviews with several experts in the field of auditing to clarify our understanding of the peculiarities of the auditing sector. Then, to ensure a systematic and consistent generation of the instances, for each of the parameters of the MASSP a domain and a discrete distribution function on the domain were defined, based upon the survey and the interview results. From these definitions, a test bed of representative instances was generated randomly, using a classification scheme to build instances with specific properties. In this way we tried to construct instances reflecting the specifics of audit-staff scheduling in the industry as closely as possible, yet to employ a systematic design for the generation procedure.

Clearly, the performance of an algorithm cannot be evaluated from running it on infeasible instances. It is therefore noteworthy that, in spite of the strong **NP**-completeness of the associated feasibility problem, it was possible to rig up the design of the (complicated) generation procedure in a way guaranteeing that for each constructed instance there exists at least one provably feasible solution.

We assumed that only two instance-related factors do have a major influence on the performance of a solution method, viz. the size and the tractability of the instance attempted. Although the *size* of an instance is determined by the length of the planning horizon, the number of subphases, *and* the number of modes, statistical analyses of the survey results found all these to depend on the length of the planning horizon. In the sequel, three types of instances will be distinguished with respect to their size: the planning horizon equals 13 weeks for *small* instances (with up to 30 auditors, 95 engagements, and 98,800 binary variables), 26 weeks for *medium-size* instances (with up to 55 auditors, 280 engagements, and 728,000 binary variables), and 52 weeks for *large* instances (with up to 125 auditors, 880 engagements, and 5,948,800 binary variables). In addition, *very small* instances (with up to 6 auditors, 10 engagements, and 10,400 binary variables) were generated. While these instances are too small to bear practical relevance, they can be solved to optimality with standard MIP-solvers as LINDO and OSL and thus can be used as benchmarks.

The *tractability* of an instance intends to reflect how easy or how difficult that particular instance is to solve. In our study, the auditor capacities are assumed to be the only factor in-

fluencing the tractability of an instance: the higher the auditor capacities are, the easier the corresponding instance is *ceteris paribus* to solve since its solution space becomes larger. Accordingly, the auditor capacities are calculated from the average expected demand, adjusted by a multiplicative factor RS (resource strength). Throughout this work, three types of instances will be distinguished with respect to their tractability: *easy* instances where RS is taken equal to 3.5, *medium* instances where RS equals 2.5, and *hard* instances where RS is 1.5.

Detailed definitions for the remaining parameters as well as the feasibility proofs can be obtained from the authors.

5.2 Statistical Model

For the purposes of this study, the execution of *RAMSES* is regarded as a random experiment, the outcome of which is determined by the following factors:

- ω priority rule employed for engagement selection
- π priority rule employed for mode selection
- ρ priority rule employed for subphase scheduling
- α control parameter for engagement selection
- β control parameter for mode selection
- γ control parameter for subphase scheduling
- σ size of instances attempted
- τ tractability of instances attempted
- ι number of iterations performed

Specifying a set of values for each factor describes over which levels it is varied during an experiment, while one value for each factor determines one run of an experiment.

Definition 1: An experiment is a tuple $(\Omega, \Pi, P, A, B, \Gamma, \Sigma, T, H)$, where

- Ω is a set of priority rules for engagement selection
- Π is a set of priority rules for mode selection
- P is a set of priority rules for subphase scheduling
- $A \subseteq \mathbb{R}_{\geq 0}$ is a set of values for the engagement selection control parameter
- $B \subseteq \mathbb{R}_{\geq 0}$ is a set of values for the mode selection control parameter
- $\Gamma \subseteq \mathbb{R}_{\geq 0}$ is a set of values for the subphase scheduling control parameter
- $\Sigma \subseteq \{ \text{very small, small, medium-size, large} \}$ is a set of sizes
- $T \subseteq \{ \text{easy, medium, hard} \}$ is a set of tractabilities
- $H \subseteq \mathbb{N}$ is a set of numbers of iterations

Definition 2: A run of an experiment $(\Omega, \Pi, P, A, B, \Gamma, \Sigma, T, H)$ is a tuple

$$(\omega, \pi, \rho, \alpha, \beta, \gamma, \sigma, \tau, \iota) \in (\Omega \times \Pi \times P \times A \times B \times \Gamma \times \Sigma \times T \times H).$$

The outcome of a run is - for each instance attempted - summarized in terms of four result variables. One, $BF_{\omega\pi\rho\beta\gamma\sigma\tau\iota}$ denotes the objective function value of the best solution found in all iterations of that run. Two, $SA_{\omega\pi\rho\beta\gamma\sigma\tau\iota}$ denotes the number of successful attempts, i.e. the number of iterations of the run in which a feasible solution was found. Note that $SA_{\omega\pi\rho\beta\gamma\sigma\tau\iota}$ does not necessarily give the number of different feasible solutions found since the same solution may be generated more than once. Three, $BS_{\omega\pi\rho\beta\gamma\sigma\tau\iota}$ denotes the best, i.e. the maximum number of subphases scheduled in all iterations of that run. Four, $CPU_{\omega\pi\rho\beta\gamma\sigma\tau\iota}$ denotes the average CPU-time of an iteration in a run.

These variables are regarded as random variables which are assumed to be functions of the factors mentioned above, i.e.

$$BF_{\omega\pi\rho\beta\gamma\sigma\tau\iota} = f^{BF}(\omega, \pi, \rho, \alpha, \beta, \gamma, \sigma, \tau, \iota) + \Delta^{BF}_{\omega\pi\rho\beta\gamma\sigma\tau\iota} \quad (36)$$

$$SA_{\omega\pi\rho\beta\gamma\sigma\tau\iota} = f^{SA}(\omega, \pi, \rho, \alpha, \beta, \gamma, \sigma, \tau, \iota) + \Delta^{SA}_{\omega\pi\rho\beta\gamma\sigma\tau\iota} \quad (37)$$

$$BS_{\omega\pi\rho\beta\gamma\sigma\tau\iota} = f^{BS}(\omega, \pi, \rho, \alpha, \beta, \gamma, \sigma, \tau, \iota) + \Delta^{BS}_{\omega\pi\rho\beta\gamma\sigma\tau\iota} \quad (38)$$

$$CPU_{\omega\pi\rho\beta\gamma\sigma\tau\iota} = f^{CPU}(\omega, \pi, \rho, \alpha, \beta, \gamma, \sigma, \tau, \iota) + \Delta^{CPU}_{\omega\pi\rho\beta\gamma\sigma\tau\iota} \quad (39)$$

where $\Delta^{BF}_{\omega\pi\rho\beta\gamma\sigma\tau\iota}$, $\Delta^{SA}_{\omega\pi\rho\beta\gamma\sigma\tau\iota}$, $\Delta^{BS}_{\omega\pi\rho\beta\gamma\sigma\tau\iota}$, and $\Delta^{CPU}_{\omega\pi\rho\beta\gamma\sigma\tau\iota}$ denote the respective random errors. It is assumed that the errors are mutually independent and drawn from the same continuous but not necessarily normal distribution.

5.3 Experimental Design

Due to the computational effort required to attempt a sample of all sizes, the scope of the experiment was limited to include only small and very small instances. Though no obstacle for using *RAMSES* even on large instances, this effort prevents the undertaking of a full factorial design experiment covering all instance classes. However, it is a widely accepted conjecture that rules performing well on small instances are also the best-performing ones for the larger ones (Davis and Patterson 1975, Badiru 1988, Alvarez-Valdés and Tamarit 1989). Of each instance class (σ, τ) , ten instances were considered in the experiment. They were tackled by the following algorithms: In the first stage, the rule combinations (PRS, MAX PRF), (MIN TRU MAX PRF, MAX PRC), and (MIN TRU MAX PRF, MAX RCP) were employed, combined in the second stage with all rules presented above. This reduced scope of the experiment can be justified by the following observations: First, using the static rule MAX PRF in the first stage, all possible engagement selection rules will yield the same results. Second, regardless of the specific mode selection rule used, the first stage will produce the same results as PRS whenever $\alpha = 0$.

For each algorithm (ω, π, ρ) we set $\varepsilon = 1$. The control parameters α , β , and γ were varied independently over the values 0, 2, and 4. Each instance was attempted 1000 times per algorithm and combination of control parameter values. Comprising three values each for α , β , γ , three tractabilities, two sizes, 10 instances for each instance class, and 1000 iterations for each combination of these factors, this design amounts to a number of 1,620,000 iterations for each algorithm (ω, π, ρ) or 48,600,000 iterations in total. In a more compact form, the experimental design can be summarized as

$$\begin{aligned}
 \Omega \times \Pi &= \{(\text{PRS}, \text{MAX PRF}), (\text{MIN TRU MAX PRF}, \text{MAX PRC}), \\
 &\quad (\text{MIN TRU MAX PRF}, \text{MAX RCP})\} \\
 P &= \{\text{MTS}, \text{MIN EFT}, \text{MIN LFT}, \text{MIN SLK}, \text{MAX TRU}, \text{MIN TRU}, \\
 &\quad \text{MAX TRC}, \text{MIN TRC}, \text{MAX RRU}, \text{MIN RRU}\} \\
 A &= \{0, 2, 4\} \\
 B &= \{0, 2, 4\} \\
 \Gamma &= \{0, 2, 4\} \\
 \Sigma &= \{\text{very small, small}\} \\
 T &= \{\text{easy, medium, hard}\} \\
 H &= \{1000\}
 \end{aligned}$$

5.4 Performance Measures

Based upon the result variables defined above, we introduce several performance measures allowing to summarize the outcome of an experiment in a convenient way.

The efficiency ratio $\Phi_{\omega\pi\rho\alpha\beta\gamma\sigma\tau}$ of an algorithm (ω, π, ρ) using the control parameter values (α, β, γ) for an instance class (σ, τ) in i iterations is computed - as the average over all instances of that class attempted - as the quotient of $\text{BF}_{\omega\pi\rho\alpha\beta\gamma\sigma\tau}$ and the objective function value of the best known solution for each instance attempted. (This measure is developing further the rule efficiency ratio proposed by Badiru 1988.) Clearly, $\Phi_{\omega\pi\rho\alpha\beta\gamma\sigma\tau} \in [0, 1]$; a ratio of one indicates that the considered run produced the best value currently known, while a ratio of zero reflects the fact that no feasible solution was generated at all.

The acceptance ratio $\chi_{\omega\pi\rho\alpha\beta\gamma\sigma\tau}$ adjusts $\text{SA}_{\omega\pi\rho\alpha\beta\gamma\sigma\tau}$ by the total number of iterations performed in the run. Thus, it yields an estimate of the number of successful attempts to be expected from a run. Again, $\chi_{\omega\pi\rho\alpha\beta\gamma\sigma\tau} \in [0, 1]$; a ratio of one indicates that all iterations of the run found a feasible solution, a ratio of zero reflects the fact that no feasible solution was generated at all.

The feasibility ratio $\psi_{\omega\pi\rho\alpha\beta\gamma\sigma\tau}$ divides $\text{BS}_{\omega\pi\rho\alpha\beta\gamma\sigma\tau}$ by the total number of subphases to be scheduled in that run. $\psi_{\omega\pi\rho\alpha\beta\gamma\sigma\tau} \in [0, 1]$, as well; a ratio of zero indicates that no subphase has been scheduled in that run, a ratio of 1 indicates that in the best iteration all the sub-

phases have been scheduled, in other words that the corresponding schedule is feasible. While $\chi_{\omega\pi\rho\alpha\beta\gamma\sigma\tau}$ indicates how many of all iterations produced a feasible solution, $\Psi_{\omega\pi\rho\alpha\beta\gamma\sigma\tau}$ gives an indication of how far away the best solutions found are from feasibility.

While it is common practice to use some kind of efficiency ratio to evaluate the performance of an algorithm, we would like to expand somewhat on the other two measures. Note that a small efficiency ratio may either indicate a low performance of the respective algorithm, i.e. its inability to find substantial numbers of feasible solutions, or as well its difficulty to find feasible solutions. In that situation further insight is provided by the second performance measure: High acceptance ratios combined with low efficiency ratios indeed mark a poor performance in terms of the quality of the solutions found whereas low values for both measures point to the little number of feasible solutions which were found at all. In this latter case, where in most iterations only a part of the subphases was scheduled, we found another information to be of interest. Usually one would regard an algorithm scheduling 99 % of the subphases as better than another one scheduling only 1 % of them. This consideration brought up the third measure: High feasibility ratios combined with low acceptance ratios exhibit that in most cases - even if not all - at least nearly all subphases were successfully scheduled, while low values for both measures demonstrate that in fact the algorithm is unfit for scheduling major proportions of subphases.

Finally, the average running time $\vartheta_{\omega\pi\rho\alpha\beta\gamma\sigma\tau}$ is defined as equal to $\text{CPU}_{\omega\pi\rho\alpha\beta\gamma\sigma\tau}$.

To evaluate the effect of varying the above factors, different aggregate measures were derived from the above definitions. These aggregations served to isolate the effects of certain factors. We refrain, however, from the tedious task of citing the respective definitions since they use simple averaging over all factors except of those to be tested. Clearly, the average values can be interpreted as approximations of the expected values of the performance measures in general.

6. Computational Results

The results of our experiment on the small instances with respect to efficiency, acceptance, and feasibility are summarized in Tables 4, 5, and 6 where - as far as possible - the ten algorithms performing best are listed in decreasing order of their performance. Regarding the hard instances, however, only one of them was successfully solved: In 1000 iterations two algorithms found a feasible solution while a third algorithm produced a feasible solution under three different control parameter constellations. Hence, for the hard instances only the feasibility ratios are given.

From Table 4 it is obvious that the tractability has a substantial effect on the efficiency ratio. The rather small values indicate that the algorithms produce good solutions only under certain constellations of the control parameters. In addition, one has to be aware of the fact that also runs which found no feasible solution at all are reflected in these values. The algorithms $(\omega, \pi, \rho) \in \{\text{PRS}\} \times \{\text{MAX PRF}\} \times \{\text{MIN LFT, MIN SLK, MIN TRC}\}$ demonstrate the best efficiency ratios.

Table 4

Rankings of the algorithms according to the efficiency ratios on the small instances

easy		medium	
$\omega\pi\rho$	$\Phi_{\omega\pi\rho\sigma\tau}$	$\omega\pi\rho$	$\Phi_{\omega\pi\rho\sigma\tau}$
PRS, MAX PRF, MIN SLK	0.6024	PRS, MAX PRF, MIN LFT	0.1640
PRS, MAX PRF, MIN TRC	0.5925	PRS, MAX PRF, MIN TRC	0.1541
PRS, MAX PRF, MIN LFT	0.5925	PRS, MAX PRF, MIN SLK	0.1537
PRS, MAX PRF, MIN EFT	0.5636	MIN TRU MAX PRF, MAX PRC, MIN TRC	0.1528
PRS, MAX PRF, MIN RRU	0.5561	MIN TRU MAX PRF, MAX PRC, MIN LFT	0.1505
PRS, MAX PRF, MAX TRU	0.5495	PRS, MAX PRF, MIN RRU	0.1426
MIN TRU MAX PRF, MAX PRC, MIN LFT	0.5489	MIN TRU MAX PRF, MAX PRC, MIN EFT	0.1410
PRS, MAX PRF, MIN TRU	0.5421	MIN TRU MAX PRF, MAX RCP, MIN TRC	0.1361
MIN TRU MAX PRF, MAX PRC, MIN TRC	0.5363	MIN TRU MAX PRF, MAX PRC, MIN TRU	0.1355
PRS, MAX PRF, MAX TRC	0.5328	MIN TRU MAX PRF, MAX RCP, MIN SLK	0.1334

The small acceptance ratios in Table 5 indicate that often a run generates only few feasible solutions. Again, the tractability has a considerable effect on the performance. Table 5 shows that the algorithm $(\omega, \pi, \rho) = (\text{MIN TRU MAX PRF, MAX RCP, MIN TRC})$ produces the best acceptance ratios, regardless of the tractability of the instances attempted.

Table 5

Rankings of the algorithms according to the acceptance ratios on the small instances

easy		medium	
$\omega\pi\rho$	$\chi_{\omega\pi\rho\sigma}$	$\omega\pi\rho$	$\chi_{\omega\pi\rho\sigma}$
MIN TRU MAX PRF, MAX RCP, MIN TRC	0.0369	MIN TRU MAX PRF, MAX RCP, MIN TRC	0.0018
MIN TRU MAX PRF, MAX RCP, MIN LFT	0.0357	PRS, MAX PRF, MIN TRC	0.0017
MIN TRU MAX PRF, MAX RCP, MIN TRU	0.0349	PRS, MAX PRF, MIN LFT	0.0017
MIN TRU MAX PRF, MAX RCP, MIN SLK	0.0336	MIN TRU MAX PRF, MAX PRC, MIN TRC	0.0016
MIN TRU MAX PRF, MAX RCP, MIN RRU	0.0323	MIN TRU MAX PRF, MAX RCP, MIN LFT	0.0016
PRS, MAX PRF, MIN LFT	0.0318	MIN TRU MAX PRF, MAX PRC, MIN LFT	0.0016
PRS, MAX PRF, MIN TRC	0.0308	PRS, MAX PRF, MIN TRU	0.0015
MIN TRU MAX PRF, MAX RCP, MTS	0.0307	PRS, MAX PRF, MIN SLK	0.0014
MIN TRU MAX PRF, MAX PRC, MIN LFT	0.0306	MIN TRU MAX PRF, MAX RCP, MIN SLK	0.0013
PRS, MAX PRF, MIN SLK	0.0298	MIN TRU MAX PRF, MAX RCP, MIN TRU	0.0013

Table 6 demonstrates that the tractability of an instance does also affect the feasibility ratio. The algorithms $(\omega, \pi, \rho) \in \{\text{PRS}\} \times \{\text{MAX PRF}\} \times \{\text{MIN LFT, MIN SLK, MIN TRC}\}$, which already demonstrated the best efficiency ratios, also show the best feasibility ratios, but only as long as the resources are plentiful available. In fact, the scarcer the resources, i.e. the harder the instances attempted, the worse the ranking of the static rules $(\omega, \pi) = (\text{PRS, MAX PRF})$ becomes. In other words, the harder the instances, the better do the dynamic, resource-based rules $(\omega, \pi) \in \{\text{MIN TRU MAX PRF}\} \times \{\text{MAX RCP, MAX PRC}\}$ score against the above static rules. For the medium and the hard instances, the algorithms $(\omega, \pi, \rho) \in \{\text{MIN TRU MAX PRF}\} \times \{\text{MAX RCP}\} \times \{\text{MTS, MIN LFT, MIN TRC}\}$ are the highest ranking ones.

Table 6

Rankings of the algorithms according to the feasibility ratios on the small instances

easy		medium		hard	
$\omega\pi\rho$	$\Psi_{\omega\pi\rho\sigma}$	$\omega\pi\rho$	$\Psi_{\omega\pi\rho\sigma}$	$\omega\pi\rho$	$\Psi_{\omega\pi\rho\sigma}$
PRS, MAX PRF, MIN TRC	0.9943	MIN TRU MAX PRF, MAX RCP, MIN LFT	0.9376	MIN TRU MAX PRF, MAX RCP, MTS	0.8041
PRS, MAX PRF, MIN SLK	0.9942	MIN TRU MAX PRF, MAX RCP, MTS	0.9354	MIN TRU MAX PRF, MAX RCP, MIN LFT	0.8031
PRS, MAX PRF, MIN LFT	0.9941	MIN TRU MAX PRF, MAX RCP, MIN SLK	0.9336	MIN TRU MAX PRF, MAX RCP, MIN TRC	0.8017
PRS, MAX PRF, MIN TRU	0.9937	MIN TRU MAX PRF, MAX RCP, MIN TRC	0.9934	MIN TRU MAX PRF, MAX PRC, MIN LFT	0.7974
PRS, MAX PRF, MIN RRU	0.9936	PRS, MAX PRF, MIN LFT	0.9328	MIN TRU MAX PRF, MAX PRC, MTS	0.7972
PRS, MAX PRF, MIN EFT	0.9935	PRS, MAX PRF, MTS	0.9326	MIN TRU MAX PRF, MAX PRC, MIN TRC	0.7968
PRS, MAX PRF, MTS	0.9933	MIN TRU MAX PRF, MAX RCP, MIN TRU	0.9318	MIN TRU MAX PRF, MAX RCP, MIN TRU	0.7964
PRS, MAX PRF, MAX TRU	0.9928	MIN TRU MAX PRF, MAX PRC, MIN LFT	0.9313	MIN TRU MAX PRF, MAX RCP, MIN SLK	0.7944
PRS, MAX PRF, MAX RRU	0.9927	MIN TRU MAX RCP, MAX RCP, MIN RRU	0.9302	MIN TRU MAX PRF, MAX RCP, MIN RRU	0.7938
MIN TRU MAX PRF, MAX PRC, MIN TRU	0.9926	PRS, MAX PRF, MIN TRC	0.9298	MIN TRU MAX PRF, MAX PRC, MIN TRU	0.7923

All in all, different algorithms score differently for the above mentioned performance measures. However, some common patterns can be derived from our results. The algorithms $(\omega, \pi, \rho) \in \{\text{PRS}\} \times \{\text{MAX PRF}\} \times \{\text{MIN LFT, MIN SLK, MIN TRC}\}$ score best on the efficiency ratio and - in the presence of nonscarce resources - on the feasibility ratio. On the other hand, the algorithms $(\omega, \pi, \rho) \in \{\text{MIN TRU MAX PRF}\} \times \{\text{MAX RCP}\} \times \{\text{MTS, MIN LFT, MIN TRC}\}$ perform - in the presence of scarce resources - best regarding the feasibility ratio. Though the results for the acceptance ratio are rather inconclusive, it is easy to see that the rules $\rho \in \{\text{MIN LFT, MIN TRC}\}$ rank best, regardless of the resource scarcity. We may thus summarize some general conclusions:

- The decision of which combination of rules ω , π , and ρ to apply to a given kind of instance can be simplified by selecting separately a combination of rules (ω, π) and a rule ρ .
- With respect to the efficiency ratio, the combination $(\omega, \pi) = (\text{PRS, MAX PRF})$ performs best. However, the ranking of $(\omega, \pi) \in \{\text{MIN TRU MAX PRF}\}$

- × {MAX RCP, MAX PRC} improves with decreasing resource availability. We thus conjecture that in the presence of very scarce resources the latter combinations may yield equally good or even better efficiency values.
- With respect to the feasibility ratio, in the case of nonscarce resources the combination $(\omega, \pi) = (\text{PRS}, \text{MAX PRF})$ performs best, in the case of scarce resources the combination $(\omega, \pi) \in \{\text{MIN TRU MAX PRF}\} \times \{\text{MAX RCP, MAX PRC}\}$ produces the best results.
- The rules $\rho \in \{\text{MIN LFT, MIN TRC}\}$ perform best with respect to all these performance measures.

The effects of varying the values of the control parameter α , β , and γ were examined in a separate analysis where for each ratio the above best performing rules were considered, leading to the following observations:

- The highest efficiency ratios were obtained for $\alpha \in \{0, 2, 4\}$, $\beta = 0$, and $\gamma = 4$. Due to the static nature of the rule MAX PRF, choosing different rules ω or varying the values of α has no effect on the efficiency ratio. Since $\beta = 0$ implies a purely random selection, the good results for this value indicate that MAX PRF is performing rather poorly, producing a high percentage of infeasible solutions. The rules $\rho \in \{\text{MIN LFT, MIN SLK, MIN TRC}\}$, however, yield their best results for $\gamma = 4$ and thus demonstrate their general suitability.
- The highest acceptance ratios as well as the highest feasibility ratios were reached for $\alpha = 4$, $\beta = 0$ und $\gamma = 4$. It is clear that under all circumstances the consideration of remaining capacities as done in MAX PRF and MAX PRC (cp. the second case of equations (19) and (21), respectively) will have a positive effect on acceptance and feasibility ratio since it guarantees that those modes are excluded from consideration for which the remaining capacities are not sufficient such that such a mode assignment would render the corresponding schedule infeasible. The surprising result for β , however, allows to conclude that the specific selection strategy incorporated in MAX PRF and MAX PRC (cp. the first case of equations (19) and (21), respectively) is performing worse than pure random sampling.

The average running times of the best-performing algorithms are provided by Table 7. All algorithms were coded in C and implemented on an IBM RS/6000 model 550. The running times are given in terms of milliseconds. Here, also the results for the very small instances are provided. Obviously, the running times increase with increasing instance sizes. In the first stage, the application of dynamic rules increases the running times between three and four times, compared to those of static rules, while the respective differences between static and dynamic rules in the second stage are rather small.

Table 7

Average running times for very small and small instances and all tractabilities

$\omega\pi\rho$	$\vartheta_{\omega\pi\rho\sigma}$	
$\omega\pi\rho$	very small	small
PRS, MAX PRF, MIN SLK	1.31	8.39
PRS, MAX PRF, MIN LFT	1.32	8.44
PRS, MAX PRF, MIN TRC	1.45	11.66
MIN TRU MAX PRF, MAX RCP, MTS	1.99	27.13
MIN TRU MAX PRF, MAX PRC, MTS	2.42	30.01
MIN TRU MAX PRF, MAX RCP, MIN TRC	2.13	30.64

In addition, we tried to solve the very small and the small instances with the scientific optimization program LINDO (Version 5.0). LINDO as well as the algorithms were able to find the optimum solutions for all very small instances. While *RAMSES* took about 1,9 seconds for 1000 iterations on each instance, LINDO required between several seconds and several minutes. The exact results are summarized in Table 8.

The differences became even more pronounced for the small and easy instances where LINDO found within a time limit of 1500 minutes a solution for only four of the ten instances, taking about 2, 5 3/4, 10 3/4, and 11 1/4 hours. In contrast, each algorithm found a feasible solution for each instance, which was the optimum solution for each of the ones solved by LINDO. On average, this took about 37 seconds for 1000 iterations.

Table 8
Average running times for very small instances

	RAMSES (1000 iterations)			LINDO		
	easy	medium	hard	easy	medium	hard
95%	0:01.95	0:01.93	0:01.94	1:04.31	3:08.56	3:18.10
Maximum	0:03.50	0:03.40	0:03.20	2:52.96	18:13.24	12:25.10
Std. Dev.	0:00.50	0:00.49	0:00.50	0:56.86	5:19.33	4:18.42

Entries in the format minutes : seconds . milliseconds

In another attempt to solve the small and easy instances used in the experiment with standard solvers, we employed OSL (Release 2). Even though OSL is generally reported to be considerably faster than LINDO, it solved - within the same time limit of 1500 minutes - only two of the instances already solved by LINDO, taking about two minutes and 59 minutes, respectively.

To summarize, we infer from the results outlined that *RAMSES*, using the algorithms presented, is capable of constructing feasible solutions for at least small instances of the MASSP in neglectable amounts of time. Moreover, the solutions produced are - at least in the presence of non-scarce resources - near optimal. Under scarce resources, *RAMSES* may not always generate a feasible solution; still, by scheduling the majority of all subphases it can contribute to a more rational and cost-efficient way of audit-staff scheduling in praxi.

7. Summary and Conclusions

Using the medium-term audit-staff scheduling problem MASSP as an example, we demonstrated how the solution approach *RAMSES*, introduced in Part I of this paper, can be tailored to a specific scheduling problem. We outlined the design of a corresponding instance generator, being able to produce feasible *practical* instances of the MASSP, along with the statistical model, the experimental design, and the performance measures used in an extensive computational study. The results of this study demonstrate that *RAMSES* is in fact capable of producing good solutions in neglectable amounts of time.

References

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, 1989, 113-134.

Badiru, A.B., "Towards the Standardization of Performance Measures for Project Scheduling Heuristics", *IEEE Transactions on Engineering Management*, 35 (1988), 82-89.

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.

Balachandran, B.V. and A.A. Zoltners, "An Interactive Audit-Staff Scheduling Decision Support System", *The Accounting Review*, 56 (1981), 801-812.

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.

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.

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.

Kolisch, R., A. Sprecher, and A. Drexl, "Characterization and Generation of a General Class of Resource-Constrained Project Scheduling Problems", to appear in *Management Science*, 41 (1996).

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.

Levy, F.K., G.L. Thompson, and J.D. Wiest, "Introduction to the Critical-Path Method", in: Muth, J.F. and G.L. Thompson (eds.), *Industrial Scheduling*, Prentice-Hall, Englewood Cliffs (N.J.), 1963.

Patterson, J.H., "A Comparison of Exact Approaches for Solving the Multiple Constrained Resource, Project Scheduling Problem", *Management Science*, 30 (1984), 854-867.

Pritsker, A.A.A., 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.

Salewski, F. and A. Drexel, "Personaleinsatzplanung in Wirtschaftsprüfungsgesellschaften - Bestandsaufnahme und konzeptioneller Ansatz", *Zeitschrift für Betriebswirtschaft*, 63 (1993), 1357-1376.

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).