ECONSTOR
Make Your Publications Visible.

A Service of
ZBW
Leibniz-Informationszentrum Wirtschaft
Leibniz Information Centre for Economics

Drexl, Andreas; Juretzka, Jan; Salewski, Frank

Working Paper — Digitized Version

Academic course scheduling under workload and changeover constraints

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 337

Provided in Cooperation with:
Christian-Albrechts-University of Kiel, Institute of Business Administration

Suggested Citation: Drexl, Andreas; Juretzka, Jan; Salewski, Frank (1993) : Academic course scheduling under workload and changeover constraints, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 337, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:
https://hdl.handle.net/10419/155413

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.

WWW.ECONSTOR.EU

Mitglied der Leibniz-Gemeinschaft

No. 337

Academic Course Scheduling Under Workload and Changeover Constraints

Drexl / Juretzka / Salewski

December 1993

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

Abstract: Two phases can be distinguished in academic course scheduling: In phase one lectures have to be assigned to professors, whereas in phase two the lectures have to be scheduled. Here we assume that the lectures are already assigned to professors, i.e. the first phase has been done. For the second phase we look for a schedule of lectures (for one week) in such a way that several restrictions (availability of professors and students; workloads per day; changeover times; etc.) are observed. The problem under consideration is represented as a binary optimization model, where we make use of a new type of resources, i.e. so-called partially renewable resources. It is shown that even the feasibility version of the problem is NP-complete. We present heuristics which are essentially based on three ideas: First, prevent blocking of lectures, rooms and/or time-slots; second, look at the worst-case consequence of local decisions; third, perform regret based biased random sampling. We provide an instance generator for the generation of a representative set of instances. The generator along with a statistical model are used for a thorough experimental evaluation of the methods. Computational results show that the methods solve medium-sized instances to suboptimality.

Keywords: Academic course scheduling, workload constraints, changeover times, partially renewable resources, regret based biased random sampling

1. Introduction

In academic course scheduling, two phases are usually distinguished: In phase one lectures have to be assigned to professors, whereas in phase two the lectures have to be scheduled. Here we assume that the lectures are already assigned to professors. For the second phase we look for a schedule of lectures (for one week) in such a way that several restrictions (availability of professors and students; workloads per day; changeover times; etc.) are observed. In the following we will use the terms "course" and "lecture" as synonyms; moreover, we are concerned with "course scheduling" despite the fact, that we don't consider phase one. Note that in course scheduling for universities there do not exist classes like in timetabling for schools.

The following items may be used in order to identify a specific course scheduling model: Lectures of equal/different length; precedence relations; availability of rooms of equal/different size and/or equipment; changeover times between rooms belonging to different zones; workload constraints, i.e. upper bounds on the maximum number of lectures per day for a specific subject; preferences of professors for rooms and/or hours. Note that the model presented in Section 3 incorporates all these items in a rather general form.

A survey of related work may be given as follows: **Tripathy** (1980) considers a course scheduling problem, which does not allow to take care of rooms of different sizes. **Ferland, Roy** (1985) decompose the whole problem into two subproblems. Loosely speaking, one

coordinates the temporal requirements, whereas the other looks at the requirements for rooms. Note that travel times and upper bounds on the maximum number of lectures per day are not taken into acccount. An interactive man-machine approach is described in **Mulvey** (1982). **Harwood, Lawless** (1975), **Dyer, Mulvey** (1976) as well as **McClure, Wells** (1984) concentrate on phase one, whereas **Shih, Sullivan** (1977) and **Dinkel, Mote, Venkataramanan** (1989) consider both phases simultaneously, yet in a rather restricted way. A general model is described in **Heinrich** (1984) and **Bettin** (1986); one of the main differences to our model is, that rooms are differentiated only with respect to their size and not with respect to their specific equipment. Approaches based on graph colouring may be found in **de Werra** (1985), **Hertz** (1991), (1992) and **Kiaer, Yellen** (1992). While in the former it is difficult to incorporate the requirements for specific rooms, one of the drawbacks of the latter is the absence of lectures of different length. Timetabling problems are dealt with in, e.g., **Aubin, Ferland** (1989), **Abramson** (1991) and **Cangalovic, Schreuder** (1991), wheras the related exam scheduling problem is treated in, e.g., **Carter** (1986), **Balakrishnan, Lucena, Wong** (1992).

The outline of the paper is as follows: In Section 2 we present the assumptions and the notation. In Section 3 the mathematical programming is described. Section 4 provides methods for solving the model approximately. The results of the experimental evaluation can be found in Section 5. Section 6 gives a summary and some conclusions.

2. Assumptions and Notation

The problem under consideration is based on the following assumptions:

- The planning horizon is denoted with W (e.g. one week with W = 5 days), $w = 1,\dots,W$ represents macro-periods (days). Each macro-period w is subdivided into p_w micro-periods (e.g. hours). Thus we have $t = 1,\dots,T$ periods (time slots) with $T = \Sigma\, p_w$.
- $r = 1,\dots,R$ categories of rooms are available, with λ_{rt} being the number of rooms of category r available in time slot t. (Noteworthy to mention that this allows to model the availability of rooms of equal/different size and/or equipment.)
- Let $j = 1,\dots,J$ denote all possible assignments of room categories to time slots, in short room-periods. Obviously, we have $J = R \cdot T$.
- $i = 1,\dots,I$ lectures have to be scheduled, where lecture i has a duration of d_i micro-periods, hence the general case with lectures of different lengths is considered.
- $J_i \in \{1,\dots,J\}$ denotes the set of possible assignments of lecture i to room-periods, which are considered to be relevant to a specific instance. (Note that the cardinality of J_i in general turns out to be rather small).

- c_{ij} denotes the "cost" of assigning subject i to room-period $j \in J_i$. Clearly, c_{ij} should be used for taking into account the preferences of professors for specific rooms and/or hours.
- V_i denotes the set of subjects i' which have to scheduled in parallel with subject i. (Note that this defines a specific partial order of lectures, i.e. precedence relations.)
- There are d = 1,...,D professors available; in addition, there are s = 1,...,S' sets of students attending the lectures; note that in general more than one such set of students has to/wants to attend a specific lecture. Θ_{st} denotes the "availability level" (e.g. 0 or 1) of set/professor s = 1,...,S',S'+1,...,S'+D-1,S'+D in time slot t. For the sake of shortness we will set S = S' +D in the following. Though keeping in mind that in course scheduling we don't have classes (like, e.g., in timetabling) for ease of description we will use the term "class" s in order to denote a set of students s in the following.
- $S_i \in \{1,...,S',S'+1,...,S\} \cup \{\phi\}$ denotes the set of professors and "classes" belonging to lecture i.
- There are z = 1,...,Z zones, the rooms belong to; H_z denotes the set of rooms in zone z. Changing from a room in zone z to a room in zone z', $z \neq z'$, in subsequent time slots is feasible only, if there is enough changeover (travel, setup) time. We assume that one micro-period is necessary in order to change between rooms of different zones in subsequent time slots.
- K_m, m = 1,...,M, denotes the set of (conflicting) lectures, which are only allowed to be scheduled at day w up to a maximum number of κ_{mw} lectures. In other words, a subject (e.g. mathematics) is only allowed to be given/attended up to the maximum workload of κ_{mw} lectures per day. (Note that the generalization of this item, i.e. to consider not only the number of lectures but the sum of their durations, could easily be incorporated.)

Based on these assumptions we are going to formulate the model. In order to simplify the presentation we first relate

- j to t, R, and r,
- r to j and R, and
- t to j, r, and R

by the use of the following mappings:

	Function Symbol	Explicit Function
j	$j = e(t,r,R)$	$j = (t-1)R+r$
r	$r = \bar{f}(j,R) = f(j)$	$r = j \bmod R$ [with $r := R$ if $(j \bmod R) = 0$]
t	$t = \bar{\bar{g}}(j,r,R) = \bar{g}(j,f(j)) = g(j)$	$t = (j-r)/R+1$

Then we derive the symbols

$$FP_w = 1 + \sum_{v=1}^{w-1} p_v$$

(first period of day w)

$$LP_w = \sum_{v=1}^{w} p_v$$

(last period of day w)

and the sets

$$J_{iw} = \{ j \mid j \in J_i \wedge FP_w \leq j \leq LP_w \}$$

(set of room-periods of lecture i on day w)

$$R_{iz} = \{ r \mid \exists (j \in J_i): r = f(j) \wedge r \in H_z \}$$

(set of rooms of lecture i in zone z)

$$R_i = R_{i1} \cup \ldots \cup R_{iZ}$$

(set of rooms of lecture i within all zones)

$$IR_{rt} = \{ i \mid \exists (j \in J_i): r = f(j) \wedge g(j) - d_i + 1 \leq t \leq g(j) \}$$

(set of lectures i being schedulable in room categorie r and period t)

$$IS_{st} = \{ i \mid s \in S_i \wedge \exists (j \in J_i): g(j) - d_i + 1 \leq t \leq g(j) \}$$

(set of lectures i being schedulable for professor/class s in period t)

$$A_{zst} = \{ i \mid s \in S_i \wedge \exists (j \in J_i): t = g(j) - d_i + 1 \wedge f(j) \in H_z \}$$

(set of lectures i being schedulable for professor/class s and starting in period t within zone z)

$$E_{zst} = \{ i \mid s \in S_i \wedge \exists (j \in J_i): t = g(j) \wedge f(j) \in H_z \}$$

(set of lectures i being schedulable for professor/class s and ending in period t within zone z)

By the use of these abbreviations we are able to state in the following the problem under consideration in a compact way.

3. Model

Now we are going to formulate the mathematical programming model. We use binary variables

$$x_{ij} = \begin{cases} 1, & \text{if lecture i is assigned to room-period j} \\ 0, & \text{otherwise} \end{cases}$$

and derive the following Course Scheduling Problem (CSP):

$$\min \sum_{i=1}^{I} \sum_{j \in J_i} c_{ij} x_{ij} \tag{1}$$

$$\text{s.t.} \quad \sum_{j \in J_i} x_{ij} = 1 \qquad (\forall i) \tag{2}$$

$$\sum_{j \in J_h} g(j)\, x_{hj} = \sum_{j \in J_i} g(j)\, x_{ij} \qquad (\forall i, \forall h \in V_i \text{ with } h < i) \tag{3}$$

$$\sum_{i \in IR_{rt}} \sum_{\substack{\tau = t \\ [(\tau-1)R+r] \in J_i}}^{t+d_i-1} x_{i[(\tau-1)R+r]} \le \lambda_{rt} \qquad (\forall r, \forall t \text{ with } |IR_{rt}| > \lambda_{rt}) \tag{4}$$

$$\sum_{i \in IS_{st}} \sum_{r \in R_i} \sum_{\substack{\tau = t \\ [(\tau-1)R+r] \in J_i}}^{t+d_i-1} x_{i[(\tau-1)R+r]} \le \Theta_{st} \qquad (\forall s, \forall t \text{ with } |IS_{st}| > \Theta_{st}) \tag{5}$$

$$\sum_{i \in A_{zst}} \sum_{\substack{r \in R_{iz} \\ [((t+d_i-1)-1)R+r] \in J_i}} x_{i[((t+d_i-1)-1)R+r]} + \sum_{i \in E_{\bar{z}s(t-1)}} \sum_{\substack{r \in R_{i\bar{z}} \\ [((t-1)-1)R+r] \in J_i}} x_{i[((t-1)-1)R+r]} \le 1 \qquad (\forall s, \forall t, \forall z, \forall \bar{z} \text{ with } z \ne \bar{z}) \tag{6}$$

$$\sum_{i \in K_m} \sum_{j \in J_{iw}} x_{ij} \le \kappa_{mw} \qquad (\forall m, \forall w \text{ with } |\bigcup_{\substack{i \in K_m \\ J_{iw} \ne \phi}} i| > \kappa_{mw}) \tag{7}$$

$$x_{ij} \in \{0, 1\} \qquad (\forall i, \forall j) \tag{8}$$

The objective function (1) minimizes the sum of the costs (or equivalently maximizes the sum of preferences) for scheduling all lectures. Equations (2) are lecture completion constraints. Equations (3) secure the required lectures to be scheduled in parallel. Equations (4)

take care of the availability of the room capacities. Equations (5) guarantee, that each professor/class is able to participate at the the relevant lectures. Equations (6) provide for the changeover times needed between lectures to be given in different zones in succeeding time slots; note that only time slots $t = (FP_1 + 1),\ldots,LP_1,(FP_2+1),\ldots,LP_w$, where $A_{zst} \neq \phi$ and $E_{\bar{z}s(t-1)} \neq \phi$ need to be considered. Equations (7) obey the maximum workload per subject and day. (8) establishes the binarity of the variables.

The following theorem states the complexity of the Feasibility variant CSPF of the CSP, where the question is, whether there exists at least a feasible solution for a specific instance or not.

Theorem. The CSPF is NP-complete.

Proof. By reduction to PARTITION. ■

The proof uses the PARTITION problem which is well-known to be NP-complete itself (cf. **Garey , Johnson** (1979)). Details of the proof (for the special case: $T = 2$, $R = 1$, $D = S = Z = 0$, $d_i = 1$ and $V_i = \phi\ \forall i$, no setups) can be found in **Salewski** et al. (1994). Clearly, the NP-completeness of the feasibility problem CSPF in turn implies the NP-hardness of the corresponding optimization problem CSP. In fact, this theorem implies that any scheduling feasibility (optimization) problem comprising at least two periods and one renewable resource with arbitrary usages is NP-complete (NP-hard).

Noteworthy to say, that the CSP (1)-(8) is closely related to nonpreemptive multi-mode resource-constrained project scheduling. Here in addition to the renewable and nonrenewable resources (well-known from project scheduling; cf. e.g. **Drexl** (1991)) the resources are partially renewable, i.e. their usage is restricted to subsets of periods.

4. Methods

Now we present a class of methods with which the CSP can be solved to suboptimality. We proceed as follows: First, we describe the fundamental algorithmic scheme, denoted as method M1. Second, we present an extension, denoted as M2. Third, we provide several local decision rules which can be incorporated into both methods.

A formal description of **M1** in pseudo-code may be given as follows (with comments in brackets):

Initialize

```
I    := ∪_{i=1}^{I} i;                {the set of all lectures}
I0   := I;                            {the set of unscheduled lectures}
I1   := φ;                            {the set of scheduled lectures}
J0_i := J_i        (∀i);              {the set of unscheduled room-periods of lectures i}
x_ij := 0          (∀i, ∀j);          {initialize the decision variables}

while (I0 ≠ φ) do                     {stop when there is no schedulable lecture}
begin
   select (i ∈ I0) and (j ∈ J0_i);    {the criteria will be defined below}
   x_ij := 1;                         {lecture i is assigned to room-period j}
   I0 := I0 \ {i};                    {update I0}
   I1 := I1 ∪ {i};                    {update I1}
   if (V_i ≠ φ) then
   begin
     for (i' ∈ V_i) do                {schedule the parallel lectures i'}
     begin
        select (j' ∈ J0_i');          {the criteria will be defined below}
        x_i'j' := 1;                  {lecture i' is assigned to room-period j'}
        I0 := I0 \ {i'};              {update I0}
        I1 := I1 ∪ {i'};              {update I1}
     end
   end
   update J0_i (∀ i ∈ I0);            {update the set of room-periods}
   for i ∈ I0 with |J0_i| = 0 do      {lecture i may not be scheduled}
   begin
     I0 := I0 \ {i};                  {lecture i is not schedulable}
   end
end

evaluate the schedule x;              {compute the performance measures}
```

Clearly, the solution produced is a feasible schedule if $\mathcal{I}1 = \mathcal{I}$ holds. Performance measures are computed for both infeasible and feasible schedules in order to evaluate specific variants of the general algorithmic scheme (for details see Section 5).

M2 differs from M1 with respect to: (i) Lectures $i \in \mathcal{I}0$ for which only one room-period remains feasible (i.e. $|\mathcal{J}0_i| = 1$), are scheduled immediately. (ii) A lecture $i \in \mathcal{I}0$ is forbidden

to be scheduled within a room-period, if for any other lecture $i' \in \mathcal{I}0$ no feasible room-period would remain.

Now we are going to provide the local decision rules, which are used to do the "select ($i \in \mathcal{I}0$) and ($j \in \mathcal{J}0_i$)-"job. First, we present local decision rules selecting lectures and room-periods sequentially. Second, we provide rules selecting lectures and room-periods simultaneously.

I. Sequential Selection of Lectures and Room-Periods

The selection of lectures and room-periods is done in two stages. In stage 1 we select $i \in \mathcal{I}0$ with probability proportional to ω_i^p; in stage 2 we select $j \in \mathcal{J}0_i$ with probability proportional to π_j^p.

In *stage 1* one of the following two criteria is used:

RPMIN: Choose a lecture $i \in \mathcal{I}0$ according to a criterion which prefers those for which the number of Room-Periods (RP_i) available is MINimal. More formally, compute

$$RP_i := |\mathcal{J}0_i| \quad (\forall\, i \in \mathcal{I}0),$$

$$RP^{max} := \max\{RP_i \mid i \in \mathcal{I}0\} \quad \text{and}$$

$$\bar{\omega}_i := RP^{max} - RP_i \quad (\forall\, i \in \mathcal{I}0).$$

PSMAX: Choose a lecture $i \in \mathcal{I}0$ according to a criterion which prefers those for which the number of Professors and Students required is MAXimal. More formally, compute

$$PS_i := |S_i| \quad (\forall\, i \in \mathcal{I}0),$$

$$PS^{min} := \min\{PS_i \mid i \in \mathcal{I}0\} \quad \text{and}$$

$$\bar{\omega}_i := PS_i - PS^{min} \quad (\forall\, i \in \mathcal{I}0).$$

Note that $\bar{\omega}_i$ in both cases represents the worst-case alternative of not choosing lecture i in correspondence with the criterion defined. Thus it establishes the regret of the worst-case decision alternative; cf. **Drexl** (1991) w.r.t. a similar idea successfully applied to project scheduling.

$\bar{\omega}_i$ may be used for calculating decision probabilities ω_i^p as follows: Compute

$$\bar{\bar{\omega}}_i := (\bar{\omega}_i + \epsilon)^a \quad (\forall\, i \in \mathcal{I}0) \quad \text{and}$$

$$\omega_i^p := \bar{\bar{\omega}}_i \Big/ \sum_{i' \in \mathcal{I}0} \bar{\bar{\omega}}_{i'} \quad (\forall\, i \in \mathcal{I}0).$$

$\epsilon > 0$ makes sure that each decision alternative may be chosen with a positive probability; a transformes the term (.) exponentially and thus gives way to control the generation of probabilities.

In *stage 2* the following criterion is used:

BPSEQ: For the lecture i chosen in stage 1 select a room-period j based on the number of pairs (i',j'), $i \neq i'$, going to be blocked by scheduling pair (i,j), i.e. use the Blocked Pairs SEQuential rule. More formally, compute

BP_j := the number of currently unblocked pairs (i',j') going to be blocked by scheduling pair (i,j) $(\forall\, j \in \mathcal{J}0_i)$,

BP^{max} := $\max\{BP_j \mid j \in \mathcal{J}0_i\}$ and

$\bar{\pi}_j$:= $BP^{max} - BP_j$ $(\forall\, j \in \mathcal{J}0_i)$.

$\bar{\pi}_j$ may be used for calculating decision probabilities π^p_i as follows: Compute

$\bar{\bar{\pi}}_j$:= $(\bar{\pi}_j + \epsilon)^b$ $(\forall\, j \in \mathcal{J}0_i)$ and

π^p_j := $\bar{\bar{\pi}}_j \Big/ \sum_{j' \in \mathcal{J}0_i} \bar{\bar{\pi}}_{j'}$ $(\forall\, j \in \mathcal{J}0_i)$.

Once more $\epsilon > 0$ makes sure that each decision alternative may be chosen with a positive probability; b transformes the term (.) exponentially and thus gives way to control the generation of probabilities.

In summary we have for the sequential selection of lectures and room-periods, i.e. for computing the "select $(i \in \mathcal{I}0)$ and $(j \in \mathcal{J}0_i)$"-line of the algorithm, the following two (combined) rules: **RPMIN/BPSEQ** and **PSMAX/BPSEQ**.

II. Simultaneous Selection of Lectures and Room-Periods

The selection of lectures and room-periods is done as follows: Select $i \in \mathcal{I}0$ and $j \in \mathcal{J}0_i$ simultaneously with probability proportional to ξ^p_{ij}. One of the following two criteria may be used:

BPSIM: Choose a pair (i,j) according to a criterion which prefers those with the minimum number of Blocked Pairs (i',j') SIMultaneously. More formally, compute

BP_{ij} := number of currently unblocked pairs (i',j') going to be blocked by scheduling pair (i,j) $(\forall\, i \in \mathcal{I}0, \forall\, j \in \mathcal{J}0_i)$,

BP^{max} := $\max\{BP_{ij} \mid (i,j): i \in \mathcal{I}0 \wedge j \in \mathcal{J}0_i\}$ and

$\bar{\xi}_{ij}$:= $BP^{max} - BP_{ij}$ $(\forall\, i \in \mathcal{I}0, \forall\, j \in \mathcal{J}0_i)$.

CMIN: Choose a pair (i,j) according to a criterion which prefers those for which the Costs are MINimal. More formally, compute

C^{max} := $\max\{c_{ij} \mid (i,j): i \in \mathcal{I}0 \wedge j \in \mathcal{J}0_i\}$ and

$\bar{\xi}_{ij}$:= $C^{max} - c_{ij}$ $(\forall\, i \in \mathcal{I}0, \forall\, j \in \mathcal{J}0_i)$.

$\bar{\xi}_j$ may be used for calculating decision probabilities ξ^P_i as follows: Compute

$$\bar{\bar{\xi}}_{ij} := (\bar{\xi}_{ij} + \epsilon)^g \quad (\forall\, i \in \mathcal{I}0, \forall\, j \in \mathcal{J}0_i) \quad \text{and}$$

$$\xi^P_{ij} := \bar{\bar{\xi}}_{ij} \Big/ \sum_{i' \in \mathcal{I}0} \sum_{j' \in \mathcal{J}0_i} \bar{\bar{\xi}}_{i'j'} \quad (\forall\, i \in \mathcal{I}0, \forall\, j \in \mathcal{J}0_i).$$

Once more $\epsilon > 0$ makes sure that each decision alternative may be chosen with a positive probability; g transformes the term (.) exponentially and thus gives way to control the generation of probabilities.

5. Experimental Evaluation

The methods presented in the preceeding Section have been coded in TURBO PASCAL and implemented on a 386 machine. For experimental purposes we randomly generated test instances. An informal description of the generator may be given as follows: In step 1, the parameters W, p_w $(1 \leq w \leq W)$, R, λ_{rt} $(1 \leq r \leq R,\ 1 \leq t \leq T)$, Z, H_z $(1 \leq z \leq Z)$ and M are used as input. In step 2, with these parameters the other model parameters I, d_i, J_i, V_i, S_i, S, θ_{st}, c_{ij}, K_m, κ_{mw} are computed subsequently. The output is an instance, for which we know that at least one feasible solution exists. The details are rather technical and therefore omitted here for the sake of shortness. The generator and the test instances are available from the authors upon request.

The instances generated are classified with respect to two items, i.e. the *size s* and the (computational) *tractability t.* Clearly, several criteria may be used in order to measure the size and the tractability. According to preliminary computational results not reported here the criteria which will be used in the following allow to discriminate between "small" /" large" as well as "easy" /" hard" instances.

The size $s \in \{s, m, l\}$ of instances is characterized with respect to the number of lectures I as follows:

$s \in \{s, m, l\}$ with	s = small	m = medium	l = large
	$10 \leq I \leq 50$	$51 \leq I \leq 100$	$101 \leq I \leq 500$

The tractability $t \in \{e, m, h\}$ is characterized via the average usage u $(0 \leq u \leq 1)$ of room categories as follows:

$t \in \{e, m, h\}$ with	e = easy	m = medium	h = hard
	$0.5 \leq u \leq 0.7$	$0.7 < u \leq 0.9$	$0.9 < u \leq 1.0$

In order to provide benchmark solutions we used the general MIP-solver LINDO. On an IBM RISC/6000 model 550 it took seconds to minutes to find and to verify the optimal solution for the small instances. For the medium-sized instances, especially for the hard ones,

it took even hours of CPU-time to find one feasible solution. Thus, in general optimal benchmark solutions are unavailable for medium-sized instances. Moreover, w.r.t. the CPU-times required, we did not investigate the large instances $s = \{l\}$.

In order to investigate the performance of the *sequential rules* (RPMIN/BPSEQ and PSMAX/BPSEQ) described in Section 4, the following *statistical model* with nine factors can be stated: An experiment is described by a tupel (M,Q,R,A,B,S,T,N,I) where the symbols have the following meaning:

$M \subseteq \{\mathrm{M1}, \mathrm{M2}\}$	(methods)
$Q \subseteq \{\mathrm{RPMIN}, \mathrm{PSMAX}\}$	(selection of lectures)
$R \subseteq \{\mathrm{BPSEQ}\}$	(selection of room-periods)
$A \subseteq \{0,2,4\}$	(control parameter a for RPMIN and PSMAX)
$B \subseteq \{0,2,4\}$	(control parameter b for BPSEQ)
$S \subseteq \{\mathrm{s}, \mathrm{m}\}$	(size of the instances)
$T \subseteq \{\mathrm{e}, \mathrm{m}, \mathrm{h}\}$	(tractability of the instances)
$N \subseteq \{1,\ldots, 10\}$	(instances generated)
$I \subseteq \{1000\}$	(number of iterations of the method)

Clearly, $(m,q,r,a,b,s,t,n,i) \in (M \times Q \times R \times A \times B \times S \times T \times N \times I)$ now defines a specific experiment. We use the following *performance measures* (cf. **Badiru** (1988)):

- $A_{mqrabstni} \in [0,1]$ is the first acceptance ratio of the experiment (m,q,r,a,b,s,t,n,i). More precisely, it denotes the fraction of feasible solutions provided with the method (m) by the rule (q,r) based on the control parameters (a,b) for the instance (s,t,n) when (i) solutions have been iteratively generated. Clearly, $A_{mqrabstni} = 0$ denotes that no feasible solution has been generated, whereas $A_{mqrabstni} = 1$ denotes that only feasible solutions have been generated.
- $A'_{mqrabstni} \in [0,1]$ is the second acceptance ratio. It defines the fraction of iterations within the experiment (m,q,r,a,b,s,t,n,i) where at least 95 % of the lectures have been scheduled.
- $\mathcal{F}_{mqrabstni} \in [0,1]$ is the feasibility ratio of the experiment (m,q,r,a,b,s,t,n,i). More precisely, it denotes the average number of scheduled lectures with the method (m) by the rule (q,r) based on the control parameters (a,b) for the instance (s,t,n) when (i) solutions have been iteratively generated.
- $C_{mqrabstni}$ denotes the CPU-time required by the experiment (m,q,r,a,b,s,t,n,i).

- $Z_{mqrabstni}$ denotes the best objective function value computed within the experiment (m,q,r,a,b,s,t,n,i). Clearly, we have $Z_{mqrabstni} = \infty$ when no feasible schedule has been found.
- $O_{mqrabstni} \in (0,1]$ denotes the optimality rate, i.e. the ratio of the best known objective function value B_{stn} for a specific instance (s,t,n) divided by $Z_{mqrabstni}$. (Clearly, B_{stn} either corresponds to the objective function value computed by LINDO or to the one computed by our methods.) Note that $O_{mqrabstni} \to 0$ when we did not find a feasible solution within the experiment (m,q,r,a,b,s,t,n,i). $O_{mqrabstni} = 1$ when B_{stn} equals $Z_{mqrabstni}$.

In the following we report aggregated computational results of several experiments. Aggregation means to compute averages and variances, e.g., as follows:

$$A_{mqra} := 1/(|B|\cdot|S|\cdot|T|\cdot|N|\cdot|I|) \sum_{b\in B}\sum_{s\in S}\sum_{t\in T}\sum_{n\in N}\sum_{i\in I} A_{mqrabstni}$$
$$(m\in M,\ q\in Q,\ r\in R,\ a\in A)$$

$$\sigma(A_{mqra})^2 := 1/(|B|\cdot|S|\cdot|T|\cdot|N|\cdot|I|) \sum_{b\in B}\sum_{s\in S}\sum_{t\in T}\sum_{n\in N}\sum_{i\in I} (A_{mqrabstni} - A_{qra})^2$$
$$(m\in M,\ q\in Q,\ r\in R,\ a\in A)$$

$$\sigma(A_{mqra}) := \text{square root of } \sigma(A_{mqra})^2$$
$$(m\in M,\ q\in Q,\ r\in R,\ a\in A)$$

In the following, first we are going to evaluate the performance of the RPMIN- and the BPSEQ-rule within the framework of the method M1 for solving small instances the tractability of which is medium or hard. Thus, unless stated otherwise, the experimental design is as follows: $M = \{\text{M1}\}$, $Q = \{\text{RPMIN}\}$, $R = \{\text{BPSEQ}\}$, $A = \{0,2,4\}$, $B = \{0,2,4\}$, $S = \{\text{s}\}$, $T = \{\text{m},\text{h}\}$, $N = \{1,\ldots,10\}$ and $I = \{1000\}$.

Tables 1 and 2 provide the acceptance ratios A_a and A_a' as well as their variances of the RPMIN/BPSEQ-rule as a function of the control parameter a, respectively. ($A_a' = 0.091$, e.g., means that on the average 9.1% of the iterations can schedule more than 95% of the lectures.) As can be seen both acceptance ratios increase with increasing values of α, where in both cases the variances only slightly increase. Thus, lectures with a low number of available room-periods shoud more likely to be scheduled first than those with a high number of available room-periods. Note that (in comparison with pure random sampling, i.e. $a = 0$) the effect of a on A_a demonstrates that the rule RPMIN/BPSEQ is a decisive local decision criterion.

Table 1: Performance measure $\mathcal{A}_a$ for RPMIN/BPSEQ

a	0	2	4
$\mathcal{A}_a$	0.013	0.020	0.030
$\sigma(\mathcal{A}_a)$	0.016	0.023	0.032

Table 2: Performance measure $\mathcal{A}'_a$ for RPMIN/BPSEQ

a	0	2	4
$\mathcal{A}'_a$	0.045	0.067	0.091
$\sigma(\mathcal{A}'_a)$	0.059	0.077	0.095

Tables 3 and 4 provide the feasibility ratios $\mathcal{F}_a$ and the optimality rates $\mathcal{O}_a$ as a function of a, respectively. The average number of scheduled lectures increases with increasing a, where the variances are decreasing slightly. With $a=4$ in each iteration 81.8% of the lectures could be scheduled on the average. Similarly, a increases the optimality rate though not dealing with the cost coefficients of the objective function.

Table 3: Performance measure $\mathcal{F}_a$ for RPMIN/BPSEQ

a	0	2	4
$\mathcal{F}_a$	0.785	0.805	0.818
$\sigma(\mathcal{F}_a)$	0.058	0.051	0.048

Table 4: Performance measure $\mathcal{O}_a$ for RPMIN/BPSEQ

a	0	2	4
$\mathcal{O}_a$	0.717	0.860	0.941
$\sigma(\mathcal{O}_a)$	0.436	0.328	0.218

Tables 5, 6, 7 and 8 provide the acceptance ratios $\mathcal{A}_b$ and $\mathcal{A}'_b$, the feasibility ratios $\mathcal{F}_b$ and the optimality rates $\mathcal{O}_b$ of the RPMIN/BPSEQ-rule as a function of b, respectively. The interpretation of the results is similar to the one for Tables 1 to 4. Especially, the effect of b on $\mathcal{A}_b$

demonstrates that the rule BPSEQ is a suitable local decision criterion in comparison with pure random sampling.

Table 5: Performance measure $\mathcal{A}_b$ for RPMIN/BPSEQ

b	0	2	4
$\mathcal{A}_b$	0.011	0.023	0.030
$\sigma(\mathcal{A}_b)$	0.013	0.025	0.031

Table 6: Performance measure $\mathcal{A}_b'$ for RPMIN/BPSEQ

b	0	2	4
$\mathcal{A}_b'$	0.034	0.075	0.094
$\sigma(\mathcal{A}_b')$	0.044	0.079	0.096

Table 7: Performance measure $\mathcal{F}_b$ for RPMIN/BPSEQ

b	0	2	4
$\mathcal{F}_b$	0.778	0.810	0.820
$\sigma(\mathcal{F}_b)$	0.053	0.050	0.049

Table 8: Performance measure $\mathcal{O}_b$ for RPMIN/BPSEQ

b	0	2	4
$\mathcal{O}_b$	0.699	0.879	0.940
$\sigma(\mathcal{O}_b)$	0.443	0.309	0.218

Tables 9 provides a comparison of the methods M1 and M2 within the sequential framework considered so far. The experimental design is as follows: $Q = \{\text{RPMIN}\}$, $R = \{\text{BPSEQ}\}$, $A = \{0,2,4\}$, $B = \{0,2,4\}$, $S = \{\text{s}\}$, $T = \{\text{m},\text{h}\}$, $N = \{1,\ldots,10\}$ and $I = \{1000\}$. Note that M2 outperforms M1 w.r.t. each performance measure. Most noteworthy to mention is that the optimality rate $\mathcal{O}_m$ increases from 0.686 to 0.992.

Table 9: Performance measures for M1 and M2

m	M1	M2
$\mathcal{A}_m$	0.010	0.033
$\sigma(\mathcal{A}_m)$	0.017	0.028
$\mathcal{A}'_m$	0.043	0.092
$\sigma(\mathcal{A}'_m)$	0.064	0.087
$\mathcal{F}_m$	0.786	0.819
$\sigma(\mathcal{F}_m)$	0.056	0.045
$\mathcal{O}_m$	0.686	0.992
$\sigma(\mathcal{O}_m)^2$	0.446	0.023

Table 10: Results of M2 for $a = 4$, $b = 4$

m	M2
$\mathcal{A}_m$	0.058
$\sigma(\mathcal{A}_m)$	0.042
$\mathcal{A}'_m$	0.144
$\sigma(\mathcal{A}'_m)$	0.119
$\mathcal{F}_m$	0.839
$\sigma(\mathcal{F}_m)$	0.046
$\mathcal{O}_m$	0.998
$\sigma(\mathcal{O}_m)$	0.005
$\mathcal{C}_m$	1:44:02
$\sigma(\mathcal{C}_m)$	0:08:51

Table 10 provides the results of the method M2 and the RPMIN/BPSEQ-rule associated with the "best" parameters $a = 4$, $b = 4$ evaluated so far. In 5.8% of the iterations a feasible solution has been computed. The fraction of iterations with at least 95% of the lectures scheduled equals 14.4% The feasibility ratio equals 0.839 and is thus the highest obtained so

far. Most noteworthy to mention is the fact, that only for some of the instances the optimal objective function value has not been obtained within the specified number of iterations. The CPU- times are provided in hours:min:sec on a 386 machine with 40 MHz clockpulse. Thus, on the average one iteration of M2 takes 6.2 sec. Note that the computation of the local decision criteria (especially $\mathcal{J}0_i$) is rather time-consuming and that our code is an experimental version which could be speeded-up substantially.

Table 11: Performance measures for PSMAX and RPMIN

q	PSMAX	RPMIN
$\mathcal{A}_q$	0.035	0.058
$\sigma(\mathcal{A}_q)$	0.026	0.042
$\mathcal{A}'_q$	0.106	0.144
$\sigma(\mathcal{A}'_q)$	0.090	0.119
$\mathcal{F}_q$	0.824	0.839
$\sigma(\mathcal{F}_q)$	0.049	0.046
$\mathcal{O}_q$	0.986	0.998
$\sigma(\mathcal{O}_q)$	0.037	0.005

In Table 11 we are going to compare the performance of the PSMAX- and the RPMIN-rule within method M2 for the most promising parameter values $a = 4$ and $b = 4$. The experimental design is as follows: $M = \{\text{M2}\}$, $Q = \{\text{RPMIN}, \text{PSMAX}\}$, $R = \{\text{BPSEQ}\}$, $A = \{4\}$, $B = \{4\}$, $S = \{\text{s}\}$, $T = \{\text{m}, \text{h}\}$, $N = \{1,..., 10\}$ and $I = \{1000\}$. Clearly, RPMIN outperformes PSMAX w.r.t. all performance criteria and therefore PSMAX will be skipped in the sequel.

In order to investigate the performance of the *simultaneous rules* (BPSIM, CMIN) described in section 4, the following *statistical model* with seven factors can be stated: An experiment is described by a tupel (M,L,G,S,T,N,I) where the symbols have the following meaning

$L \subseteq \{\text{BPSIM}, \text{CMIN}\}$ (criteria for the selection of lectures and room-periods),

$G \subseteq \{0,2,4\}$ (control parameter g used for BPSIM and CMIN), and

$M \subseteq \{\text{M1}, \text{M2}\}$, $S \subseteq \{\text{s}, \text{m}\}$, $T \subseteq \{\text{e}, \text{m}, \text{h}\}$, $N \subseteq \{1,..., 10\}$, $I \subseteq \{1000\}$ as defined above.

Clearly, $(m,l,g,s,t,n,i) \in (M \times L \times G \times S \times T \times N \times I)$ now defines a specific experiment. We use the following *performance measures:* Acceptance ratios $\mathcal{A}_{mlgstni}$ and $\mathcal{A}'_{mlgstni}$, feasibility ratio

$\mathcal{F}_{mlgstni} \in [0,1]$, CPU-time $\mathcal{C}_{mlgstni}$, objective function value $\mathcal{Z}_{mlgstni}$ and optimality rate $\mathcal{O}_{mlgstni}$ which are defined analogously as above.

In Table 12 we compare the performance of the rule BPSIM (within the framework of method M1) with the most powerful sequential competitor, i.e. the RPMIN/BPSEQ-rule (with $a = 4$, $b = 4$ and method M1). The experimental design is as follows: $M = \{M1\}$, $L = \{BPSIM\}$, $G = \{4\}$, $S = \{s\}$, $T = \{m\}$, $N = \{1,..., 10\}$ and $I = \{1000\}$. Obviously, BPSIM is outperformed by RPMIN/BPSEQ w.r.t. all performance measures. Especially remarkable is the drastic increase of computation time needed by BPSIM.

Table 12: Performance measures for BPSIM and RPMIN/BPSEQ

l	BPSIM	RPMIN/BPSEQ
$\mathcal{A}_l$	0.029	0.044
$\sigma(\mathcal{A}_l)$	0.026	0.035
$\mathcal{A}'_l$	0.110	0.183
$\sigma(\mathcal{A}'_l)$	0.071	0.090
$\mathcal{F}_l$	0.846	0.869
$\sigma(\mathcal{F}_l)$	0.029	0.024
$\mathcal{O}_l$	0.949	0.987
$\sigma(\mathcal{O}_l)$	0.065	0.017
$\mathcal{C}_l$	29:15:05	2:17:09

Table 13 provides the performance measures of the CMIN-rule within the framework of method M1. Clearly, CMIN is outperformed by the sequential rule RPMIN/BPSEQ w.r.t. all performance measures except CPU-time.

Table 13: Performance measures for M1 / CMIN with $g = 4$

l	CMIN
$\mathcal{A}_l$	0.000
$\sigma(\mathcal{A}_l)$	0.001
$\mathcal{A}'_l$	0.001
$\sigma(\mathcal{A}'_l)$	0.001
$\mathcal{F}_l$	0.704
$\sigma(\mathcal{F}_l)$	0.034
$\mathcal{O}_l$	0.097
$\sigma(\mathcal{O}_l)$	0.305
$\mathcal{C}_l$	1:07:14
$\sigma(\mathcal{C}_l)$	0:01:55

Table 14: Further results of M2 with RPMIN/BPSEQ

(a, b)	(4,4)	(8,8)
$\mathcal{A}_{ab}$	0.045	0.063
$\sigma(\mathcal{A}_{ab})$	0.034	0.047
$\mathcal{A}'_{ab}$	0.059	0.078
$\sigma(\mathcal{A}'_{ab})$	0.040	0.057
$\mathcal{F}_{ab}$	0.800	0.810
$\sigma(\mathcal{F}_{ab})$	0.023	0.026
$\mathcal{O}_{ab}$	1.000	1.000
$\sigma(\mathcal{O}_{ab})$	0.000	0.000

Table 14 provides further results for the currently most promising rule RPMIN/BPSEQ within the framework of the method M2. For instances $S = \{s\}$ and $T = \{h\}$ we increased the control parameters (a, b) to (8,8). Obviously, this leads to an improvement of the acceptance ratios. In addition, we get remarkable optimality rates for both sets of parameters.

Table 15 provides additional results for the rule RPMIN / BPSEQ within the framework of the method M2. For one instance with $S = \{m\}$ and $T = \{h\}$ we increased the control parameters (a, b) from (2,2) to (4,4) and then successively to (12, 12). The best tuple (a, b) w.r.t. $\mathcal{A}_{ab}$, $\mathcal{A}'_{ab}$, $\mathcal{F}_{ab}$ and $\mathcal{Z}_{ab}$ is (10,10), (12,12), (12,12) and (12,12), respectively. Thus it is worthwhile to increase the control parameters in order to get better schedules.

Table 15: RPMIN / BPSEQ results for one instance

(a, b)	$\mathcal{A}_{ab}$	$\mathcal{A}'_{ab}$	$\mathcal{F}_{ab}$	$\mathcal{Z}_{ab}$
(2,2)	0.000	0.034	0.923	∞
(4,4)	0.000	0.155	0.938	∞
(6,6)	0.000	0.275	0.944	∞
(8,8)	0.003	0.346	0.949	826
(10, 10)	0.007	0.391	0.951	843
(12, 12)	0.002	0.435	0.952	847

Note that it takes about four times the CPU-time to solve medium-sized instances in comparison to small-sized ones. This moderate increase of computational requirements presumes the methods provided in this paper to be suitable for solving medium-to-large problem instances as well. By the way we should keep in mind that (as argued in literature: cf., e.g., **Alvarez-Valdés, Tamarit** (1989), **Badiru** (1988), and **Davis, Paterson** (1975)) the efficiency of heuristic methods in terms of performance is supposed to remain largely unaltered when problem size increases.

6. Summary and Conclusions

In this paper we propose a new model for academic course scheduling which addresses most of the items which are relevant for applications: Lectures of different length; precedence relations; availability of rooms of different size and equipment; changeover times between rooms belonging to different zones; workload constraints, i.e. upper bounds on the maximum number of lectures per day for a specific subject; preferences of professors for rooms and/or hours. In addition, we present heuristics which are essentially based on three ideas: First, prevent blocking of lectures, rooms and/or time-slots; second, look at the worst-case consequence of local decisions; third, perform regret based biased random sampling. We provide an instance generator for the generation of a representative set of instances. The

generator along with a statistical model is used for a thorough experimental evaluation of the methods. Computational results show that the methods solve medium-sized instances to suboptimality.

In the future improved (local search) methods should be developed. Amongst others, the following meta heuristics should be considered: First, simulated annealing methods (cf., e.g., **Johnson, Aragon, McGeoch, Schevon** (1989, 1991), **van Laarhoven, Aarts, Lenstra** (1992)), second, genetic algorithms, (cf., e.g., **Dorndorf, Pesch** (1992), **Goldberg** (1989), **Mühlenbein, Gorges-Schleuter, Krämer** (1988)), and third, tabu search methods (cf., e.g., **Faigle, Kern** (1992), **Glover** (1989, 1990)).

References

Abramson, D., "Constructiong school timetables using simulated annealing: Sequential and parallel algorithms", *Management Science,* Vol. 37 (1991), pp. 98-113.

Alvarez-Valdés, R., Tamarit, J.M., "Heuristic algorithms for resource-constrained project scheduling: a review and an empirical analysis", in: **Słowiński, R., Węglarz, J.** (eds.), *"Advances in project scheduling",* Amsterdam 1989, pp. 113-134.

Aubin, J., Ferland, J.A., "A large scale timetabling problem", *Computers and Operations Research,* Vol. 16 (1989), pp. 67-77.

Badiru, A.B., "Towards the standardization of performance measures for project scheduling heuristics", *IEEE Transactions on Engineering Management,* Vol. 35 (1988), pp. 82-89.

Balakrishnan, N., Lucena, A., Wong, R.T., "Scheduling examinations to reduce second-order consflicts", *Computers and Operations Research,* Vol. 19 (1992), pp. 353-361.

Bettin, M., "A heuristic algorithm for constructing university timetables on microcomputers", in: OR Models on Microcomputers, **Coelho, J.D. and L.V. Tavares** (eds.), Elsevier, Amsterdam 1986, pp. 105-110.

Cangalovic, M., Schreuder, J.A.M., "Exact colouring algorithm for weighted graphs applied to timetabling problems with lectures of different lengths", *European J. of Operational Research,* Vol. 51 (1991), pp. 248-258.

Carter, M.W., "A survey of practical applications of examination timetabling algo rithms", *Operations Research,* Vol. 34 (1986), pp. 193-202.

Davis, E.W., Paterson, J.H., "A comparison of heuristic and optimum solutions in resource-constrained project scheduling", *Management Science,* Vol. 21 (1975), pp. 944-955.

Dinkel, J.J., Mote, J., Venkataramanan, M.A., "An efficient decision support system for academic course scheduling", *Operations Research,* Vol. 37 (1989), pp. 853-864.

Dorndorf, U., Pesch, E., "Evolution based learning in a job shop scheduling environment", Working Paper, University Limburg/Maastricht 1992 (*Computers and Operations Research,* to appear).

Drexl, A., "Scheduling of project networks by job assignment", *Management Science,* Vol. 37 (1991), pp. 1590-1602.

Dyer, J.S., Mulvey, J.M., "An integrated optimization/information system for academic departmental planning", *Management Science,* Vol. 22 (1976), pp. 1332-1341.

Faigle, U., Kern, W., "Some convergence results for probabilistic tabu search", *ORSA J. on Computing,* Vol. 4 (1992), pp. 32-37.

Ferland, J.A., Roy, S., "Timetabling problem for university as assignment of activities to resources", *Computers and Operations Research,* Vol. 12 (1985), pp. 207-218.

Garey, M.R., Johnson, D.S.: *"Computers and intractability: a guide to the theory of NP-completeness".* Freeman, San Francisco 1979.

Glover, F., "Tabu search - Part I", *ORSA J. on Computing,* Vol. 1 (1989), pp. 190-206.

Glover, F., "Tabu search - Part II", *ORSA J. on Computing,* Vol. 2 (1990), pp. 4-32.

Goldberg, D.E., *"Genetic algorithms in search, optimization, and machine learning".* Reading/Mass. 1989.

Harwood, G.B., Lawless, R.W., "Optimizing organizational goals in assigning faculty teaching schedules", *Decision Sciences,* Vol. 6 (1975), pp. 513-524.

Heinrich, H., "Ein heuristisches Lösungsverfahren zur Stundenplanung und Raumbelegung an Hochschulen", *Angewandte Informatik,* Vol. 26 (1984), pp. 379-388.

Hertz, A., "Tabu search for large scale timetabling problems", *European J. of Operational Research,* Vol. 54 (1991), pp. 39-47.

Hertz, A., "Finding a feasible course schedule using tabu search", *Discrete Applied Mathematics,* Vol. 35 (1992), pp. 255-270.

Johnson, D.S., Aragon, C.R., McGeoch, L.A., Schevon, C., "Optimization by simulated annealing: an experimental evaluation; Part I, graph partitioning", *Operations Research,* Vol. 37 (1989), pp. 865-892.

Johnson, D.S., Aragon, C.R., McGeoch, L.A., Schevon, C., "Optimization by simulated annealing: an experimental evaluation; Part II, graph coloring and number partitioning", *Operations Research,* Vol. 39 (1991), pp. 378-406.

Kiaer, L., Yellen, J., "Weighted graphs and university course timetabling", *Computers and Operations Research,* Vol. 19 (1992), pp. 59-67.

van Laarhoven, P.J.M., Aarts, E.H.L., Lenstra, J.K., "Job shop scheduling by simulated annealing", *Operations Research,* Vol. 40 (1992), pp. 113-125.

McClure, R.H., Wells, C.E., "A mathematical programming model for faculty course assignments", *Decision Sciences,* Vol. 15 (1984), pp. 409-420.

Mühlenbein, H., Gorges-Schleuter, M., Krämer, O., "Evolution algorithms in combinatorial optimization", *Parallel Computing,* Vol. 7 (1988), pp. 65-85.

Mulvey, J.M., "A classroom/time assignment model", *European J. of Operational Research,* Vol. 9 (1982), pp. 64-70.

Salewski, F., Schirmer, A., Drexl, A., "Project scheduling under resource- and mode identity constraints", Working Paper, University Kiel 1994 (in preparation).

Shih, W., Sullivan, J.A., "Dynamic course scheduling for college faculty via zero-one programming", *Decision Sciences,* Vol. 8 (1977), pp. 711-721.

Tripathy, A., "A Lagrangean relaxation approach to course timetabling", *J. of* the *Operational Research* Society, Vol. 31 (1980), pp. 599-603.

de Werra, D., "An introduction to timetabling", *European J. of Operational Research,* Vol. 19 (1985), pp. 151-162.