Kolisch, Rainer; Sprecher, Arno

PSPLIB - a project scheduling problem library

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 396

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

Manuskripte
aus den
Instituten für Betriebswirtschaftslehre
der Universität Kiel

No. 396

PSPLIB - A project scheduling problem library

Rainer Kolisch and Arno Sprecher

Manuskripte

aus den

Instituten für Betriebswirtschaftslehre

der Universität Kiel

No. 396

PSPLIB - A project scheduling problem library

Rainer Kolisch and Arno Sprecher

March 1996

Rainer Kolisch and Arno Sprecher

Institut für Betriebswirtschaftslehre,

Christian-Albrechts-Universität zu Kiel,

Olshausenstraße 40, 24098 Kiel, Germany.

Abstract

We present a set of benchmark instances for the evaluation of solution procedures for single- and multi-mode resource-constrained project scheduling problems. The instances have been systematically generated by the standard project generator ProGen. They are characterized by the input-parameters of ProGen. The entire benchmark set including its detailed characterization and the best solutions known so-far are available on a public ftp-site. Hence, researchers can download the benchmark sets they need for the evaluation of their algorithms. Additionally, they can make available new results. Depending on the progress made in the field, the instance library will be continously enlarged and new results will be made accessible. This should be a valuable and driving source for further improvements in the area of project type scheduling.

0 General Information

Contact address: Rainer Kolisch / Arno Sprecher, Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel, Olshausenstraße 40, 24098 Kiel, Germany.

The project generator ProGen, a detailed description of ProGen [17], the instance-sets, their characterizations as well as the optimal or presently best known objective function values are available on the ftp-site **ftp.bwl.uni-kiel.de** under the path **/pub/operations-research/progen.** New results, comments, and questions can be communicated to the e-mail address **progen@bwl.uni-kiel.de.**

1 Introduction

Whereas the standard methods of project scheduling, CPM and MPM, base on the assumption of unlimited capacity of resources, modern approaches include the more realistic limitation of the resources' availabilities.

Consequently, numerous publications have dealt with exact and heuristic methods for solving the so-called single-mode resource-constrained project scheduling problem (SMRCPSP) where each of the activities of the project has to be performed in one prescribed way (mode) using specified amounts of the resources provided. The most common objective of the SMRCPSP is the minimization of the makespan (cf., e.g., [3, 4, 5, 24, 32]).

Recent developments have incorporated more reality by allowing the activities to be executed in one out of several modes. The modes reflect alternative combinations of resources and belonging quantities employed to fulfill the tasks related to the activities. The activity duration is a discrete function of the employed quantities, that is, using this concept, e.g., an activity can be accelerated

by raising the quantities coming into operation (time-resource-tradeoff). Moreover, by raising the required quantities of some resources while reducing the required quantities of others the resource substitution (resource-resource-tradeoff) can be realized. The problem at hand is the multi-mode resource-constrained project scheduling problem (MMRCPSP) which is commonly considered with makespan minimization as objective too (cf. [22, 23, 28, 29, 31, 34]).

Using the categorization scheme proposed by Slowinski (cf. [26, 27]) and Weglarz (cf. [35, 36]) three categories of resources required for the execution of a project are distinguished. Namely, renewable, nonrenewable, and doubly constrained resources.

Renewable resources are available on a period-by-period basis, that is, the quantities available are renewed from period to period (hour, day, week, month). The per-period availability is assumed as constant. E.g., manpower, machines, fuelflow and space are renewable resources.

In contrast to the renewable resources, nonrenewable ones are limited on a total project basis, that is, instead of the limited per-period usage of the renewable resources we have a limited overall consumption of the nonrenewable resources for the entire project. Money, energy and raw material belong to this category.

Resources which are limited on total project basis as well as on per-period basis are called doubly constrained. Money represents a resource of this category if beside the project's budget the per-period cashflow is limited. Manpower can be a doubly constrained resource, too, if for example a skilled worker can spend only a limited number of periods on the project. Clearly, since the doubly constrained resources can easily be taken into account by appropriately enlarging the sets of renewable and nonrenewble resources, respectively, they do not have to be considered explicitly.

The remaining of the paper is organized as follows: Section 2 describes the resource-constrained project scheduling problem in detail and presents an integer programming formulation, Section 3 introduces the parameters used for characterizing the instances generated. Section 4 provides the characterization of the benchmark sets and the nomenclature. Section 5 summarizes research performed on the instance sets. Finally, Section 6 specifies how new benchmark results can be made available to the research community.

2 The Model

We consider a project which consists of J activities (jobs, tasks). Due to technological requirements precedence relations between some of the activities enforce that an activity j, $j = 2, \ldots, J$, may not be started before all its predecessors h, $h \in \mathcal{P}_j$, are finished. The structure of the project is depicted by

a so-called activity-on-node (AON) network where the nodes represent the activities and the arcs the precedence relations. The network is acyclic and numerically labelled, that is, an activity has always a higher label than all its predecessors. W.o.l.o.g. activity 1 is the only start activity (source) and activity J is the only finish activity (sink). Both have a single mode with zero duration and resource request; they are dummy activities.

The activities j, $j = 1, \ldots, J$, have to be executed in one out of M_j modes. The activities may not be preempted and a mode once selected may not change, i.e., an activity j once started in mode m has to be completed in mode m without interruption. Performing activity j in mode m takes d_{jm} periods and is supported by a set R and N of renewable and nonrenewable resources, respectively. Considering a horizon, that is, an upper bound $\overline{T}$ on the project's makespan, K_r^ρ units of renewable resource r, $r \in R$, are available in period t, $t = 1, \ldots, \overline{T}$. The overall capacity of the nonrenewable resource r, $r \in N$, is given by K_r^ν. If activity j is scheduled in mode m, then k_{jmr}^ρ units of the renewable resource r, $r \in R$, are used each period activity j is in process. Additionally, k_{jmr}^ν units of the nonrenewable resource r, $r \in N$, are consumed. The parameters are summarized in Table 1 and assumed as integer-valued.

J	:	number of activities,
M_j	:	number of modes activity j can be performed in,
d_{jm}	:	duration of activity j being performed in mode m,
R (N)	:	set of renewable (nonrenewable) resources,
$\overline{T}$	:	upper bound on the project's makespan,
$K_r^\rho \geq 0$	:	number of units of renewable resource r, $r \in R$, available in period t, $t = 1, \ldots, \overline{T}$,
$K_r^\nu \geq 0$	:	total number of units available of nonrenewable resource r, $r \in N$,
$k_{jmr}^\rho \geq 0$	:	number of units of renewable resource r, $r \in R$, used by activity j being performed in mode m each period the activity is in process,
$k_{jmr}^\nu \geq 0$	:	number of units of nonrenewable resource r, $r \in N$, consumed by activity j being performed in mode m,
$\mathcal{P}_j$ ($\mathcal{S}_j$)	:	set of immediate predecessors (successors) of activity j,
ES_j (EF_j)	:	earliest start time (finish time) of activity j, calculated by using minimal activity durations and neglecting resource usage (consumption),
LS_j (LF_j)	:	latest start time (finish time) of activity j, calculated by using minimal activity durations, neglecting resource usage (consumption) and taking into account the upper bound $\overline{T}$ on the project's duration.

Table 1: Symbols and Definitions

The objective is to find a makespan minimal schedule that meets the constraints imposed by the precedence relations and the limited resource availabilities.

Due to the the constant per-period availability of the renewable resources, an upper bound $\overline{T}$ on the project's minimum makespan can be determined by the sum of the maximum activity durations. Given $\overline{T}$ we can use the precedence relations and the modes of shortest duration to calculate time windows, i.e., intervals $[EF_j, LF_j]$, with earliest finish times EF_j and latest finish times LF_j, containing the precedence feasible completion times of activity j, $j = 1, \ldots, J$, by traditional forward and backward recursion as performed by MPM.

With the time windows derived we can state the problem as a linear program as similarly presented by Talbot (cf. [34]). We use binary decision variables $x_{jmt}, j = 1, \ldots, J, m = 1, \ldots, M_j, t = EF_j, \ldots, LF_j$,

$$x_{jmt} = \begin{cases} 1 & , \text{ if activity } j \text{ is performed in mode } m \text{ and completed at the end of period } t \\ 0 & , \text{ otherwise.} \end{cases}$$

$$\text{Minimize } \Phi(x) = \sum_{m=1}^{M_J} \sum_{t=EF_J}^{LF_J} t \cdot x_{Jmt} \tag{1}$$

s.t.

$$\sum_{m=1}^{M_j} \sum_{t=EF_j}^{LF_j} x_{jmt} = 1 \qquad j = 1, \ldots, J \tag{2}$$

$$\sum_{m=1}^{M_h} \sum_{t=EF_h}^{LF_h} t \cdot x_{hmt} \leq \sum_{m=1}^{M_j} \sum_{t=EF_j}^{LF_j} (t - d_{jm}) x_{jmt} \qquad j = 2, \ldots, J, h \in \mathcal{P}_j \tag{3}$$

$$\sum_{j=1}^{J} \sum_{m=1}^{M_j} k_{jmr}^{\rho} \sum_{q=\max\{t, EF_j\}}^{\min\{t+d_{jm}-1, LF_j\}} x_{jmq} \leq K_r^{\rho} \qquad r \in R, t = 1, \ldots, \overline{T} \tag{4}$$

$$\sum_{j=1}^{J} \sum_{m=1}^{M_j} k_{jmr}^{\nu} \sum_{t=EF_j}^{LF_j} x_{jmt} \leq K_r^{\nu} \qquad r \in N \tag{5}$$

$$x_{jmt} \in \{0, 1\} \qquad j = 1, \ldots, J, m = 1, \ldots, M_j, \quad t = EF_j, \ldots, LF_j \tag{6}$$

Table 2: The Model of the MMRCPSP

The model is presented in Table 2 and is referred to as the multi-mode resource-constrained project scheduling problem (MMRCPSP).

Since there is exactly one finish activity, the objective function (1) realizes the minimization of the project's makespan. Constraints (2) ensure that exactly one mode and one completion time is assigned to each activity. The precedence relations are taken into account by (3). (4) guarantees, that the per-period availabilities of the renewable resources are not exceeded. Finally, (5) secures feasibility with respect to the consumable (nonrenewable) resources.

Obviously, given $M_j = 1$, $j = 1, \ldots, J$, and $|N| = 0$, the MMRCPSP degenerates to the single-mode resource-constrained project scheduling problem (SMRCPSP). Moreover, the well-known flow-shop, job-shop, and open-shop problem are included in the model outlined. Thus, the problem is a member of the class of NP-hard problems (cf. [12]). Furthermore, and if $|N| > 1$, then the feasibility problem (1) – (6) is already NP-complete (cf. [15]).

3 Project Characteristics

In this section we give a brief summary of the characteristics of the project instances, that is, the parameters of ProGen. A detailed description of the parameters and their realization can be found in [17, 18].

J^{min}	(J^{max})	minimum (maximum) number of non-dummy activities the project comprises,
M_j^{min}	(M_j^{max})	minimum (maximum) number of modes an activity j, $j = 2, \ldots, J-1$, can be performed in,
d^{min}	(d^{max})	minimum (maximum) duration of an activity j, $j = 2, \ldots, J-1$,
R^{min}	(R^{max})	minimum (maximum) number of renewable resources to be taken into account,
N^{min}	(N^{max})	minimum (maximum) number of nonrenewable resources to be taken into account,
$\mathcal{S}_1^{min}$	$(\mathcal{S}_1^{max})$	minimum (maximum) number of start activities,
$\mathcal{P}_J^{min}$	$(\mathcal{P}_J^{max})$	minimum (maximum) number of finish activities,
$\mathcal{S}_j^{max}$	$(\mathcal{P}_j^{max})$	maximum number of successors (predecessors) of an activity j, $j = 2, \ldots, J-1$,
NC		network complexity, i.e., the average number of non-redundant arcs per node including the dummy activities. The number of arcs actually incorporated into the network $ActArcs$ is controlled by ϵ_{NET}, the network complexity deviation tolerance, such that $ActArcs \in [J \cdot NC \cdot (1 - \epsilon_{NET}); J \cdot NC \cdot (1 + \epsilon_{NET})]$,

Q_τ^{min} (Q_τ^{max}) minimum (maximum) number of resources of category τ, $\tau \in \{R, N\}$, used and consumed, respectively, by an activity-mode combination $[j, m]$, $j = 2, \ldots, J-1$, $m = 1, \ldots, M_j$, i.e., minimum and maximum power of the sets

$Q_{jm}^\tau = \{[j, m, r]; k_{jmr} > 0, r \in \tau\}$, $j = 2, \ldots, J-1$, $m = 1, \ldots, M_j$,

U_τ^{min} (U_τ^{max}) minimum (maximum) level of per-period usage and total consumption, respectively, of resource r, $r \in \tau$, $\tau \in \{R, N\}$, by an activity-mode combination $[j, m]$ with $[j, m, r] \in Q_{jm}^\tau$, $j = 2, \ldots, J-1$, $m = 1, \ldots, M_j$,

P_1^τ (P_2^τ) probability that the level of per-period usage and total consumption, respectively, of a resource of category τ, $\tau \in \{R, N\}$, is duration constant (monotonically decreasing with the duration),

RF_τ resource factor of resources of category τ, $\tau \in \{R, N\}$. RF_τ reflects the average portion of the resources of category τ, $\tau \in \{R, N\}$, used and consumed, respectively. More precisely, the actual resource factor ARF_τ of a project instance is given by

$$ARF_\tau = \frac{1}{J-2} \sum_{j=2}^{J-2} \frac{1}{M_j} \frac{1}{|\tau|} \sum_{m=1}^{M_j} |Q_{jm}^\tau|$$

and it is controlled by ϵ_{RF}, the resource factor deviation tolerance, such that $ARF_\tau \in [(1 - \epsilon_{RF}) \cdot RF_\tau; (1 + \epsilon_{RF}) \cdot RF_\tau]$,

RS_τ resource strength of resources of category τ, $\tau \in \{R, N\}$. RS_τ measures the strength of the resource constraints of type τ. It is a scaling parameter expressing resource availability K_τ^τ as a convex combination of a minimum and maximum level K_τ^{min} and K_τ^{max}, $\tau \in \tau$, respectively. Using the function $round(\cdot)$, that rounds a real value to an integer, it is

$$K_\tau^\tau = K_\tau^{min} + \text{round}(RS_\tau \, (K_\tau^{max} - K_\tau^{min})).$$

For a nonrenewable resource r, $r \in N$, the minimum level K_τ^{min} and maximum level K_τ^{max} is obtained by cumulating the consumptions obtained when performing each activity in the mode having minimum and maximum consumptions, respectively, that is,

$$K_\tau^{min} = \sum_{j=2}^{J-1} \min_{m=1}^{M_j} \{k_{jmr}^\nu\} \quad , \quad K_\tau^{max} = \sum_{j=2}^{J-1} \max_{m=1}^{M_j} \{k_{jmr}^\nu\}.$$

For a renewable resources r, $r \in R$, the minimum level K_r^{min} is the lowest availability level allowing resource feasibility with respect to the considered resource, that is,

$$K_r^{min} = \max_{j=2}^{J-1} \min_{m=1}^{M_j} \{k_{jmr}^{\rho}\}$$

The maximum level K_r^{max} is determined via the resource dependant earliest start schedule obtained when performing the activities j, $j = 2, \ldots, J-1$, in the lowest indexed modes m_{jr}^* having maximum per-period usage of the considered resource, that is,

$$m_{jr}^* = \min\{m \in \{1, \ldots, M_j\}; k_{jmr}^{\rho} = \max_{m=1}^{M_j} \{k_{jmr}^{\rho}\}\}$$

K_r^{max} is determined by the peak per-period usage of resource r in the resource dependant earliest start schedule.

4 Characterization of the Benchmark Instances

In this section we present the parameter settings used for generating the benchmark instances. Currently, 2 benchmark sets are avaible for the SMRCPSP and 25 benchmark sets for the MMRCPSP. We group the input parameters given in Section 3 into three classes: First, fixed parameters which are constant for all benchmark sets, second, base parameters mainly one of which is adjusted individually for each benchmark set, and third, variable parameters which are systematically varied within each benchmark set. Table 3 gives the fixed parameters.

P_1^R	P_2^R	P_1^N	P_2^N	ϵ_{NET}	ϵ_{RF}
0.00	1.00	0.00	1.00	0.05	0.05

Table 3: Fixed Parameter Setting - SMRCPSP and MMRCPSP

	J	M_j	d_j	$\|R\|$	U_R	Q_R	$\|N\|$	U_N	Q_N	$\mathcal{S}_1$	$\mathcal{S}_j$	$\mathcal{P}_J$	$\mathcal{P}_j$
min	30	1	1	4	1	1	0	0	0	3	1	3	1
max	30	1	10	4	10	2	0	0	0	3	3	3	3

Table 4: Base Parameter Setting - SMRCPSP

The instances for the SMRCPSP have been generated with the fixed, base, and variable parameter settings given in Table 3, 4, and 5, respectively. Utilizing a full factorial design of the variable

Parameter	Levels			
NC	1.50	1.80	2.10	
RF_R	0.25	0.50	0.75	1.00
RS_R	0.20	0.50	0.70	1.00

Table 5: Variable Parameter Settings – SMRCPSP

parameters NC, RF_R, and RS_R with 10 replications per cell we have generated a total of 3·4·4·10 = 480 benchmark problems for each set. Table 6 provides a summary of the instances produced.

Instance Set	P	E	Type	Varied Base Parameter Table 4	Variable Setting	Number of Instances	Solution obtained by
J30	[1..48]	[1..10]	SM	$J^{min} = J^{max} = 30$	Table 5	480	opt.[6]
J60	[1..48]	[1..10]	SM	$J^{min} = J^{max} = 60$	Table 5	480	hrs.[16]

Table 6: Instance Sets - SMRCPSP

The 1-st column (instance set) gives the prefix of the file names the instances are stored under, the 2-nd column (P) displays the range of the cell index, reflecting the combination of the variable parameters. The 3-rd column (E) specifies the range of the instance index within a cell. The 4-th column abbreviates the acronym SMRCPSP to SM and serves as the suffix of the filenames. The 5-th column shows the varied base parameters, here the number of activities which has been set to 30 and 60, respectively. The 6-th column references to the table with the variable parameter levels employed. The 7-th column displays the number of instances within the benchmark set. Finally, the 8-th column shows how solutions of the benchmark sets have been obtained. A complete file name, e.g., J3012_10.SM, corresponds to instance set J30, variable parameter combination 12, and problem number 10 of the SMRCPSP. The level of the variable paramter settings for each parameter cell index can be found in the files J30PAR.SM and J60PAR.SM, respectively. Optimal objective function values for the instances of the J30 benchmark set have been obtained by [6] and are documented in the file J30OPT.SM. Currently, the instance set J60 cannot be solved by exact solution procedures. Hence, the best objective function values known so far have been computed with the heuristic of [16]. They can be found in the file J60HRS.SM.

Note, originally in [17] the instances J301_1.SM,...,J3048_10.SM were named J171_1.DAT,..., J1764_10.DA

and now have been renamed in the library for purpose of standardization.

The instance sets for the MMRCPSP are displayed in Table 9. They have been generated with the fixed, base, and variable parameter settings given in Table 3, 7, and 8, respectively. Note, the slight corruption in denoting the network complexity NC within the base parameter setting of Table 7.

	J	M_j	d_j	$\|R\|$	U_R	Q_R	$\|N\|$	U_N	Q_N	$\mathcal{S}_1$	$\mathcal{S}_j$	$\mathcal{P}_J$	$\mathcal{P}_j$	NC
min	16	3	1	2	1	1	2	1	1	3	1	3	1	1.8
max	16	3	10	2	10	2	2	10	2	3	3	3	3	1.8

Table 7: Base Parameter Setting – MMRCPSP

	A Levels				B Levels				C Levels			
RF_R	0.50	1.00			0.50	1.00			0.50	1.00		
RS_R	0.20	0.50	0.70	1.00	0.25	0.50	0.75	1.00	0.25	0.500	0.75	1.00
RF_N	0.50	1.00			0.50	1.00						
RS_N	0.20	0.50	0.70	1.00	0.25	0.50	0.75	1.00				

Table 8: Variable Parameter Settings – MMRCPSP

As for the single-mode case, we have generated 10 instances per cell defined by the variable parameter setting. Moreover, we have varied several base parameters as given in the 5-th column of Table 9. For technical reason, beside the base parameter varied, minor adaptations of depending base parameters have been necessary to generate the instance set R1 to R5 and N0 to N3, respectively. More precisely, if required, we have adapted $|R|^{min}$, $|R|^{max}$, $|N|^{min}$, $|N|^{max}$, Q_R^{min}, Q_R^{max}, Q_N^{min}, Q_N^{max}, U_N^{min}, U_N^{max}, RF_N, and RS_N. Note, in accordance with the systematic giving of names the instance set J16 could also be named M3, C18, R2, and N2.

Again, the variable paramter combination related to a cell can be found in the files XYZPAR.MM, e.g., J10PAR.MM. All the benchmark sets but J30 have been optimally solved with the branch-and-bound procedure presented in [30, 31]. The objective function values are available in the corresponding XYZOPT.MM files, e.g. J10OPT.MM. The instance set J30 has been heuristically solved by the truncated branch-and-bound algorithm of [30, 31] with an allotted CPU-time of 60 seconds and by the local search method of [15], respectively. The best objective function values are documented in the file J30HRS.MM.

Note, once more, for standardizational purpose we have renamed the files originally presented in [17] from MM1_1.DAT,..., MM64_10.DAT to J101_1.MM,..., J1064_10.MM.

Instance-Set	P	E	Type	Varied Base Parameter Table 7	Variable Setting	Number of Instances	Solutions obtained by
J10	[1..64]	[1..10]	MM	$J^{min} = J^{max} = 10$	Table 8, A	536	opt.[31]
J12	[1..64]	[1..10]	MM	$J^{min} = J^{max} = 12$	Table 8, B	547	opt.[31]
J14	[1..64]	[1..10]	MM	$J^{min} = J^{max} = 14$	Table 8, B	551	opt.[31]
J16	[1..64]	[1..10]	MM	$J^{min} = J^{max} = 16$	Table 8, B	550	opt.[31]
J18	[1..64]	[1..10]	MM	$J^{min} = J^{max} = 18$	Table 8, B	552	opt.[31]
J20	[1..64]	[1..10]	MM	$J^{min} = J^{max} = 20$	Table 8, B	554	opt.[31]
J30	[1..64]	[1..10]	MM	$J^{min} = J^{max} = 30$	Table 8, B	640	hrs.[15, 31]
M1	[1..64]	[1..10]	MM	$M_j^{min} = M_j^{max} = 1$	Table 8, B	640	opt.[31]
M2	[1..64]	[1..10]	MM	$M_j^{min} = M_j^{max} = 2$	Table 8, B	551	opt.[31]
M4	[1..64]	[1..10]	MM	$M_j^{min} = M_j^{max} = 4$	Table 8, B	555	opt.[31]
M5	[1..64]	[1..10]	MM	$M_j^{min} = M_j^{max} = 5$	Table 8, B	558	opt.[31]
C15	[1..64]	[1..10]	MM	$NC = 1.5$	Table 8, B	551	opt.[31]
C21	[1..64]	[1..10]	MM	$NC = 2.1$	Table 8, B	552	opt.[31]
R1	[1..64]	[1..10]	MM	$R^{min} = R^{max} = 1$	Table 8, B	553	opt.[31]
R3	[1..64]	[1..10]	MM	$R^{min} = R^{max} = 3$	Table 8, B	557	opt.[31]
R4	[1..64]	[1..10]	MM	$R^{min} = R^{max} = 4$	Table 8, B	552	opt.[31]
R5	[1..64]	[1..10]	MM	$R^{min} = R^{max} = 5$	Table 8, B	546	opt.[31]
N0	[1..8]	[1..10]	MM	$J^{min} = J^{max} = 10$	Table 8, C	75	opt.[31]
N0	[9..16]	[1..10]	MM	$J^{min} = J^{max} = 12$	Table 8, C	77	opt.[31]
N0	[17..24]	[1..10]	MM	$J^{min} = J^{max} = 14$	Table 8, C	79	opt.[31]
N0	[25..32]	[1..10]	MM	$J^{min} = J^{max} = 16$	Table 8, C	79	opt.[31]
N0	[33..40]	[1..10]	MM	$J^{min} = J^{max} = 18$	Table 8, C	80	opt.[31]
N0	[41..48]	[1..10]	MM	$J^{min} = J^{max} = 20$	Table 8, C	79	opt.[31]
N1	[1..64]	[1..10]	MM	$N^{min} = N^{max} = 1$	Table 8, B	637	opt.[31]
N3	[1..64]	[1..10]	MM	$N^{min} = N^{max} = 3$	Table 8, B	600	opt.[31]

Table 9: Instance Sets - MMRCPSP

Contrary to the single-mode case, due to mode-coupling via resource constraints, some of the multi-mode instances do not have a feasible solution (cf. [18]). Infeasible instances detected so far have been removed from the instance sets.

5 Use of the Instance Sets and State-of-the-Art Results

Since the presentation of the project generator ProGen, the instances produced for its evaluation, i.e. J30[1..48]_[1..10].SM and J10[1..64]_[1..10].MM, as well as additionally generated problem sets have been used in numerous publications. In the following we give a brief summary.

5.1 Single Mode Instances

Kolisch et al. [17, 18] solved the instance set J30 with the exact solution procedure of Demeulemeester and Herroelen (cf. [5]) for studying the influence of the variation of project characteristics, like the number of activities J, the number of renewable resources $|R|$, the resource factor RF_R , the resource strength RS_R, and the network complexity NC, on the computation time of the exact branch-and-bound procedure. As to be expected, solution times have been positively correlated with the number of jobs and the number of renewable resources. Moreover, a negative correlation of the CPU-time and the resource strength RS_R as well as a positive correlation of the CPU-time and the resource factor RF_R have been detected. A negative correlation between the network complexity NC and the CPU-time has not been significant. Numerous of the ProGen instances have not been solved to optimality within 3600 CPU-seconds on a personal computer (80386sx processor, 15MHz clockpulse). On the other hand, the Patterson benchmark set (cf. [21]), though having nearly the same size as the ProGen instances, has been solved to optimality in considerably less average CPU-time on the same computer.

In [19] Mingozzi et al. have used the instance set J30 for testing a recently developed branch-and-bound approach and new bounds. They claim that their algorithm performs better than the one of Demeulemeester and Herroelen (cf. [5]), especially when trying to solve the hard instances which could not be solved by Demeulemeester and Herroelen within the allotted time of 3600 CPU-seconds.

The procedure currently state-of-the-art is the revised and enhanced branch-and-bound procedure of Demeulemeester and Herroelen (cf. [6]). The new version improves its predecessor by the additional implementation of a variant of the Mingozzi et al. bound. Moreover it exploits the 32 bit architecture of an IBM PS/2 Model P75 (80486 processor, 25 MHz clockpulse, 32MB memory) operating under Windows NT. The entire set of instances J30 has been solved for the first time. Using 24 MB of data memory the computation time is about 33.68 seconds on average.

Kolisch [13] has performed a rigorous experimental investigation of the two basic heuristic scheduling strategies, serial and parallel scheduling, employed in a single-pass as well as in a (biased) random sampling approach. From the instance set J30 he has used those instances, which are resource-constrained, i.e., $RS_R < 1$, and the optimal solutions of which are known from the analysis in [17]. Kolisch has found out that the performance-ranking of priority rules does not differ for single-pass scheduling and sampling, that sampling improves the performance of single-pass scheduling, and that parallel scheduling is not superior in general. In [14] Kolisch has analyzed four new and four well-known priority rules for deterministic parallel scheduling on the subset of the benchmark set J30 described above. The newly developed worst case slack rule has provided the best results. The average deviation from the optimal objective function value has been 4.27 % compared to 4.83 % of LFT, the best classical rule. An adaptive search method for the RCPSP has been proposed by Kolisch and Drexl (cf. [16]) and again benchmarked on the specified subset. The procedure has achieved an average deviation from optimum of 0.71 %.

Naphade et al. (cf. [20]) introduced a local search heuristic for the RCPSP which builds up on ideas of Storer et al. (cf. [33]) for the job shop problem. They have benchmarked the approach on those problems of the instance set J30, which have been optimally solved in [17, 18]. The average deviation from the optimal objective function value has been 0.28 %.

Some scientists have used the project generator ProGen in order to generate project scheduling instances for their special needs. De Reyck and Herroelen (cf. [7]) utilized ProGen for creating 1980 assembly line balancing problems (ALB). They have assessed the efficiency of resource-constrained project scheduling techniques for solving ALB-type problems. Furthermore, the same authors, cf. [8], have analyzed the impact of the network structure on solution times. For experimental purposes they have generated 2500 instances. Icmeli and Erenguc (cf. [10]) have generated modified ProGen-instances in order to study the SMRCPSP with discounted cash flows. They have tested their exact branch-and-bound procedure which employs the bounding scheme devised in [5] and derives bounds by solving the resource unconstrained payment scheduling problem with the method given in [9]. Icmeli and Ron (cf. [11]) have created ProGen-instances for problems with relaxed integrality assumptions on the project's time line and activity durations, respectively. Solutions have been derived with the optimization package OSL.

Finally, Schwindt (cf. [25]) extended the project generator ProGen to ProGen/max capable of generating problem instances with minimal and maximal time lags between activities.

5.2 Multi Mode Instances

The multi-mode benchmark set J10 has been optimally solved by the basic version of the precedence tree guided enumeration scheme (cf. [22, 28]) and by the algorithms presented in [29, 30, 31]. The remaining multi-mode instance sets have been employed for the evaluation of the solution procedure presented in [30, 31]. The outlined algorithm currently provides the most powerful and general multi-mode approach. It is capable of solving the instance set J20 within an average CPU-time of less than four minutes on a personal computer (80486 processor, 66 MHz, 16 MB memory). Moreover, the related truncated method shows reasonable heuristic capabilities. It has solved all the instances of the set J10 to optimality and it has determined a feasible solution for 519 of the 640 instances of the set J30 within 60 CPU-seconds. The deviation of the makespan from the precedence-based lower bound averages at 30.84 %.

Kolisch and Drexl (cf. [15]) have solved the J10 and J30 multi-mode benchmark sets with a local search heuristic specificially developed to tackle problems with highly constrained nonrenwable resources. The procedure has derived feasible solutions for all problems of the instance set J10 and for 550 problems of J30 , respectively. The deviation from optimal solutions of J10 and precedence based lower bounds of J30 averages at 1.75 % and 21.01 %, respectively.

Additionally, the project generator ProGen has been used for producing instances for variants of the MMRCPSP. In [1, 2] Ahn and Erenguc have combined the MMRCPSP and the Time/Cost Trade-off Problem to the so-called multi-mode resource-constrained project scheduling problem with crashable modes, where a given mode duration can be reduced at some cost. The objective is the minimization of the project costs made of the sum of activity and penalty costs. Ahn and Erenguc proposed an exact solution procedure of the branch-and-bound type using some sort of LP-relaxation and an underestimation of the objective function. The procedure has been tested on 160 newly created instances with problem specific adaptions. The authors report that the algorithm outperforms an adapted version of [28].

6 Further Development of the Problem Library

The further extension of the problem library depends on the progress made in the development of heuristic and exact solution procedures. We plan to continuously extend the problem library to problems with characteristics similar to the ones already presented, but larger with respect to the number of activities, the number of modes, and the number of resources, respectively. Results obtained on the instances can be communicated to the research community as follows:

First, for the instance sets the optimal solutions of which are not known or verified so far, improved solutions can be send via e-mail with the subject "heuristicsolution". The format has to be as specified in Table 10. Note, since we will check feasibilty of the solutions automatically, it is necessary to meet the format exactly. The head of the file, line 1 through 4, has to be given once, the complete body, line 5 through 22, has to be repeated depending on the number of solutions suggested. A model file can be obtained by sending an e-mail with the subject "heuristicformat" to the address given in Section 0. For each instance set, e.g., J30 of type MM, which has not been entirely solved to optimality so far, a file with the best makespans known, here J30HRS.MM, is accessible and will be updated each month at the end of its final week.

```
================================================================
Authors' Name    :Rainer Kolisch / Arno Sprecher
Authors' Email   :progen@bwl.uni-kiel.de
================================================================
Instance Set     :XYZ
Type             :MM
Parameter Number:23
Instance Number :6
Makespan         :20
----------------------------------------------------------------
Solution
Job Mode Start Time
----------------------------------------------------------------
 1    1      0
 2    1      0
 3    2      0
 4    2      4
 5    1      8
 6    2      4
 7    1     12
 8    1     20
================================================================
```

Table 10: Format - Heuristic Solutions

Second, for instance sets entirely solved to optimality, the optimal makespans and CPU-times for all problems of the set can be send via e-mail with the subject "optimalsolution". Obviously, we cannot guarantee optimality of the makespans submitted. Therefore, the solutions are only accepted if a research report or a publication in a journal describing the solution procedure can be referenced and is commonly accessible. Again, a model file can be obtained by sending an e-mail with the subject "optimalformat" to the address provided in Section 0. The format of an optimal makespan file is specified in Table 11. The results will be made available without change using the instance set specifier followed by the type specifier and the initials of the author(s). The shelfmark is used as

extension. The example of Table 11 would produce file J30SMDH.95a.

Finally, e-mails send to the authors which are of common interest are made available in the file LATENEWS.

```
======================================================================
Authors' Name                 :E. Demeulemeester / W. Herroelen
Authors' Email                :uvw@test.uni-loaction.de
Authors' Initials [≤ 3 Signs]:DH
Shelfmark [≤ 3 Signs]         :95a
Instance Set                  :J30
Type                          :SM
Date                          :6/15/95
======================================================================
Research Report:  New benchmark results for the
                  resource-constrained project scheduling problem.
======================================================================
 Computer         :IBM PC PS/2 Model P75
 Processor        :80486
 Clockpulse       :25 MHz
 Operating System:Windows NT
 Memory Code      :110 KB
 Memory Data      :16 MB
 Language         :MS Visual C++
 Average CPU-time:33.68 sec.
======================================================================
Parameter Instance Makespan CPU-Time[sec.]
----------------------------------------------------------------------
 1         1        43       0.30
 1         2        47       0.11
 1         3        47       0.12
 1         4        62       0.64
 1         5        39       0.48
 .......
======================================================================
```

Table 11: Format - Optimal Solutions

References

[1] AHN, T. AND S. ERENGUC (1995): Resource constrained project scheduling problem with multiple crashable modes: An exact solution procedure. *Research Report*, Department of Decision and Information Sciences, University of Florida.

[2] AHN, T. AND S. ERENGUC (1995): Resource constrained project scheduling problem with multiple crashable modes: A heuristic solution procedure. *Research Report*, Department of Decision and Information Sciences, University of Florida.

[3] CHRISTOFIDES, N., R. ALVAREZ-VALDES, AND J.M. TAMARIT (1987): Project scheduling with resource constraints: A branch and bound approach. *European Journal of Operational Research*, Vol. 29, pp. 262-273.

[4] DAVIS, E.W. AND G.E. HEIDORN (1971): An algorithm for optimal project scheduling under multiple resource constraints. *Management Science*, Vol. 17, pp. B803-B816.

[5] DEMEULEMEESTER, E. AND W. HERROELEN (1992): A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *Management Science*, Vol. 38, No. 12, pp. 1803-1818.

[6] DEMEULEMEESTER, E. AND W. HERROELEN (1995): New benchmark results for the resource-constrained project scheduling problem. *Research Report* 9521, Department of Applied Economics, Katholieke Universiteit Leuven.

[7] DE REYCK, B. AND W. HERROELEN (1995): Assembly line balancing by resource-constrained project scheduling techniques - A critical appraisal. *Research Report* 9505, Department of Applied Economics, Katholieke Universiteit Leuven.

[8] DE REYCK, B. AND W. HERROELEN (1996): On the use of the complexity index as a measure of complexity in activity networks. *European Journal of Operational Research*, to appear.

[9] GRINOLD, R.C. (1972): The payment scheduling problem. *Naval Research Logistics Quarterly*, Vol. 19, pp. 123-136.

[10] ICMELI, O. AND S.S. ERENGUC (1995): A branch and bound procedure for the resource constrained project scheduling problem with discounted cash flows. *Research Report*, Department of Decision and Information Sciences, University of Florida.

[11] ICMELI, O. AND W.O. ROM (1995): Solving the resource constrained project scheduling problem with optimization subroutine library. *Research Report*, Departement of Operations Management and Business Statistics, Cleveland State University.

[12] GAREY, M.R. AND D.S. JOHNSON (1979): *Computers and intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, CA.

[13] KOLISCH, R. (1996): Serial and parallel resource-constrained project scheduling methods revisited - Theory and computation. *European Journal of Operational Research*, to appear.

[14] KOLISCH, R. (1996): Efficient priority rules for the resource constrained project scheduling problem. *Journal of Operations Management*, to appear.

[15] KOLISCH, R. AND A. DREXL (1994): Local search for nonpreemptive multi-mode resource-constrained project scheduling. *Research Report* 360, Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel.

[16] KOLISCH, R. AND A. DREXL (1996): Adaptive search for solving hard project scheduling problems. *Naval Research Logistics*, Vol. 43, pp. 23-40.

[17] KOLISCH, R., A. SPRECHER, AND A. DREXL (1992): Characterization and generation of a general class of resource-constrained project scheduling problems - Easy and hard instances. *Research Report* 301, Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel.

[18] KOLISCH, R., A. SPRECHER, AND A. DREXL (1995): Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science*, Vol. 41, pp. 1693-1703.

[19] MINGOZZI, A., V. MANIEZZO, S. RICCIARDELLI, AND L. BIANCO (1994): An exact algorithm for project scheduling with resource constraints based on a new mathematical formulation. *Research Report* 32, Department of Mathematics, University of Bologna.

[20] NAPHADE, K.S., S.D. WU, AND R.H. STORER (1995): Problem space search algorithms for the resource-constrained project scheduling problem. *Research Report*, Department of Industrial & Manufacturing Systems Engineering, Lehigh University.

[21] PATTERSON, J.H., (1984): A comparison of exact approaches for solving the multiple constrained resource, project scheduling problem. *Management Science*, Vol. 30, pp. 854-867.

[22] PATTERSON, J.H., R. SLOWINSKI, F.B. TALBOT, AND J. WEGLARZ (1989): An algorithm for a general class of precedence and resource constrained scheduling problems. In: Slowinski, R. and J. Weglarz (Eds.): *Advances in project scheduling.* Elsevier, Amsterdam, pp. 3-28.

[23] PATTERSON, J.H., R. SLOWINSKI, F.B. TALBOT, AND J. WEGLARZ (1990): Computational experience with a backtracking algorithm for solving a general class of precedence and resource-constrained scheduling problems. *European Journal of Operational Research*, Vol. 49, pp. 68-79.

[24] RADERMACHER, F.J. (1985/86): Scheduling of project networks. *Annals of Operations Research*, Vol. 4, pp. 227-252.

[25] SCHWINDT, C. (1995): ProGen/max: A new problem generator for different resource-constrained project planning problems with minimal and maximal time lags. *Research Report* WIOR-449, Universität Karlsruhe.

[26] SLOWINSKI, R. (1980): Two approaches to problems of resource allocation among project activities: A comparative study. *Journal of the Operational Research Society*, Vol. 31, pp. 711-723.

[27] SLOWINSKI, R. (1981): Multiobjective network scheduling with efficient use of renewable and nonrenewable resources. *European Journal of Operational Research*, Vol. 7, pp. 265-273.

[28] SPRECHER, A. (1994): *Resource-constrained project scheduling - Exact methods for the multi-mode case*. Springer, Berlin.

[29] SPRECHER, A., S. HARTMANN, AND A. DREXL (1995): Project scheduling with discrete time-resource and resource-resource tradeoffs. *Research Report* 357, Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel.

[30] SPRECHER, A. AND A. DREXL (1996): Solving Multi-Mode Resource-Constrained Project Scheduling Problems by a Simple, General and Powerful Sequencing Algorithm. Part I: Theory. *Research Report* 385, Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel.

[31] SPRECHER, A. AND A. DREXL (1996): Solving Multi-Mode Resource-Constrained Project Scheduling Problems by a Simple, General and Powerful Sequencing Algorithm. Part II: Computation. *Research Report* 386, Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel.

[32] STINSON, J.P., E.W. DAVIS, AND B.M. KHUMAWALA (1978): Multiple resource-constrained scheduling using branch and bound. *AIIE Transactions*, Vol. 10, pp. 252-259.

[33] STORER, R.H., S.D. WU, AND R. VACCARI (1992): New search spaces for sequencing problems with application to job shop scheduling. *Management Science*, Vol. 38, pp. 1495-1509.

[34] TALBOT, F.B. (1982): Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case. *Management Science*, Vol. 28, pp. 1197-1210.

[35] WEGLARZ, J. (1979): Project scheduling with discrete and continuous resources. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, pp. 644-650.

[36] WEGLARZ, J. (1980): On certain models of resource allocation problems. *Kybernetics*, Vol. 9, pp. 61-66.