

Schirmer, Andreas

**Working Paper — Digitized Version**

## 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, No. 390

**Provided in Cooperation with:**

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

*Suggested Citation:* Schirmer, Andreas (1996) : 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, No. 390, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/149839>

**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.*

Manuskripte  
aus den  
Instituten für Betriebswirtschaftslehre  
der Universität Kiel

No. 390

**New Insights on the  
Complexity of Resource-  
Constrained Project Scheduling -**

**A Case of Single-Mode Scheduling**

Schirmer



No. 390

**New Insights on the  
Complexity of Resource-  
Constrained Project Scheduling -  
A Case of Single-Mode Scheduling**

Schirmer

January 1996

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

Phone, Fax +49-431-880-15 31  
E-Mail [schirmer@bwl.uni-kiel.de](mailto:schirmer@bwl.uni-kiel.de)

## Contents

1. Introduction and Motivation .....	1
2. The Single-Mode Project Scheduling Problem (SMPSP).....	2
3. A PWW-Model Formulation of the SMPSP.....	4
4. The PWW-Model of the SMPSP is Exponential .....	5
5. Improving the Complexity .....	7
6. Summary and Outlook .....	14
Appendix.....	15
References.....	16

## Figures

3PP Transforms to $SMPSP_{PWW}^{feas}$ .....	9
Overview of Different Complexity Classes.....	14

## Tables

Problem Parameters of the SMPSP.....	3
Derived Parameters of the SMPSP .....	4

**Abstract:** Complexity proofs often restrict themselves to stating that the problem at hand is a generalization of some other intractable problem. This proof technique relies on the widely accepted assumptions that complexity results hold regardless of the model formulation used to represent the problem and the encoding used to represent its instances. However, recent results indicate that these assumptions are not always justified. Brüggemann (1995) showed that for at least one model formulation of the *discrete lotsizing and scheduling problem* (DLSP) and certain, pathological instances the corresponding decision problem cannot be a member of **NP**; therefore that DLSP-model is exponential. With respect to the field of resource-constrained project scheduling, the question arises whether here this or a similar situation may occur as well. We extend the findings of Brüggemann to the *single mode project scheduling problem* (SMPSP) where we prove the well-known binary programming model of the SMPSP to be exponential as well - regardless of the particular encoding used. In addition, we demonstrate that this result can be improved to a strongly **NP**-equivalence result by adding one moderate restriction on the problem parameters.

**Keywords:** RESOURCE-CONSTRAINED PROJECT SCHEDULING; COMPUTATIONAL COMPLEXITY

## 1. Introduction and Motivation

Complexity proofs often restrict themselves to stating that the problem at hand is a generalization of some other intractable problem; a complexity proof for project scheduling problems for example often consists of the hint that it generalizes the *job shop problem* (JSP) (Baker 1974; Drex1 1990; Sprecher 1994) which is known to be strongly **NP**-hard (Garey, Johnson, and Sethi 1976). This proof technique relies on the widely accepted assumptions that complexity results hold regardless of the model formulation used to represent the problem and regardless of the encoding used to represent its instances. (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.)

However, recent results indicate that these assumptions are not always justified. Brüggemann (1995) was able to prove for at least one model formulation of the *discrete lotsizing and scheduling problem* (DLSP) that a commonly used encoding which is widely accepted as being reasonable and concise may cease to remain concise when confronted with certain, pathological instances. As a consequence, different encodings may lead to categorially different input lengths such that complexity results can no longer be safely assumed to be encoding-independent. Further, Brüggemann showed that under a slightly different encoding, which is actually concise for these pathological instances, said model of the corresponding decision problem cannot be a member of **NP**; therefore the model is exponential. Finally, he demonstrated that for the DLSP different model formulations may entail different complexity results; so complexity results need not always hold regardless of the employed model formulation.

With respect to the field of resource-constrained project scheduling, the question arises whether here this or a similar situation may occur as well. We examine the standard practice of representing project scheduling problems under resource constraints, i.e. in terms of binary

optimization models. We extend the findings of Brüggemann to the *single mode project scheduling problem* (SMPSP). Since these findings suggest that different model formulations may give rise to different complexity results, it is first of all necessary to describe a specific model of the SMPSP. For this purpose, we formulate the SMPSP in terms of what we call a PWW-model, i.e. a binary optimization model following the approach by Pritsker, Watters, and Wolfe (1969) who used binary variables to represent schedules. These models are well established in literature as the standard practice of representing scheduling as well as many other problems. We prove that the PWW-model of the SMPSP is exponential - regardless of the particular encoding used. In addition, we demonstrate that this result can be improved to a strongly NP-equivalence result by adding one moderate restriction.

The remainder of this work is organized as follows: Section 2 briefly introduces the SMPSP for which Section 3 provides the mathematical formalization in terms of a PWW-model. Section 4 is devoted to the finding that this model of the SMPSP is exponential. In Section 5, shows how to turn the exponential model of the SMPSP into a strongly NP-equivalent one. Finally, Section 6 provides a brief summary and closes with an outlook on future research.

## 2. The Single-Mode Project Scheduling Problem (SMPSP)

The principal components of any *project scheduling problem* (PSP) are *activities*, *resources*, *precedence relations*, and a *quality measure*. A project consists of a number of activities; all activities have to be executed in order to complete the project. During their nonpreemptable execution the activities request resources. Within this conceptual framework, resources are generally classified according to their availability. The available amount of *nonrenewable* resources is limited over the whole planning horizon by a *total capacity*. In contrast, the available amount of *renewable* resources is limited in every period by a *period capacity*. Resources having both a total and a period capacity are called *doubly constrained*; however, as they can be split into a renewable and a nonrenewable part (cf. Slowinski 1981) they are usually not considered explicitly. Often, an activity can be performed in several different ways (*modes*), resulting in different durations and different resource demands. Accordingly, each mode of an activity is associated with a *duration*, one *resource usage* for each renewable resource, and one *resource consumption* for each nonrenewable resource. Further, technological, legal, or other reasons may induce precedence relations, meaning that some activities need to be completed before others can be started. Finally, once the execution of an activity has begun, it may not be interrupted (*nonpreemptive scheduling*).

A *schedule* assigns a finishing time and - if necessary - a mode to each activity. A *project scheduling problem* (PSP) then amounts to finding a schedule that is both feasible, i.e. respecting a given set of constraints, and optimal, i.e. attaining the optimum of a specified *quality measure*. The constraints may represent e.g. precedence relations between certain activities or

limited resource availabilities while the quality measure may aim at minimizing the total duration (makespan) of the project. The quality of such a schedule is measured in terms of a function mapping each schedule to a numerical value (for more details cf. Kolisch 1995).

In the following we consider the most simple PSP, viz. the classical single-mode project scheduling problem (SMPSP) which is characterized by the fact that each activity can be performed in only one specific mode and which aims exclusively at minimizing the project makespan. W.r.t. the nonrenewable resources this implies that their consumption is neither open to decision, nor does it form part of the quality measure. As a consequence, nonrenewable resources are not relevant in this case; only the way the renewable resources are used over the processing time of the project remains open to decision.

The problem parameters of the SMPSP are summarized (in alphabetical order) in Table 1.

**Table 1**  
*Problem Parameters of the SMPSP*

Problem Parameter	Definition
$d_j$	Duration of activity $j$
$J$	Number of activities, indexed by $j$
$k_{jr}$	Per-period usage of renewable resource $r$ required to perform activity $j$
$K_r$	Per-period availability of renewable resource $r$
$R$	Number of renewable resources, indexed by $r$
$\angle$	Partial order on the activities, representing precedence relations

W.l.o.g. the parameters  $J$ ,  $R$ , all  $d_j$  (except of  $d_1$  and  $d_J$ ), and all  $K_r$  are assumed to be positive integers while the  $k_{jr}$  are assumed to be nonnegative integers. This entails no loss of generality since it is equivalent to allowing rational numbers, i.e. fractions, and multiplying them with the smallest common multiple of their denominators.

The goal is to find an assignment of periods to activities (a *schedule*) that covers all activities, ensures for each renewable resource  $r$  that in each period the total usage of  $r$  by all activities performed in that period does not exceed the per-period availability of  $r$ , respects the partial order  $\angle$ , and minimizes the total project length.

### 3. A PWW-Model Formulation of the SMPSP

To simplify the formulation of the model, w.l.o.g. it is assumed that the (fictitious) activities 1 and  $J$  are *dummy activities*, meaning that their duration and resource requirements equal zero, and that activity 1 is the unique first and activity  $J$  the unique last activity w.r.t.  $\prec$ , i.e. that  $1 \prec j$  and  $j \prec J$  ( $2 \leq j \leq J-1$ ). Also, some parameters are derived from the above problem parameters. First, in order to restrict the number of periods to be considered, an upper bound for the makespan of the project needs to be determined. Note that a, however poor, bound can be derived from

$$T = \sum_{j=2}^{J-1} d_j \quad (1)$$

Second, let denote  $P_j$  ( $1 \leq j \leq J$ ) the set of all *immediate* predecessors of activity  $j$  w.r.t.  $\prec$ . Third, for each activity  $j$  ( $1 \leq j \leq J$ ) earliest finishing times  $EF_j$  and latest finishing times  $LF_j$  may be calculated. While not being a necessary prerequisite, they usually allow to reduce the number of binary variables needed to formulate a specific instance of the problem.

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

**Table 2**  
*Derived Parameters of the SMPSP*

Derived Parameter	Definition
$EF_j$	Earliest finishing time of activity $j$
$LF_j$	Latest finishing time of activity $j$
$P_j$	Set of all immediate predecessors of activity $j$
$T$	Number of periods (planning horizon), indexed by $t$

Using the above conceptualization, the assignment of activities to periods can be represented by binary variables  $x_{jt}$  ( $1 \leq j \leq J$ ;  $1 \leq t \leq T$ ) where

$$x_{jt} = \begin{cases} 1 & \text{if activity } j \text{ is completed in period } t \\ 0 & \text{otherwise} \end{cases} \quad (2)$$



Now the SMPSP can be expressed in terms of the following PWW-model:

Minimize

$$Z(x) = \sum_{t=EF_j}^{LF_j} t x_{jt} \quad (3)$$

subject to

$$\sum_{t=EF_j}^{LF_j} x_{jt} = 1 \quad (1 \leq j \leq J) \quad (4)$$

$$\sum_{t=EF_i}^{LF_i} t x_{it} \leq \sum_{t=EF_j}^{LF_j} (t - d_j) x_{jt} \quad (2 \leq j \leq J; i \in P_j) \quad (5)$$

$$\sum_{j=1}^J k_{jr} \sum_{\tau=t}^{t+d_j-1} x_{j\tau} \leq K_r \quad (1 \leq r \leq R; 1 \leq t \leq T) \quad (6)$$

$$x_{jt} \in \{0, 1\} \quad (1 \leq j \leq J; EF_j \leq t \leq LF_j) \quad (7)$$

This model formulation of the SMPSP can be explained as follows: Minimization of the objective function (3) enforces the earliest possible completion of the last activity J and thus leads to the minimal schedule length. The activity completion constraints (4) guarantee that each activity is executed once and completed within the interval [1, T]. The precedence constraints (5) represent the precedence order. The capacity constraints (6) limit the total resource usage of each renewable resource in each period to the available amount.

In the following,  $SMPSP_{PWW}$  is understood to denote the specific PWW-model formulation given above, rather than the general problem setting SMPSP as outlined in Section 2.

#### 4. The PWW-Model of the SMPSP is Exponential

We will now prove for the feasibility version  $SMPSP^{feas}$  of the SMPSP that its PWW-model  $SMPSP^{feas}_{PWW}$  cannot be verified in polynomial time. This implies that solving it in polynomial time is impossible, hence  $SMPSP_{PWW}$  in turn is exponential.

**Theorem 1**  $SMPSP^{feas}_{PWW}$  is not polynomially verifiable, i.e.  $SMPSP^{feas}_{PWW} \notin NP$ .

*Proof:* Let  $J \in \mathbb{N}$ ,  $d_1 = d_J = 0$ ,  $d_j \in \mathbb{N}$  ( $2 \leq j \leq J-1$ ),  $R \in \mathbb{N}$ ,  $k_{jr} \in \mathbb{N}_0$  ( $1 \leq j \leq J$ ;  $1 \leq r \leq R$ ),  $K_r \in \mathbb{N}$  ( $1 \leq r \leq R$ ), and  $\angle$  a partial order on  $\{1, \dots, J\}$  constitute an arbitrary instance  $I$  of  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$ . Letting denote  $K^{\text{max}} = \max\{K_r \mid 1 \leq r \leq R\}$ , the following bounds are straightforward (even though tighter bounds could be derived for the  $d_j$ , these will be sufficient for our purposes):

$$d_j \leq T \quad (1 \leq j \leq J) \quad (8)$$

$$k_{jr} \leq K_r \quad (1 \leq j \leq J; 1 \leq r \leq R) \quad (9)$$

$$K_r \leq K^{\text{max}} \quad (1 \leq r \leq R) \quad (10)$$

$$|\angle| \leq J^2 \quad (11)$$

$$[\text{EF}_j, \text{LF}_j] \subseteq [1, T] \quad (1 \leq j \leq J) \quad (12)$$

Hence, the magnitude of an instance is  $\text{MAX}_{\text{STD}}(I) = \max\{T, K^{\text{max}}\}$  and the input length is  $\text{LNG}_{\text{STD}}(I) \approx O(J^2 \cdot R \cdot \log \text{MAX}_{\text{STD}}(I))$ .

We now have to show that any certificate may require time exponential in  $\text{LNG}_{\text{STD}}(I)$  for verification. A certificate  $C$  of  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$  consists of one value for each of the  $J \cdot T$  decision variables. Since all variables are binary,  $\text{MAX}_{\text{STD}}(C) = 1$ . Hence,  $\text{LNG}_{\text{STD}}(C) = J \cdot T \cdot \lceil \log(1) + 1 \rceil = J \cdot T$  such that reading it has a time complexity of  $O(J \cdot T)$ . But  $J \cdot T$  is not polynomial in  $\text{LNG}_{\text{STD}}(I)$  since there is no way of polynomially bounding  $T$  in  $J^2 \cdot R \cdot \log \text{MAX}_{\text{STD}}(I)$ . This becomes especially apparent in the case  $\text{MAX}_{\text{STD}}(I) = T$ , implying  $\text{LNG}_{\text{STD}}(I) \approx O(J^2 \cdot R \cdot \log T)$ , since  $T$  is clearly exponential in  $\log T$ . But also in the opposite case, where  $\text{MAX}_{\text{STD}}(I) = K^{\text{max}}$ ,  $T$  remains exponential since all parameters  $d_j$ , which determine the value of  $T$  (cf. equation (1)), are free and may thus attain arbitrarily large values. As even reading a certificate will require exponential time, it is impossible to verify it in polynomial time such that  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$  can be no member of **NP**. ■

**Corollary 1**  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$  and  $\text{SMPSP}_{\text{PWW}}$  are exponential.

*Proof:* It is obvious from the proof of Theorem 1 that the length of any certificate  $C$  of  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$  cannot be polynomially bounded in the length of the corresponding instance  $I$  of  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$ . Therefore even writing down a solution for an  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$ -instance, to say nothing of constructing one, may in the worst case take exponential time:  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$  cannot be solved by a polynomial algorithm; so  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$  is exponential.

Further, as any solution of an  $\text{SMPSP}_{\text{PWW}}$ -instance also provides a solution for the corresponding  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$ -instance,  $\text{SMPSP}_{\text{PWW}}$  cannot be easier to solve than  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$ ; so also  $\text{SMPSP}_{\text{PWW}}$  is exponential. ■

We should emphasize that this result does not exclude the possibility that *some or even most* instances of the  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$  or the  $\text{SMPSP}_{\text{PWW}}$  may be solved in polynomial time. Since the definition of complexity aims at the worst case, it rather means that *not all* instances will be polynomially solvable. To illustrate this, consider the following pathological situation:

Let  $J = 3$ ,  $d_1 = d_J = 0$ ,  $d_2$  free,  $R = 1$ ,  $k_{jT} = 1$ ,  $K_T = 1$ , and  $\angle$  empty characterize a class of instances of  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$ . The length of these instances is essentially  $O(\log T)$  while the length of a certificate is  $O(T)$ . Obviously,  $T$  is exponential in  $\log T$  such that solving such instances requires exponential time.

From the proof of Theorem 1 it is clear that also more realistic instances with larger parameter values may share the property of requiring exponential effort. As long as  $T$  is free to take arbitrarily large values which can go beyond any polynomial bound in the length of the respective instance,  $\text{SMPSP}_{\text{PWW}}$ -instances may turn up which are only solvable in exponential time.

## 5. Improving the Complexity

The results of the previous section are quite discouraging with respect to the tractability of the  $\text{SMPSP}_{\text{PWW}}$ : Since a worst-case result alone does not allow to infer how often the worst case occurs, one does not know whether only a few or almost all instances will require exponential time for solving them. In order to gain further knowledge, one might try to identify distribution functions for at least some of the problem parameters. Such information might help to determine the structure of typical instances from which the frequency of worst-case results could be inferred. This approach clearly requires a detailed analysis of the specific planning situation at hand and thus cannot be employed for the  $\text{SMPSP}_{\text{PWW}}$  in general.

However, by adding a relatively lenient assumption a better result can be established which holds for all instances. The  $\text{SMPSP}_{\text{PWW}}$  under this additional condition can be shown to be strongly **NP**-equivalent, in other words of the same categorial complexity as the strongly **NP**-complete problems. This result is better than exponential since it allows for the possibility that  $\text{SMPSP}_{\text{PWW}}$  may be polynomially solvable, even if only under the precondition that  $\mathbf{P} = \mathbf{NP}$ .

The mentioned additional condition requires for each instance that all  $d_j$  ( $1 \leq j \leq J$ ) and thus  $T$  be limited to values which are polynomially bounded in the respective input length. It is noteworthy to mention that, although being quite restrictive from the formal point of view, this condition is not too severe in practical terms. The only instances which it precludes comprise arbitrarily large planning horizons, a situation which is extremely improbable to

occur in practice. In most applications, natural choices of the planning horizon will be fixed or at least sufficiently small to meet this condition. In fact, the proof of Theorem 1 indicates that when considering a sequence of essentially similar instances which only differ in their linearly growing values of  $T$ , the time required to solve them will in the worst case increase exponentially in the input length.

We will now show that under the above assumption  $\text{SMSPSP}^{\text{feas}}_{\text{PWW}}$  is strongly **NP**-complete. The proof essentially provides a pseudo-polynomial transformation, denoted by  $\alpha_p$ , from the strongly **NP**-complete *3-partition problem* (3PP) (Garey, Johnson 1975) to the  $\text{SMSPSP}^{\text{feas}}_{\text{PWW}}$ :

**(3PP)** Given  $N \in \mathbb{IN}$ ,  $S = \{1, \dots, 3 \cdot N\}$ ,  $B \in \mathbb{IN}$  (called a *bound*), and  $\mu_j \in \mathbb{IN}$  ( $1 \leq j \leq 3 \cdot N$ ) (called a *size* for each element of  $S$ ) such that

$$B/4 < \mu_j < B/2 \quad (1 \leq j \leq 3 \cdot L) \quad (13)$$

$$\text{and } \sum_{j \in S} \mu_j = L \cdot B \quad (14)$$

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

$$\sum_{j \in S_i} \mu_j = B \quad (1 \leq i \leq L) \quad (15)$$

holds true?

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

**Theorem 2**  $\text{SMSPSP}^{\text{feas}}_{\text{PWW}}$  - under the assumption that all  $d_j$  ( $1 \leq j \leq J$ ) are restricted to values which are polynomial in the input length - is strongly **NP**-complete.

*Proof:*

( $\text{SMSPSP}^{\text{feas}}_{\text{PWW}} \in \text{NP}$ ) By virtue of the above restriction,  $T$  is polynomial in  $\text{LNG}_{\text{STD}}(I)$  such that the magnitude of an instance  $I$  of the  $\text{SMSPSP}^{\text{feas}}_{\text{PWW}}$  is  $\text{MAX}_{\text{STD}}(I) = K^{\text{max}}$  and the input length is  $\text{LNG}_{\text{STD}}(I) \approx O(J^2 \cdot R \cdot \log K^{\text{max}})$ . We now have to show that any certificate can be read and verified in polynomial time. A certificate  $C$  of the  $\text{SMSPSP}^{\text{feas}}_{\text{PWW}}$  consists of one value for each of the  $J \cdot T$  decision variables; due to their binarity  $\text{MAX}_{\text{STD}}(C) = 1$  and thus  $\text{LNG}_{\text{STD}}(C) = J \cdot T \cdot (\log(1) + 1) = J \cdot T$ . Therefore, reading a certificate has a time complexity of  $O(J \cdot T)$  which is polynomial in the input length because  $T$  is so.

Evaluating the constraints will require at most  $T-1$  additions for the  $J$  activity completion constraints (4), at most  $3 \cdot T-2$  additions and at most  $2 \cdot T$  multiplications for the at

most  $J(J-1)/2$  precedence constraints (5), and  $J-1$  times at most  $T-1$  additions and  $J$  times at most  $T$  multiplications for the  $R \cdot T$  resource constraints (6). In total, this amounts to  $O(J^2 \cdot R \cdot T^2)$  operations; assuming that each addition and multiplication takes constant time, the time complexity of the total evaluation is  $O(J^2 \cdot R \cdot T^2)$ . So, it can be verified in polynomial time that the schedule satisfies the constraints (4) - (6).

( $3PP \propto_p \text{SMPSP}^{\text{feas}}_{PWW}$ ) Let  $L \in \mathbb{N}$ ,  $S = \{1, \dots, 3 \cdot L\}$ ,  $B \in \mathbb{N}$ ,  $\mu_j \in \mathbb{N}$  ( $1 \leq j \leq 3 \cdot L$ ) constitute an arbitrary instance of  $3PP$ . Then construct an instance of  $\text{SMPSP}^{\text{feas}}_{PWW}$  as follows:  $J = 3 \cdot L$ ,  $d_j = 1$  ( $1 \leq j \leq J$ ),  $R = 1$ ,  $k_j = \mu_j$  ( $1 \leq j \leq J$ ),  $K = B$ , and  $\angle$  empty, where due to  $R = 1$   $k_j$  stands for  $k_{j1}$  and  $K$  for  $K_1$  (cf. Figure 1). Also, let  $T = L$ . Even though applying (1) would yield  $T = 3 \cdot L$ , we will see below that here  $T = L$  is sufficient for building a feasible schedule.

**Figure 1**

*3PP Transforms to  $\text{SMPSP}^{\text{feas}}_{PWW}$*

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

In order to show that this transformation is indeed pseudo-polynomial, let us first argue that there exists a feasible schedule for the constructed instance of  $\text{SMPSP}^{\text{feas}}_{PWW}$  if and only if there is a partition of  $S$  as described above. Assuming that there exists a partition of  $S$  as described above, we can design a feasible schedule by scheduling all activities in  $S_i$  ( $1 \leq i \leq L$ ) to period  $i$ . Since the  $S_i$  form a partition of  $S$ , this procedure covers all activities, hence the activity completion constraints (4) are met. Due to  $\angle$  empty there are no precedence constraints (5). Finally, (15) translates to

$$\sum_{j \in S_i} k_j = K \quad (1 \leq i \leq T) \quad (16)$$

such that the schedule also satisfies the resource constraint (6). Conversely, if such a feasible schedule exists, a partition of  $S$  as described above can be obtained by putting together in  $S_i$  ( $1 \leq i \leq L$ ) all elements  $j$  ( $1 \leq j \leq 3 \cdot L$ ) scheduled to period  $i$ . Due to con-

straints (4) this yields a partition, and due to constraints (6) also the desired property (15) holds.

Second, the above transformation can be performed in pseudo-polynomial, even in polynomial time since each of the  $O(L)$  assignments can be done in  $O(\log B)$  time. As all values of the 3PP-instance form part of the derived instance, the third condition, viz. that the transformed instance will not be of categorially smaller length, will be met even with the identical polynomial. Finally, the maximum number occurring in the constructed instance,  $B$ , is also the maximum number in the 3PP-instance, such that it can be bounded in length and magnitude of the 3PP-instance by the polynomial  $p(x,y) = 0 \cdot x + y$ . So, the above transformation from 3PP to  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$  is indeed pseudo-polynomial which implies the strong **NP**-completeness of  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$ . ■

Consequentially,  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$  can be solved by a polynomial or by a pseudo-polynomial algorithm if and only if  $\mathbf{P} = \mathbf{NP}$  such that it is essentially of the same complexity as the strongly **NP**-complete problems. Building upon this result, it is easy to show that the  $\text{SMPSP}_{\text{PWW}}$  is strongly **NP**-hard by exhibiting a pseudo-polynomial reduction, denoted by  $\infty_{\text{PR}}$ , from the  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$  to the  $\text{SMPSP}_{\text{PWW}}$ .

**Theorem 3**  $\text{SMPSP}_{\text{PWW}}$  - under the assumption that all  $d_j$  ( $1 \leq j \leq J$ ) are restricted to values which are polynomial in the input length - is strongly **NP**-hard.

*Proof:*

- (i) ( $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$  is strongly **NP**-complete) See above.
- (ii) ( $\text{SMPSP}^{\text{feas}}_{\text{PWW}} \infty_{\text{PR}} \text{SMPSP}_{\text{PWW}}$ )

Assuming some hypothetical algorithm  $A'$  solving  $\text{SMPSP}_{\text{PWW}}$ , each instance  $I$  of  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$  can be solved by an algorithm  $A$  that proceeds in the following way: Construct from  $I$  an instance  $I'$  of  $\text{SMPSP}_{\text{PWW}}$  and solve it by  $A'$ . If  $A'$  returns an optimal solution for  $I'$ , then that solution is also feasible; accordingly the solution for  $I$  is "yes". If  $A'$  returns "no", then no optimal and thus no feasible solution for  $I'$  exists; hence the solution for  $I$  is "no" as well. This construction can be done in linear, thus in polynomial time such that  $A$  provides a polynomial reduction from  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$  to  $\text{SMPSP}_{\text{PWW}}$ .  $A$  is also pseudo-polynomial since all numbers occurring in any constructed instance  $I'$  of  $\text{SMPSP}_{\text{PWW}}$  also occur in the original instance  $I$  of  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$  such that the magnitude of  $I'$  can easily be bounded by the identical polynomial in the magnitude of  $I$ . ■

This result allows to characterize the  $\text{SMPSP}_{\text{PWW}}$  as at least as hard to solve as the strongly **NP**-complete problems. As a consequence, the  $\text{SMPSP}_{\text{PWW}}$  can neither be solved by a poly-

nomial nor by a pseudo-polynomial algorithm unless  $\mathbf{P} = \mathbf{NP}$  (for a more detailed argumentation cf. Schirmer 1995).

However, this implication provides only partial information with respect to the tractability of the  $\text{SMPSP}_{\text{PWW}}$ . To see this, let us consider the two possible answers to the " $\mathbf{P} = \mathbf{NP} ?$ " question. On one hand, if  $\mathbf{P} \neq \mathbf{NP}$  were true, all problems in  $\mathbf{NP}$  and thus also the feasibility version of the  $\text{SMPSP}_{\text{PWW}}$  could be solved only by exponential algorithms; since the  $\text{SMPSP}_{\text{PWW}}$  is at least as hard as its feasibility version, also the  $\text{SMPSP}_{\text{PWW}}$  itself would then be exponential. On the other hand, if  $\mathbf{P} = \mathbf{NP}$  would hold, then all problems in  $\mathbf{NP}$  and thus also the feasibility version of the  $\text{SMPSP}_{\text{PWW}}$  would be polynomial. In this case, however, the tractability of the  $\text{SMPSP}_{\text{PWW}}$  would still be open: Being possibly harder to solve than a polynomial problem, it then could still be as well polynomial as exponential. The following result closes this information gap by ruling out the possibility that the  $\text{SMPSP}_{\text{PWW}}$  might actually be harder to solve than the strongly  $\mathbf{NP}$ -complete problems. (The proof uses two variants of the  $\text{SMPSP}_{\text{PWW}}$  which are explained in the Appendix.)

**Theorem 4**  $\text{SMPSP}_{\text{PWW}}$  - under the assumption that all  $d_j$  ( $1 \leq j \leq J$ ) are restricted to values which are polynomial in the input length - is strongly  $\mathbf{NP}$ -easy.

*Proof:*

(i) ( $\text{SMPSP}_{\text{PWW}}^{\text{thr}}$  is strongly  $\mathbf{NP}$ -complete)

( $\text{SMPSP}_{\text{PWW}}^{\text{thr}} \in \mathbf{NP}$ ) Magnitude and length of an instance  $I$  as well length of a certificate of the  $\text{SMPSP}_{\text{PWW}}^{\text{thr}}$  are as for the  $\text{SMPSP}_{\text{PWW}}^{\text{feas}}$ . Again, reading a certificate has a time complexity of  $O(J \cdot T)$  which is polynomial in the input length because  $T$  is so.

Evaluating the constraints will accordingly require the same time as for the  $\text{SMPSP}_{\text{PWW}}^{\text{feas}}$ , plus at most  $T-1$  additions and at most  $T$  multiplications to evaluate the objective function (3). Again, the time complexity of the total evaluation is  $O(J^2 \cdot R \cdot T^2)$ . So, it can be verified in polynomial time that the schedule satisfies the constraints (4) - (6) as well as the condition on the objective function value.

( $\text{SMPSP}_{\text{PWW}}^{\text{feas}} \in_{\text{PR}} \text{SMPSP}_{\text{PWW}}^{\text{thr}}$ ) Assuming some hypothetical algorithm  $A'$  solving  $\text{SMPSP}_{\text{PWW}}^{\text{thr}}$ , each instance  $I$  of  $\text{SMPSP}_{\text{PWW}}^{\text{feas}}$  can be solved by an algorithm  $A$  that proceeds in the following way: Construct from  $I$  an instance  $I'$  of  $\text{SMPSP}_{\text{PWW}}^{\text{thr}}$  where the bound  $B$  is taken to be  $T$  and solve it by  $A'$ . If  $A'$  returns a solution for  $I'$ , then that solution is also feasible; accordingly the solution for  $I$  is "yes". If  $A'$  returns "no", then no feasible solution for  $I'$  exists; hence the solution for  $I$  is "no" as well. This construction can be done in linear, thus in polynomial time such that  $A$  provides a polynomial reduction from  $\text{SMPSP}_{\text{PWW}}^{\text{feas}}$  to  $\text{SMPSP}_{\text{PWW}}^{\text{thr}}$ .  $A$  is also pseudo-polynomial since all numbers - except of  $T$  - occurring in any constructed instance  $I'$  of  $\text{SMPSP}_{\text{PWW}}^{\text{thr}}$  also occur in the

original instance  $I$  of  $\text{SMPSP}^{\text{feas}}_{\text{PWW}}$  and  $T$  is assumed to be polynomially bounded; hence, the magnitude of  $I$  can easily be bounded by the identical polynomial in the magnitude of  $I$ .

- (ii)  $(\text{SMPSP}^{\text{ext}}_{\text{PWW}} \infty_{\text{pR}} \text{SMPSP}^{\text{thr}}_{\text{PWW}})$  Holds trivially since  $\text{SMPSP}^{\text{thr}}_{\text{PWW}}$  can be restricted to  $\text{SMPSP}^{\text{ext}}_{\text{PWW}}$ .
- (iii)  $(\text{SMPSP}_{\text{PWW}} \infty_{\text{pR}} \text{SMPSP}^{\text{ext}}_{\text{PWW}})$  Suppose the algorithm  $A$  solves  $\text{SMPSP}^{\text{ext}}_{\text{PWW}}$  when provided with an instance  $[J, d_j, R, k_{jr}, K_r, \angle, B, \Xi]$ . (To specify the particular instance which  $A$  has to solve, we will refer to the application of  $A$  as *calling*  $A[J, d_j, R, k_{jr}, K_r, \angle, B, \Xi]$ .) Let denote  $I$  an instance  $I$  of  $\text{SMPSP}_{\text{PWW}}$ . We now have to show that  $I$  could be solved by calling  $A$  on several different instances of  $\text{SMPSP}^{\text{ext}}$  while doing so no more than a number of times polynomially bounded in  $\text{LNG}_{\text{STD}}(I)$ . We will do this in two steps: First, determine the optimal objective function value of  $I$ , i.e. the minimum project length; second, construct an optimal solution, i.e. a full schedule having minimum length.

It is clear from the activity completion constraints (4) that any optimal schedule will have to comprise activity 1. In order to avoid unnecessary delays, it will be regarded as "completed" at the beginning of period 1, in other words at the end of "period" 0. Consequentially, any optimal schedule will include  $\{(1, 0)\}$  as a partial schedule. Now, from the objective function (3), equation (1), and  $d_j \in \mathbb{N}_0$  it follows that the length  $B^*$  of any optimal schedule has to satisfy  $0 \leq B^* \leq T$ . Accordingly, by performing a binary search within this interval, one can determine  $B^*$  with a sequence of at most  $\lceil \log T \rceil$  calls of  $A[J, d_j, R, k_{jr}, K_r, \angle, B, \{(1, 0)\}]$  with different values of  $B$ . The binary search procedure can be described in pseudo code as follows:

```

BMIN ← 0;
BMAX ← T;
while not (BMAX - BMIN = 1)
{
    B ← ⌈(BMIN + BMAX) / 2⌉;
    call A[J, dj, R, kjr, Kr, ∠, B, {(1, 0)}];
    if (solution = "yes")
        BMIN ← B;
    else
        BMAX ← B;
}
B* ← BMAX;
return (B*);

```

In order to build an optimal length schedule, let a candidate partial solution (cps) be a partial schedule that can be extended to an optimal one, i.e. to a full schedule of mini-

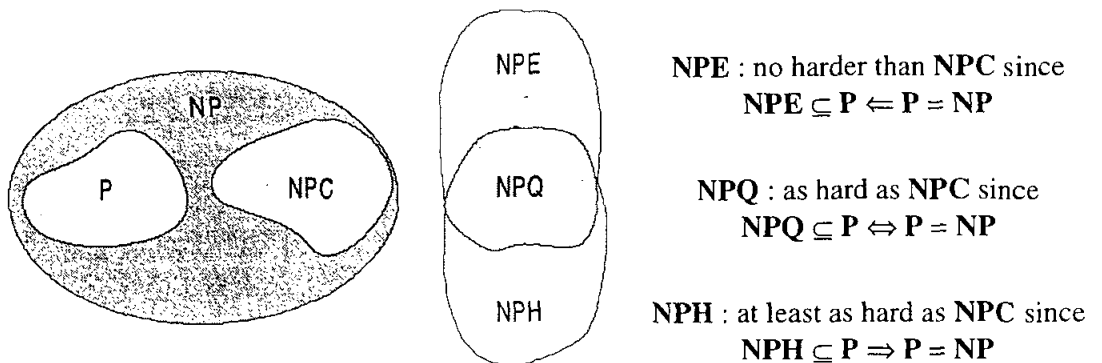


mum length. Clearly,  $\{(1, 0)\}$  is a cps; hence, there is at least one activity  $j \in J \setminus \{1\}$  such that  $\{(1, 0), (j, t)\}$  is a cps.  $j$  is always taken to be the next activity in  $J$ .  $t$  can be identified from checking all periods  $t'$  within the interval  $[EF_j, LF_j]$  by calling  $A[J, d_j, R, k_{jR}, K_R, \angle, B^*, \{(1, 0), (j, t')\}]$ ; this will involve at most  $\lceil \log T \rceil$  calls. In general, for each cps  $\{(1, 0), (2, t_2), \dots, (I, t_I)\}$  with  $I < J$ , another cps  $\{(1, 0), (2, t_2), \dots, (I, t_I), (I+1, t_{I+1})\}$  can be determined by a sequence of at most  $\lceil \log T \rceil$  calls of  $A$  such that - given the optimal schedule length  $B^*$  - an optimal schedule can be identified by at most  $J \cdot \lceil \log T \rceil$  calls of  $A$ .

Having specified the above algorithm, it remains to be verified that  $A$  is indeed a pseudo-polynomial reduction. Summing up the above numbers yields at most  $(J + 1) \cdot \lceil \log T \rceil$  as the total number of calls, which is essentially  $O(J \cdot \log T)$ ; this function can easily be polynomially bounded in the length  $LNG_{STD}(I)$  of a  $SMPSP_{PWW}$ -instance, which is essentially  $O(J \cdot R \cdot \log MAX_{STD}(I))$ , since  $T \leq MAX_{STD}(I)$ . As all the numbers that occur in a  $SMPSP_{PWW}$ -instance  $I$  also appear in the corresponding  $SMPSP^{ext}_{PWW}$ -instance  $I'$  and as  $MAX_{STD}(I) = MAX_{STD}(I')$ , the remaining conditions on a pseudo-polynomial reduction are met, as well. ■

Showing a problem to be **NP**-easy allows to characterize it as no harder to solve than the strongly **NP**-complete problems; therefore, the  $SMPSP_{PWW}$  can be solved by a polynomial or a pseudo-polynomial algorithm if  $P = NP$ . Finally, the combination of Theorems 3 and 4 together implies that the  $SMPSP_{PWW}$  is strongly **NP**-equivalent, indicating that not only the  $SMPSP^{feas}_{PWW}$  but also the  $SMPSP_{PWW}$  is of essentially the same computational complexity as the strongly **NP**-complete problems. This implies that if the fundamental question "Is  $P = NP$ ?" will eventually be answered by "yes" ("no"), this also would immediately establish the complexity of the  $SMPSP_{PWW}$  as polynomial (exponential).

Figure 2 illustrates the mutual relationships between the complexity classes used in the above proofs and summarizes the implications of membership in one of them for a problem. **NPC** (**NPH**, **NPE**, **NPQ**) is taken to denote the class of **NP**-complete (**NP**-hard, **NP**-easy, **NP**-equivalent) problems (Schirmer 1995).

**Figure 2***Overview of Different Complexity Classes*

## 6. Summary and Outlook

Motivated by recent findings of Brüggemann (1995) that certain commonly used combinations of an encoding and a model may be exponential, we have examined the most simple paradigm of project scheduling problems in the presence of resource constraints, namely the SMPSP. In doing so, we focussed on so-called PWW-models because these are well established as the standard practice of representing these problems. It turned out that the PWW-model of the SMPSP is exponential - regardless of the particular encoding used. However, this discouraging result can be improved to a strongly **NP**-equivalence result by adding one moderate restriction on the problem parameters.

Future work in this regard should be directed along the following lines:

- Investigate whether these or similar properties appear for the SMPSP only or for other PSP as well.
- If the latter case holds, derive criteria which allow to identify this property in general.
- Check whether other model formulations for the SMPSP exist which are in **NP** even for arbitrarily large values of  $T$ .

## Acknowledgement

We are very grateful to Andreas Drexl for his as usual constructive comments and encouraging advice. We would also like to thank Wolfgang Brüggemann for several inspiring and helpful discussions.

## Appendix

We here describe two variants of the SMPSP, namely its *threshold* and its *extension variant*, which are used in the proof of Theorem 4 (A detailed coverage of such variants and the proof technique involved is provided in Schirmer 1995). In order to emphasize the correspondence between the original SMPSP and its variants, we start off with recalling an - informal - description of the SMPSP. It is easy to see how to derive a PWW-model (such as given in Section 3) from these problem descriptions (such as given below). We thus feel free to omit, for the sake of brevity, the respective PWW-model formulations of both variants.

### (P1) Single-Mode Project Scheduling Problem (SMPSP)

Given  $J \in \mathbb{N}$  (number of activities),  $J = \{1, \dots, J\}$  (activities),  $d_1 = d_J = 0$ ,  $d_j \in \mathbb{N}$  ( $2 \leq j \leq J-1$ ) (duration of  $j$ ),  $R \in \mathbb{N}$  (number of renewable resources),  $k_{jr} \in \mathbb{N}_0$  ( $1 \leq j \leq J$ ;  $1 \leq r \leq R$ ) (per-period usage of  $r$  by  $j$ ),  $K_r \in \mathbb{N}$  ( $1 \leq r \leq R$ ) (per-period-availability of  $r$ ), and  $\angle$  a partial order (a precedence order) on  $J$ , find an assignment of periods to activities, i.e. a set  $\{(j_n, t_n) \mid 1 \leq n \leq J \wedge 1 \leq j_n \leq J \wedge 1 \leq t_n \leq T\}$ , covering all activities that ensures for each renewable resource  $r$  that in each period the total usage of  $r$  by all activities performed in that period does not exceed the per-period availability of  $r$ , and respects the partial order  $\angle$  (a full schedule) such that the expression (project length)

$$\max \{t_n \mid 1 \leq n \leq J\} \tag{17}$$

is minimal.

### (P2) Single-Mode Project Scheduling Problem, Threshold Variant (SMPSP<sup>thr</sup>)

Given an instance of the SMPSP and  $B \in \mathbb{N}$  (a bound), is there a full schedule such that the project length is equal to or less than  $B$ ?

### (P3) Single-Mode Project Scheduling Problem, Extension Variant (SMPSP<sup>ext</sup>)

Given an instance of the SMPSP,  $B \in \mathbb{N}$  (a bound),  $I \in \mathbb{N}$  with  $I \leq J$ , and an assignment of periods to activities, i.e. a set  $\Xi = \{(j_n, t_n) \mid 1 \leq n \leq I \wedge 1 \leq j_n \leq J \wedge 1 \leq t_n \leq T\}$ , covering part of the activities that ensures for each renewable resource  $r$  that in each period the total usage of  $r$  by all activities performed in that period does not exceed the per-period availability of  $r$ , and respects the partial order  $\angle$  (a partial schedule), can  $\Xi$  be extended to a set  $\{(j_n, t_n) \mid 1 \leq n \leq J \wedge 1 \leq j_n \leq J \wedge 1 \leq t_n \leq T\}$  (a full schedule, as defined above) such that the project length is equal to or less than  $B$ ?

Note that for  $I = J$  the above notion of a partial schedule includes the full schedule as a special case.

## References

- BAKER, K.R. (1974), *Introduction to sequencing and scheduling*, Wiley, New York.
- BRÜGGEMANN, W. (1995), *Ausgewählte Probleme der Produktionsplanung - Modellierung, Komplexität und neuere Lösungsmöglichkeiten*, Physica, Heidelberg.
- DREXL, A. (1990), "Fließbandaustaktung, Maschinenbelegung und Kapazitätsplanung in Netzwerken - Ein integrierender Ansatz", *Zeitschrift für Betriebswirtschaft* 60, pp. 53-70.
- GAREY, M.R. AND D.S. JOHNSON (1979), *Computers and intractability - A guide to the theory of NP-completeness*, W.H. Freeman, San Francisco.
- GAREY, M.R., D.S. JOHNSON, AND R. SETHI (1976), "The complexity of flowshop and job-shop scheduling", *Mathematics of Operations Research* 1, pp. 117-129.
- KOLISCH, R. (1995), *Project scheduling under resource constraints - Efficient heuristics for several problem classes*, Physica, Heidelberg.
- PAPADIMITRIOU, C.H. (1994), *Computational complexity*, Addison-Wesley, Reading, Mass.
- PAPADIMITRIOU, C.H. AND K. STEIGLITZ (1982), *Combinatorial optimization: Algorithms and complexity*, Prentice-Hall, Englewood Cliffs (N.J.).
- PRITSKER, A.A.B., W.D. WATTERS, AND P.M. WOLFE (1969), "Multiproject scheduling with limited resources: A zero-one programming approach", *Management Science* 16, pp. 93-108.
- SCHIRMER, A. (1995), "A guide to complexity theory in operations research", *Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel* 381.
- SLOWINSKI, R. (1981), "Multiobjective network scheduling with efficient use of renewable and nonrenewable resources", *European Journal of Operational Research* 7, pp. 265-273.
- SPRECHER, A. (1994), *Resource-constrained project scheduling - Exact methods for the multi-mode case*, Springer, Berlin.