Jena Research Papers in Business and Economics

Managing and modelling general resource transfers in (multi-)project scheduling

Doreen Krüger, Armin Scholl

28/2007

Revision date:
March 2008

Revised version of
"Modelling general resource transfers in (multi-)project scheduling"
(December 2007)

Jenaer Schriften zur Wirtschaftswissenschaft

Working and Discussion Paper Series
School of Economics and Business Administration
Friedrich-Schiller-University Jena

ISSN 1864-3108

Publisher:

Wirtschaftswissenschaftliche Fakultät
Friedrich-Schiller-Universität Jena
Carl-Zeiß-Str. 3, D-07743 Jena

www.jbe.uni-jena.de

Editor:

Prof. Dr. Hans-Walter Lorenz
h.w.lorenz@wiwi.uni-jena.de

Prof. Dr. Armin Scholl
armin.scholl@wiwi.uni-jena.de

www.jbe.uni-jena.de

Managing and modelling general resource transfers in (multi-)project scheduling

Doreen Krüger, Armin Scholl

Friedrich-Schiller-Universität Jena
Fakultät für Wirtschaftswissenschaften
Lehrstuhl für Betriebswirtschaftliche Entscheidungsanalyse
Carl-Zeiß-Straße 3, D-07743 Jena

e-Mail: {d.krueger, a.scholl}@wiwi.uni-jena.de

Revision date:
March 2008

Revised version of
"Modelling general resource transfers in (multi-)project scheduling"
(December 2007)

Abstract

Most approaches to multi-project scheduling are based on the assumption that resources can be transferred between projects without any expense in time and cost. As this assumption often is not realistic, we generalise the multi-project scheduling problem (RCMPSP) by additionally including transfer times and cost. To integrate this aspect, in a first step, we develop a framework for considering resource transfers in single and multi-project environments. It includes managerial approaches to handle resource transfers, a classification of resource transfer types and new roles that resources can take in these transfers. Afterwards, we define the multi-project scheduling problem with transfer times (RCMPSPTT) and formulate it in a basic and an extended version as integer linear programmes. Eventually, it is supplemented for the first time by cost considerations and introduced as resource constrained multi-project scheduling problem with transfer times and cost (RCMPSPTTC). Computational experiments compare the presented managerial approaches and prove the necessity of explicitly considering transfer times in project scheduling. Moreover, the experiments evaluate the presented MIP models and show that specialised solution procedures are vital.

Keywords:

project scheduling – combinatorial optimization – mathematical model – transfer times – transfer cost – setup – resource flow

1 Introduction

The resource constrained multi-project scheduling problem (RCMPSP) as an extension of the well-known RCPSP is considered as the simultaneous scheduling of two or more projects which demand the same scarce resources. One may distinguish two main research fields in multi-project scheduling - the static and the dynamic project environment (Dumond and Mabert 1988, p. 102). Our research focuses on the static environment, which assumes a closed project portfolio which is not changing over time.

Scheduling in a static environment has been studied amongst others by Fendley (1968), Pritsker et al. (1969), Kurtulus and Davis (1982), Kurtulus and Narula (1985), Lawrence and Morton (1985), Lova et al. (2000) or Lova and Tormos (2001). All papers apply either a single- or a multi-project approach. The single-project approach is equivalent to the RCPSP, since it merges all projects of the multi-project to an artificial super-project with a dummy start and end job and minimises the multi-project duration, i.e., the finishing time of the dummy end job (Kurtulus and Davis 1982, p. 162). The multi-project approach keeps the projects separate for priority calculation and minimises the mean project delay (Kurtulus and Narula 1985, p. 59).

Though resource transfers are highly relevant in practice, especially in a multi-project environment, most research papers neglect them. Obviously, transferring resources from one project to another (or even one job to another one in the same project) takes considerable time

- when a resource is moved physically from one location to another, e.g. heavy machines, specialists that fly around the world,
- when a resource has to be adjusted, e.g. setup times for machines, human resources that have to get acquainted with new projects. Especially for human resources the learning, forgetting and relearning life-cycle plays a vital role in transfer time considerations.

Setup times which are a variant of transfer times as considered in this paper have already been investigated in production scheduling (cf. Aldowaisan et al. 1999) and lot sizing (cf. Jans and Degraeve 2008) extensively. In single-project scheduling, limited research has been done (cf. Mika et al. 2006). Kolisch (1995) develops a zero-one programme for an extension of the RCPSP restricted to one unit resource capacities where a single resource required by a job can demand a (sequence-independent) setup. Debels and Vanhoucke (2006) consider setup times in the pre-emptive RCPSP assuming that a (sequence-independent) setup is necessary whenever a job is pre-empted. Neumann et al. (2003, ch. 2.14) extend the approach of Schwindt and Trautmann (2000) and present the RCPSP with time windows and sequence-dependent changeover times. Unlike Schwindt and Trautmann, who assume that resource requirements are binary, they allow for arbitrary resource capacities and resource requirements of jobs. Neumann et al. split the problem into two interdependent subproblems. At first, they determine a time and precedence feasible schedule. Afterwards, they check whether this schedule is feasible with respect to resource transfers. Schwindt and Trautmann (2003) consider a real-world production scheduling problem to which they apply an approach based on project scheduling concepts. They consider sequence-dependent changeover times for specified changeover resources.

In the context of multi-project scheduling, setup times or transfer times are rarely encountered in literature up to now. Neumann (2003) points out that the problem formulation presented in Neumann et al. (2003) can be applied to multi-project scheduling with distributed locations. Yang and Sum (1993, 1997) were the first, who considered resource transfer times in multi-projects. They assume a dynamic multi-project environment and a dual-level management structure. A central resource pool manager assigns resources to projects, whereas a project manager schedules jobs within his project using the allocated resources. Resource changeovers can only be handled via the central pool. They consider sequence- and resource-independent constant transfer times from this central resource pool to projects while no transfer times from the project to the pool occur. Dodin and Elimam (1997) present an audit scheduling problem with sequence- and resource dependent travel (setup) cost when auditors change their assignments. As travelling is assumed to take place during non-working hours, transfer times are not explicitly regarded.

Obviously, there is a lack in considering general resource transfers with sequence- and resource-dependent transfer times which are highly relevant in practice. This paper intends to contribute to closing this gap. It is organised as follows: Section 2 provides a framework for handling and classifying transfers as well as new resource roles in presence of transfers. In Section 3, a basic problem restricted to renewable resources and stand-alone transfers is developed and modelled. This problem and the corresponding model is extended to non-renewable resources and resource-supported transfers in Section 4. Cost considerations are integrated in Section 5. Computational experiments showing the importance of the new approaches and the necessity for new solution procedures in Section 6 and conclusions in Section 7 complete the paper.

2 Framework for resource transfers in project environments

In the following, we develop a framework for considering resource transfers in single and multi-projects which is based on (1) the way resource transfers are handled, (2) a classification of resource transfers and (3) roles the resources can take in these transfers.

2.1 Managerial approaches

We distinguish three general managerial approaches to deal with resource transfers:

- **Transfer-neglecting approach:** This approach (often only implicitly) assumes that transfers are possible in an arbitrary manner causing marginal transfer time and cost. Consequently, resource transfers are not considered at all. Most state-of-the-art approaches for multi-project scheduling as cited in the first part of Section 1 apply this approach as they ignore transfer times or try to integrate them into job durations. Yet, this type of transfer is found very seldom in enterprises (e.g., transferring money). As it might be somewhat justified to disregrad transfer aspects within single projects, a neglect is usually inappropriate in multi-projects.
- **Transfer-reducing approach:** This approach assumes that all transfer times and cost are prohibitive such that resource transfers are impossible or should be kept to a minimum. Thus, resources are allocated to (sub-)projects a priori without any or restricted possibility of transferring them to another (sub-)project. Only a few examples exist for transfers being completely impossible (within the static horizon of a multi-project), e.g., demounting, transferring and reinstalling of very large machinery like a bucket-wheel excavator for coal mining. A more realistic assumption is that resource transfers are restricted to take place only after the resources have finally been released from a project (after their last usage). That is, resources are allocated to a project during its execution even if they are idle for long time (cf. Mellen-

tien et al. 2003; Schwindt and Trautmann 2003). This approach is especially relevant for multi-projects since transfers between jobs of the same project are required anyway.

- **Transfer-using approach:** For many resource types it is possible to transfer them between jobs of different projects with certain time requirement and at certain cost. A transfer will always take place if its expenses (in terms of time and cost) pay off by an adequate increase in efficiency of project execution. It is the most general approach because it includes the previous ones. It may include inexpensive or costless transfers just as prohibitive ones. Moreover, it refers to single- as well as multi-projects. Usually transfer expenses are much higher between projects than within the same project.

2.2 Types of resource transfers

While setup operations have been classified into categories like, e. g., schedule-dependent, precedence-dependent and independent as well as divided and undivided or synchronous, asynchronous or semi-synchronous setups by Mika et al. (2006) from a rather technical point of view, we present a complementary classification from a managerial point of view in which the various setup categories described by Mika et al. may be embedded.

The transfer-using managerial approach is to be developed, modelled and evaluated in this paper. Accepting resource transfers being a realistic and important aspect to be considered explicitly, they can be categorised along three dimensions:

- **Time:** A transfer can originate in the start or the end of a job. Moreover, it can target on the beginning or the end of a job. Combining these possibilities, four types of resource transfers are identified (finish-to-start, start-to-finish, start-to-start, finish-to-finish). For further details and examples see below and Section 4.
- **Abstraction:**
    - *Physical transfers* are characterised by moving a resource from one place to another, e. g. a crane that has to be transported from construction site A to site B.
    - *Non-physical transfers* occur when resources do not change locations. For illustration, one can imagine human resources changing projects without leaving their desk but having to get acquainted with the new project or even job. Setup times for machines are another example for non-physical resource transfers which take time and may cause cost although the resource is not moved from one place to another.
- **Support**:
    - A *stand-alone transfer* takes place when a resource changes projects without any support by other resources. A project member who travels from one location (project A) to another (project B) within a city by public transport during a day may be an example for a physical stand-alone transfer. Whereas a changeover of the same employee to another project by only switching to another job, without briefing by other team members is a non-physical stand-alone transfer. At the time dimension only finish-to-start transfer-relationships are relevant if solely stand-alone transfers occur in a (multi-)project because only executing resources are considered. They leave a job after they have finished it and are required at the beginning of another job to start the execution.

- *Resource using transfers* require renewable resources that support the transfer of a job executing resource. A physical resource using transfer is identified, e. g., when a crane has to be moved from site A to site B. Construction workers, lorries, machines and equipment are necessary to dismantle, transport and put the crane up again. Afterwards, the same resource units can be used otherwise for job execution or transfers again. A non-physical resource using transfer may take place when a team member in an IT project changes to another project with the aid of another member, who briefs him on the status quo of the project.
- *Transfers* are *resource consuming* if non-renewable resources are required for support. When, e. g., due to cleaning processes which necessitate detergents, setup times for machines occur, a non-physical resource consuming transfer takes place. The detergent is consumed and not available for further jobs. A physical resource consuming transfer can be illustrated by an IT specialist which has to change between projects which are located far away from each other and can only be reached by plane. The cost for the flight is not negligible and decreases the budget of the (multi-)project, which is a non-renewable resource. Along the time dimension all four types of transfer are imaginable for resource using and consuming transfers.

It should be obvious that resource transfers often cannot be put into only one category. Frequently, a combination of several transfer forms is necessary to classify an actual transfer. Moreover, it is to mention that this classification is not only applicable to the multi-project case. All these transfer categories are also relevant for single-project environments. However, it is assumed that the impact of resource transfers is higher in a multi-project context because the distances in place and content are larger than in single projects.

2.3 Resource roles in resource transfers

There exist widely accepted classifications of general resource types concerning activity execution (renewable, non-renewable, partially renewable, dedicated etc.; cf. Brucker et al. 1999; Demeulemeester and Herroelen (2002, ch. 2.2). However, there are only very limited reflections on the roles resources might take during resource transfers. In order to close this gap, we categorise these roles.

- When resource transfers are (really) negligible, the resources to be transferred are classified as **free company resources**. This resource type may move in the company for free in sense of time and cost. As already explained, the monetary budget may be such a resource.
- If resource transfers are (really) impossible, company resources must be divided and **dedicated** to projects before (multi-)project execution starts. This requires hierarchical scheduling systems, where resources are allocated to projects in a fixed manner by an upper level decision maker. In this scenario, decision rules for optimally allocating resources to projects become vital. If resource transfers are possible on principle but only allowed after having finished a project, **allocatable** resources are present. These resources are assigned to a (sub-) project during its complete run time. For example, setup states of a machinery might be preserved until a complete subprocess is finished (cf. Schwindt and Trautmann 2003). Mellentien et al. (2003) refer to the "one face to the customer" concept. It requires that an adviser

must accompany a group of customers, who are visiting a firm, during the entire visit even if he was not required for some programme items and could do other jobs.

- If a resource can be transferred at certain time or cost without the above-mentioned limitations, it is called **transferable resource**. A lot of examples exist, e.g., personnel, tools, material and even machinery will be transferred if it is economic to spend time or cost required.

Transferable resources can fulfil two functions:

- The resources that are being transferred are **1st tier resources** which are required for executing jobs in receiving projects. Renewable as well as non-renewable resources can take this role. An example is a team member, who acquaints oneself with the new project (non-physical stand-alone transfer).
- Resources which support a transfer are called **2nd tier resources**. *Resource using transfers* are conducted when renewable 2nd tier resources support the transfer of a 1st tier resource. A *resource consuming transfer* requires non-renewable 2nd tier resources. In the example of a crane transfer, the crane is a 1st tier resource since it will be used in the new project for job execution. Lorries and workers are 2nd tier resources as they support the transfer but are not (necessarily) needed in the new project. Yet, it may happen that these lorries and workers are scheduled for job execution in the new project as well. In this case, these two resource types consecutively take two roles. Hence, our concept differs from the "auxiliary resources" presented by Mika et al. (2006), who assume that auxiliary resources are used exclusively for setups but not for activity execution.

Apart from 1st and 2nd tier resources, 3rd to nth tier ones are conceivable. In the example, it is obvious that the lorries must be driven by workers. Thus, drivers support the transportation of the heavy crane indirectly as 3rd tier resources. However, a special treatment of higher level resource roles is not necessary because they can be transformed into 2nd tier resources by directly assigning them to the supported 1st tier resource.

3 Basic problem version RCMPSPTT-1: Stand-alone transfers

In the following, a basic version of the *resource constrained multi-project scheduling problem with transfer times,* called **RCMPSPTT-1**, which only considers stand-alone transfers of 1st tier resources is described and formulated as a mathematical model.

3.1 Problem description

A multi-project containing a set of single projects $P = \{1, \ldots, m\}$ is to be realised. Each project $p \in P$ is composed of a set J_p of real jobs as well as a dummy start job s_p and a dummy end job e_p. The multi-project contains all project jobs as well as a global super source s_0 and sink e_0, which are all merged in set $J' = \{1, \ldots, n\}$.

A finish-to-start activity-on-node network represents the jobs and precedence constraints of the multi-project. A **set of direct predecessors A_j** is given for each job $j \in J'$. Each local project source is successor of the global source s_0, which has no predecessors. The global sink e_0 succeeds each local project sink. Except for this global linkage of all projects, we assume that precedence constraints exist only between jobs of the same project but not between different projects. To ease presentation, we (re-)number all jobs $j \in J'$ topologically, i.e. $j > i$ for each pair $j \in J'$ and $i \in A_j$. Thus, job 1 is the global source node s_0 and job n the global sink node e_0.

It is assumed that each job j has a fixed **duration d_j** and must not be pre-empted after it has been started.[1] Several types of renewable resources $r \in R$ with constant **capacity a_r** are available for project execution in each period. Job $j \in J'$ **requires** a constant integer number of **resource units u_{jr}** of resource type $r \in R$ per period of execution. The required amount must be transferred from other jobs to job j. The **transfer times Δ_{ijr}** depend on the sending job $i \in J'$ and the receiving job $j \in J' - \{i \cup A_i\}$ as well as the resource type $r \in R$ irrespective of the number of resource units transferred. All transfer times are assumed to fulfil the triangular inequality, i. e. $\Delta_{ijr} \le \Delta_{ikr} + \Delta_{kjr}$ for all triplets i, j, k of real jobs. Global source and sink can be considered as a global resource pool where all resources are stored before the multi-project starts and to which all resources must return after having terminated the multi-project. Resources which are kept in the pool are transferred from s_0 to e_0 only virtually with time 0.

In the basic problem, only stand-alone resource transfers are integrated. Thus, only finish-to-start transfers are relevant. In practice, projects restricted to stand-alone transfers frequently occur if only human resources are involved, e. g., in IT or consulting projects. It is not necessary to consider *non-renewable* resources in this basic problem. Either there is sufficient resource capacity for project execution or there is not, as also assumed for the RCPSP.

Dummy jobs have a duration and resource usage of zero for each renewable resource type with exception of the global source s_0 and sink e_0. Although these two jobs are assumed to be dummy jobs taking no time, their resource usage is defined by $u_{s_0 r} = u_{e_0 r} = a_r$ for $r \in R$ as the global source needs to provide all resources to the multi-project while the global sink collects them.

The basic problem RCMPSPTT-1 consists of determining finishing times F_j for all jobs $j \in J'$ (numbered light grey rectangles) and corresponding resource transfer volumes x_{ijr} (black arrows) such that a multitude of constraints is met as visualised in Figure 1. All precedence and resource constraints must be observed while sequence- and resource-type-dependent transfer times (shaded areas) for resources changing to other jobs are taken into account.

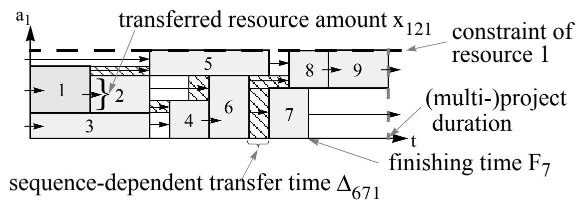

Figure 1. Example schedule for the RCMPSPTT

In the single-project approach, the **multi-project duration MPD** or its relative increase MPDI is minimised. MPD is given by the finishing time F_{e_0} of the global sink e_0. MPDI is measured as the relative deviation of MPD from the time of the multi-project's critical path from s_0 to e_0. In the multi-project approach, the objective is to minimise **mean project delay MD**. It is defined as average relative deviation of the realised finishing time F_{e_p} from the critical path time CP_p over all projects $p \in P$. These objective functions are commonly used in existing literature on the basic multi-project scheduling problem (Pritsker et al. 1969, Kurtulus and Davis 1982, Kurtulus and Narula 1985, Lova and Tormos 2001). However, these objectives for scheduling multiple projects should not be taken for granted. This paper uses them for a first model formulation of the new problem but will replace them by a cost oriented objective in Section 5.

1 For the sake of simplicity, durations are assumed to be integer. Non-integer job durations and/or transfer times can be easily transformed into integer values by changing the scale, e.g., from hours to minutes.

3.2 Mathematical model for the RCMPSPTT-1

The mixed-integer linear programme for RCMPSPTT-1 (Krüger and Scholl 2007), is based on combining the traditional model for RCPSP (Pritsker et al. 1969) with a network flow based formulation of the single-project scheduling problem with sequence-dependent setup times (Neumann et al. 2003, ch. 2.14) as well as the flow formulation of Artigues et al. (2003).

Parameters:

P set of projects; index: p

J_p set of real jobs of project $p \in P$; index: j

J set of real jobs within all projects; $J = \bigcup_{p \in P} J_p$

J'_p jobs of project $p \in P$ including dummy start job end job; $J'_p = J_p \cup \{s_p, e_p\}$

J' set of all real and dummy jobs, i.e., $J' = \bigcup_{p \in P} J_p' \cup \{s_0, e_0\}$

n number of jobs; $n = |J'|$

d_j duration of job $j \in J'$ (with $d_{s_p} = d_{e_p} = 0$ for $p \in P \cup \{0\}$)

T, t upper bound on the project duration; index for periods: t = 0,...,T

A_j set of direct predecessors of job $j \in J'$; $A_{s_0} = \{\ \}$, $A_{e_0} = \bigcup_{p \in P} \{e_p\}$, $A_{s_p} = \{s_0\}$ for $p \in P$;
for $j \in J_p$ with $p \in P$: $A_j \subseteq J_p - \{j\}$ (precedence constraints only within a project)

A_j^* set of direct and indirect predecessors of job $j \in J'$ (predecessors in the transitive closure of the graph)

S_j set of direct successors of job $j \in J'$; $S_{e_0} = \{\ \}$, $S_{s_0} = \bigcup_{p \in P} \{s_p\}$, $S_{e_p} = \{e_0\}$ for $p \in P$
for $j \in J_p$ with $p \in P$: $S_j = \{i \mid i \in J_p \cup \{e_p\} \wedge j \in A_i\}$

S_j^* set of direct and indirect successors of job $j \in J'$

EF_j earliest finishing time of job $j \in J'$; $EF_{s_p} = 0$ for $p \in P \cup \{0\}$ (forwards path)

LF_j latest finishing time of job $j \in J'$ (backwards path)

TI_j time window for finishing job $j \in J'$; $TI_j = [EF_j, LF_j]$

CP critical path time of multi-project

CP_p individual critical path time of project $p \in P$

R set of renewable resources; index: r

a_r number of units of resource $r \in R$ available per period

u_{jr} number of units of resource $r \in R$ required for performing job $j \in J'$ per period, $u_{s_0 r} = u_{e_0 r} = a_r \ \forall r \in R$

Jr_j set of real jobs (including global sink) to which resources might be transferred after having performed $j \in J$;
$Jr_j := J \cup \{e_0\} - \{j\} - A_j^*$

Js_j set of real jobs (including global source) from which resources might be transferred to job $j \in J$;
$Js_j := J \cup \{s_0\} - \{j\} - S_j^*$

Δ_{ijr} time for transferring units of resource $r \in R$ from job $i \in J \cup \{s_0\}$ to job $j \in Jr_i$, $\Delta_{s_0 e_0 r} = 0 \ \forall r \in R$

Variables:

$$f_{jt} = \begin{cases} 1 & \text{if job j is terminated at the end of period t} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } j \in J' \text{ and } t \in TI_j$$

F_j realised finishing time of job $j \in J'$

$$z_{ijr} = \begin{cases} 1 & \text{if (1st tier) resource r is transferred from job i to j} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } r \in R,\ i \in J \cup \{s_0\},\ j \in Jr_i$$

x_{ijr} number of units of resource $r \in R$ transferred from job $i \in J \cup \{s_0\}$ to job $j \in Jr_i$

The *model* is given by the objective function (1) and the set of constraints (2) to (11).

Minimise $\Phi(\mathbf{F}, \mathbf{f}, \mathbf{x}, \mathbf{z})$ such that (1)

$$\sum_{t \in TI_j} f_{jt} = 1 \qquad \text{for all } j \in J' \tag{2}$$

$$F_j = \sum_{t \in TI_j} t \cdot f_{jt} \qquad \text{for all } j \in J' \tag{3}$$

$$F_j - F_i \geq d_j \qquad \text{for } j \in J' \text{ and } i \in A_j \tag{4}$$

$$F_i + \Delta_{ijr} + d_j \leq F_j + T \cdot (1 - z_{ijr}) \qquad \text{for } i \in J \cup \{s_0\}, j \in Jr_i \text{ and } r \in R \tag{5}$$

$$x_{ijr} \leq z_{ijr} \cdot \min\{u_{ir}; u_{jr}\} \qquad \text{for } i \in J \cup \{s_0\}, j \in Jr_i \text{ and } r \in R \tag{6}$$

$$z_{ijr} \leq x_{ijr} \qquad \text{for } i \in J \cup \{s_0\}, j \in Jr_i \text{ and } r \in R \tag{7}$$

$$\sum_{h \in Js_i} x_{hir} = u_{ir} \qquad \text{for } i \in J \cup \{e_0\} \text{ and } r \in R \tag{8}$$

$$\sum_{j \in Jr_i} x_{ijr} = u_{ir} \qquad \text{for } i \in J \cup \{s_0\} \text{ and } r \in R \tag{9}$$

$$f_{jt} \in \{0, 1\}, F_j \geq 0 \qquad \text{for } j \in J' \text{ and } t \in TI_j \tag{10}$$

$$z_{ijr} \in \{0, 1\}, x_{ijr} \geq 0 \qquad \text{for } r \in R, i \in J \cup \{s_0\}, j \in Jr_i \tag{11}$$

The objective function (1) represents either the single- or the multi-project approach. In the single-project perspective, the multi-project duration increase MPDI is minimised. It is equivalent to minimising the multi-project duration MPD:

$$\text{Minimise } MPDI(\mathbf{F}, \mathbf{f}, \mathbf{x}, \mathbf{z}) = \frac{F_{e_0} - CP}{CP} \cdot 100\% \text{, or equivalently, Minimise } MPD(\mathbf{F}, \mathbf{f}, \mathbf{x}, \mathbf{z}) = F_{e_0} \tag{12}$$

In the multi-project approach, the mean project delay MD is used as performance measure:

$$\text{Minimise } MD(\mathbf{F}, \mathbf{f}, \mathbf{x}, \mathbf{z}) = \frac{1}{|P|} \cdot \sum_{p \in P} (F_{e_p} - CP_p) \tag{13}$$

Constraints (2) - (4) represent the well-known time scheduling constraints from the RCPSP formulation. Equations (8) - (9) model resource flows in the multi-project. Inequalities (5) to (7) provide interdependence of time scheduling and resource flow parts of the model by controlling resource transfers. The variables are defined in (10) - (11). Notice that the variables F_j are only contained to ease presentation but can be eliminated from the model.

From a time feasibility perspective, a (finish-to-start) transfer of resource r from job i to j can only take place if i ends before j is started and the time span between both jobs is larger than or equal to the required transfer time for resource type r (constraints (5)). From a resource feasibility point of view, a transfer of resource r from i to j may only occur if both jobs have a positive resource demand. The actually transferred amount of resource r, thus, depends on whether a time-feasible and resource-feasible transfer is possible as well as the resource requirements of both jobs (constraints (6)). Constraints (7) make sure that the binary variable z_{ijr} becomes 1 only if a transfer really takes place. Forcing z_{ijr} to zero if no transfer takes place is not necessary for this model, however, is included to represent the definition of z_{ijr} correctly. Equations (8) guarantee that the resource demand u_{ir} of job i is satisfied by all incoming flows from directly or indirectly preceding jobs to job i. Additionally, (9) ensure that the received quantity flows to succeeding jobs after job i has been finished. Constraints (9) also guarantee for $i = s_0$ that the global source provides resource capacity ar of each resource type r to the multi-project, whereas constraint (8) for $i = e_0$ ensures that all resources are collected at the end of the project.

4 Extended problem RCMPSPTT-2: All resource and transfer types

The basic problem of Section 3.1 is extended by 2nd tier resources which support the transfer of 1st tier resources as described in Section 2.3 using all transfer types presented in Section 2.1. The extended problem is called **RCMPSPTT-2**.

4.1 Problem description

While consumption of non-renewable resources was constant in the basic problem, this is no longer the case if such resources are used to support resource transfers. Thus, we have to add non-renewable resources $r \in NR$ with a total capacity of ar. Each job $j \in J'$ consumes u_{jr} resource units of resource type $r \in NR$ during its execution. Resource- and sequence-dependent transfer times Δ_{ijr} fulfilling triangular inequalities also exist for non-renewable resources.

As a general and reasonable assumption, the time of a resource transfer is determined by the time given for the supported 1st tier resource, i. e. $\Delta_{ijr} \geq \Delta_{ijq}$ for each pair of 1st tier resource $r \in R^*$ and 2nd tier resource $q \in R^*$ supporting r (with $R^* = R \cup NR$). Obviously, Δ_{ijr} cannot be lower than the transfer time of any supporting resource but might be even longer, e.g., transferring a crane (Δ_{ijr}) by a lorry (Δ_{ijq}) may take longer than driving only the lorry from i to j.

Furthermore, we assume that for each resource type only one mode, i.e., combination of supporting resources, exists. Thus, a common support matrix μ is sufficient with μ_{qr} denoting the amount of resource q used ($q \in R$) or consumed ($q \in NR$) for transferring one unit of resource $r \in R^*$. To avoid loops, it is not allowed that a resource type supports itself.

Jobs as well as resource transfers can only be executed when all resources – original and supporting – are available. The flow of resources can use the four time-based transfer types (cf. Section 2.1). The amount of resource $r \in R^*$ that flows from job i to j using transfer type k is expressed by integer variables x^k_{ijr} (with $k \in \{ES, EE, SS, SE\}$ representing finish-to-start, finish-to-finish, start-to-start, and start-to-finish transfer).

During such transfers a resource can take several roles. If it is purely a 2nd tier resource, it becomes available again immediately after the transfer and can be used for other transfers or job execution. A 1st tier resource must execute its assigned job before it becomes available again. If a resource takes both roles simultaneously, e. g., a driver who supports the transfer of a concrete mixer from site A to B and works at site B as construction worker after the transport, the resource is released only after transfer and job execution.

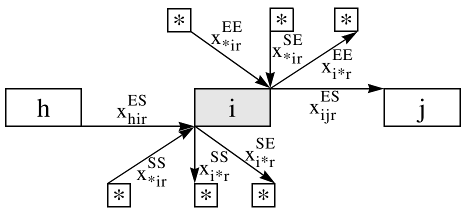

Figure 2. Resource flow variables of job i

The same situation is imaginable for non-renewable resources. However, non-renewable resources in a 1st tier role are not consumed during the transfer whereas the same resource as a 2nd tier one is consumed. Consider a sequence-dependent setup of a machine which requires a detergent. It can be classified as a resource consuming non-physical transfer with a machine as 1st tier and the detergent as consumed 2nd tier resource. However, if the detergent is transported as 1st tier resource to another place, it is not consumed during transfer but during job execution.

Apart from the global source, dummy jobs do not require non-renewable resources. The global source s_0 provides all resource capacity $u_{s_0 r} = a_r$ for $r \in NR$ to the multi-project. The global sink e_0 collects resource amounts not consumed by either jobs or resource transfers.

4.2 Mathematical model for RCMPSPTT-2

We modify and extend the model in Section 3.2 by redefined and additional variables and constraints. We incorporate resource using and consuming transfers, which have not been considered in literature up to now but are an important aspect in (multi-)project scheduling and management.

Adapted parameters:

ε, M	sufficiently small / large positive constant
NR, R*	set of non-renewable / all resources; index: r; $R^* = R \cup NR$
a_r	number of units of resource r available: per period for $r \in R$, in total for $r \in NR$
u_{jr}	number of units of resource r required for job $j \in J$: per period for $r \in R$, in total for $r \in NR$
Δ_{rij}	time for transferring units of resource $r \in R^*$ from job $i \in J \cup \{s_0\}$ to another job $j \in Jr_i$, $\Delta_{s_0 e_0 r} = 0 \; \forall r \in R^*$
μ_{qr}	amount of resource $q \in R^*$ required for transferring one unit of resource $r \in R^*$ per transfer period
sup(q)	set of 1st tier resources $r \in R^*$ which require 2nd tier resource $q \in R^* - \{r\}$
S	set of possible time-based transfer types; $S = \{ES, EE, SS, SE\}$

Variables:

$$z_{ijr}^k = \begin{cases} 1 & \text{if resource r is transferred from job i to j by transfer type k} \\ 0 & \text{otherwise} \end{cases} \quad \text{for } r \in R^*,\ i \in J \cup \{s_0\},\ j \in Jr_i,\ k \in S$$

x_{ijr}^k	total amount of resource $r \in R^*$ transferred from job $i \in J \cup \{s_0\}$ to $j \in Jr_i$ by transfer type $k \in S$
$\bar{x}_{ijr}^k$	amount of 2nd tier resource $r \in NR$ consumed during transfer of 1st tier resources from job $i \in J \cup \{s_0\}$ to job $j \in Jr_i$ with transfer type $k \in S$ (only introduced for model reduction reasons)
α_{ir}, β_{ir}	inflow surplus of resource $r \in R^*$ into job i at its beginning which flows out at start/end of job i

Objective functions (12) and (13) can still be applied in this model formulation. Time scheduling constraints (2) - (4) of the basic model as well as definitions of time related variables (10) are required for the extended model, too. Resource transfer related constraints (5) - (9) as well as variable definitions (11) are replaced by the following *constraints*:

- Implicit precedence relationships for resource transfers exist, which add up to four types of time-based transfer possibilities. A resource $r \in R^*$ can be transferred as 1st or 2nd tier resource from the end of job i to the start of job j considering transfer time Δ_{ijr}. This implicit finish-start relation is modelled in (14), where z_{ijr}^{ES} can become 1 only if this transfer is time feasible. The inequalities (15) – (17) consider the other time-based transfer types.

$$F_i + \Delta_{ijr} \leq F_j - d_j + T \cdot (1 - z_{ijr}^{ES}) \qquad \text{for } i \in J \cup \{s_0\},\ j \in Jr_i \text{ and } r \in R^* \tag{14}$$

$$F_i + \Delta_{ijr} \leq F_j + T \cdot (1 - z_{ijr}^{EE}) \qquad \text{for } i \in J \cup \{s_0\},\ j \in Jr_i - \{e_0\} \text{ and } r \in R^* \tag{15}$$

$$F_i - d_i + \Delta_{ijr} \leq F_j - d_j + T \cdot (1 - z_{ijr}^{SS}) \qquad \text{for } i \in J,\ j \in Jr_i \text{ and } r \in R^* \tag{16}$$

$$F_i - d_i + \Delta_{ijr} \leq F_j + T \cdot (1 - z_{ijr}^{SE}) \qquad \text{for } i \in J,\ j \in Jr_i - \{e_0\} \text{ and } r \in R^* \tag{17}$$

- A real transfer ($x_{ijr}^k > 0$) can only take place if the transfer is time feasible ($z_{ijr}^k = 1$). The maximal amount to be transferred is limited by the available capacity of resource $r \in R^*$.

$$x_{ijr}^k \leq z_{ijr}^k \cdot a_r, \quad z_{ijr}^k \leq M \cdot x_{ijr}^k \qquad \text{for } i \in J \cup \{s_0\},\ j \in Jr_i,\ r \in R^*,\ k \in S \tag{18}$$

- The global source s_0 provides capacity ar of each resource $r \in R^*$ to the multi-project. It is assumed that only the outflow at the ending of sink s_0 is considered. The outflow at the beginning of s_0 is set to zero (see constraints (30)). Redundant resources can be sent directly from global source s_0 to global sink e_0 ($x^{ES}_{s_0,e_0,r}$).

$$\sum_{j\in J\cup\{e_0\}} (x^{ES}_{s_0,j,r} + x^{EE}_{s_0,j,r}) = a_r \qquad \text{for } r\in R^* \qquad (19)$$

- The global sink e_0 collects resource capacity ar for each renewable resource $r\in R$. It is assumed that only the inflow at the beginning of dummy job e_0 is considered. The inflow at the ending of e_0 is set to zero (see constraints (30)).

$$\sum_{i\in J\cup\{s_0\}} (x^{ES}_{i,e_0,r} + x^{SS}_{i,e_0,r}) = a_r \qquad \text{for } r\in R \qquad (20)$$

- For non-renewable resources $r\in NR$ it must be ensured that the capacity ar is not exceeded by resource consuming transfers and the (constant) consumption of all jobs as defined in (21). Consequently, it may happen that no feasible solution can be determined for a given capacity ar and given resource demands uir even if the instance is feasible when no resource consuming transfers are considered.

$$\sum_{i\in J\cup s_0}\sum_{j\in Jr_i}\sum_{k\in S}\bar{x}^k_{ijr} + \sum_{i\in J\cup\{s_0\}}(x^{ES}_{i,e_0,r} + x^{SS}_{i,e_0,r}) \le a_r - \sum_{i\in J} u_{ir} \quad \text{for } r\in NR \qquad (21)$$

- The equations (22) determine the consumed amount $\bar{x}^k_{ijq}$ of a non-renewable 2nd tier resource q during a transfer of a supported resource r from i to j. Per transferred unit of r and per period of transfer μ_{qr} units of q are consumed.

$$\sum_{r\in sup(q)} \mu_{qr}\cdot\Delta_{ijr}\cdot x^k_{ijr} = \bar{x}^k_{ijq} \qquad \text{for } i\in J\cup\{s_0\},\ j\in Jr_i,\ q\in NR,\ k\in S \qquad (22)$$

- For resource $q\in NR$, at least the amount required for a transfer that consumes resource q must be provided. Additionally, resource $q\in NR$ may be transferred as 1st tier resource.

$$\bar{x}^k_{ijq} \le x^k_{ijq} \qquad \text{for } i\in J\cup\{s_0\},\ j\in Jr_i,\ q\in NR,\ k\in S \qquad (23)$$

- Sufficient units of renewable 2nd tier resources $q\in R$ must support a transfer of 1st tier resource $r\in sup(q)$ from job i to j of type k, i.e., at least $\sum_{r\in sup(q)}\mu_{qr}\cdot x^k_{ijr}$ units. However, this is only necessary if the transfer really takes time ($\Delta_{ijr} > 0$) as indicated by the binary factor $\lceil\Delta_{ijr}/(\Delta_{ijr}+\varepsilon)\rceil$ with sufficiently small positive number ε.

$$\sum_{r\in sup(q)} \mu_{qr}\cdot\left\lceil\frac{\Delta_{ijr}}{\Delta_{ijr}+\varepsilon}\right\rceil\cdot x^k_{ijr} \le x^k_{ijq} \qquad \text{for } i\in J\cup\{s_0\},\ j\in Jr_i,\ q\in R,\ k\in S \qquad (24)$$

- The inflow of renewable resources of type $r\in R$ at the beginning of job i must satisfy at least the demand u_{ir} of job i. Redundant units, if any, can already flow out at the beginning of i (denoted by α_{ir}) or also flow out at the ending of job i (denoted by β_{ir}). This splitting up is formalised in (25) and illustrated in Figure 3.

$$\sum_{h\in Js_i-\{s_0\}} (x^{ES}_{hir} + x^{SS}_{hir}) = u_{ir} + \alpha_{ir} + \beta_{ir} \qquad \text{for } i\in J,\ r\in R \qquad (25)$$

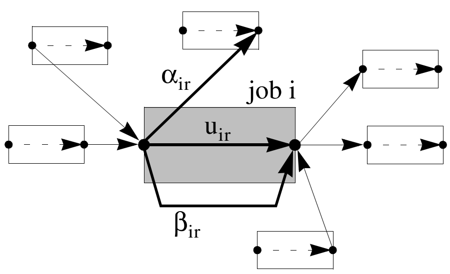

Figure 3. Splitting up inflow

- The inflow of non-renewable resources $r\in NR$ at the start of job i must satisfy at least its demand uir. Additional units flow out at the beginning (α_{ir}) or the ending (β_{ir}) of i. In contrast to renewable resources, the consumption during the transfer must be considered:

$$\sum_{h\in Js_i-\{s_0\}} (x_{hir}^{ES}-\bar{x}_{hir}^{ES}+x_{hir}^{SS}-\bar{x}_{hir}^{SS}) = u_{ir}+\alpha_{ir}+\beta_{ir} \qquad \text{for } i\in J,\ r\in NR \tag{26}$$

- The immediate outflow of resources $r\in R^*$ at the start of real job i is given by α_{ir}.

$$\sum_{j\in Jr_i} (x_{ijr}^{SS}+x_{ijr}^{SE}) = \alpha_{ir} \qquad \text{for } i\in J,\ r\in R^* \tag{27}$$

- The outflow of renewable resources $r\in R$ at the ending of real job i is given by the demand uir and the surplus β_{ir} which waited at job i for further usage. Furthermore, all resources that reach job i at its end can be transferred to any job j immediately.

$$\sum_{j\in Jr_i} (x_{ijr}^{ES}+x_{ijr}^{EE}) = u_{ir}+\beta_{ir}+\sum_{h\in Js_i} (x_{hir}^{EE}+x_{hir}^{SE}) \quad \text{for } i\in J \quad ,\ r\in R \tag{28}$$

- For $r\in NR$, the amount uir is consumed during the execution of job i. The outflow at the ending of job i to any job j is, thus, given by the idle surplus β_{ir} and just arriving (remaining) units (cf. (26)).

$$\sum_{j\in Jr_i} (x_{ijr}^{ES}+x_{ijr}^{EE}) = \beta_{ir}+\sum_{h\in Js_i} (x_{hir}^{EE}-\bar{x}_{hir}^{EE}+x_{hir}^{SE}-\bar{x}_{hir}^{SE}) \quad \text{for } i\in J,\ r\in NR \tag{29}$$

- The variables are constrained as follows:

$$z_{ijr}^{k}\in\{0,1\},\ x_{ijr}^{ES}\geq 0 \qquad \text{for } i\in J\cup\{s_0\},\ j\in Jr_i,\ r\in R^*,\ k\in S$$
$$\bar{x}_{ijr}^{ES}\geq 0 \qquad \text{for } i\in J\cup\{s_0\},\ j\in Jr_i,\ r\in NR$$
$$x_{ijr}^{SS}\geq 0,\ x_{s_0,j,r}^{SS} = z_{s_0,j,r}^{SS} = 0 \qquad \text{for } i\in J,\ j\in Jr_i,\ r\in R^*$$
$$\bar{x}_{ijr}^{SS}\geq 0,\ \bar{x}_{s_0,j,r}^{SS} = 0 \qquad \text{for } i\in J,\ j\in Jr_i,\ r\in NR$$
$$x_{ijr}^{EE}\geq 0,\ x_{i,e_0,r}^{EE} = z_{i,e_0,r}^{EE} = 0 \qquad \text{for } i\in J\cup\{s_0\},\ j\in Jr_i-\{e_0\},\ r\in R^*$$
$$\bar{x}_{ijr}^{EE}\geq 0,\ \bar{x}_{i,e_0,r}^{EE} = 0 \qquad \text{for } i\in J\cup\{s_0\},\ j\in Jr_i-\{e_0\},\ r\in NR$$
$$x_{ijr}^{SE}\geq 0,\ x_{s_0,j,r}^{SE} = x_{i,e_0,r}^{SE} = x_{s_0,e_0,r}^{SE} = z_{s_0,j,r}^{SE} = z_{i,e_0,r}^{SE} = z_{s_0,e_0,r}^{SE} = 0 \quad \text{for } i\in J,\ j\in Jr_i-\{e_0\},\ r\in R^*$$
$$\bar{x}_{ijr}^{SE}\geq 0,\ \bar{x}_{s_0,j,r}^{SE} = \bar{x}_{i,e_0,r}^{SE} = \bar{x}_{s_0,e_0,r}^{SE} = 0 \qquad \text{for } i\in J,\ j\in Jr_i-\{e_0\},\ r\in NR$$
$$\alpha_{ir}\geq 0,\ \beta_{ir}\geq 0 \qquad \text{for } i\in J,\ r\in R^* \tag{30}$$

5 Cost-oriented problem RCMPSPTC

Since large transfer times do not necessarily correlate with high transfer cost, cost effects of resource transfers must not be neglected. Hence, we extend the presented models by this aspect.

5.1 Problem description

In the resource constrained multi-project scheduling problem with transfer times and costs (RCMPSPTC), we assume job duration as well as resource availability for all resource types to be fixed. Based on the RCPSP, finishing times for all jobs must be determined such that the total cost resulting from this schedule is minimised. We consider **cost directly caused by resource transfers**, e.g., the cost of transporting a concrete mixer from one site to another on a lorry. The lorry consumes fuel and the driver must be paid. Maybe it is even a heavy load transport which requires additional escort and permission. Such variable transfer cost may arise per period and/or transferred unit. Since transfer times are known constants, it is possible and sufficient to consider the variable transfer cost rate CT_{ijr}^{var} per unit of resource r transferred from job i to j. Additionally, fixed transfer cost CT_{ijr}^{fix} emerge, when a transfer causes expenses which are independent of the transferred amounts and the transfer duration.

Moreover, we assume that each renewable resource unit waiting for job execution, brings about **idle cost** CI_r per period of idleness. These are opportunity cost since resource r might be used otherwise productively during the time it is waiting for execution of job j after being assigned to a transfer from job i to j. Additionally, 1st tier resources are idle during a transfer and, thus, cause idle cost. One could integrate this special component of idle cost into the variable transfer cost rate CT_{ijr}^{var}. As it is only an accounting matter to typify the cost as part of transfer or separate idle cost, we add it separately by using CI_r to keep model complexity low. Idle cost for a resource q does not arise when it supports another resource $r \in sup(q)$ during a transfer because resource q is assumed to contribute productively to project progress. Obviously, this type of cost is irrelevant for non-renewable resources $r \in NR$, because they would be consumed when used otherwise. An example for a renewable resource causing idle cost is a team member that is assigned to a job together with a colleague but must wait until his colleague has finished his previous job. If the employee could work productively outside the multi-project, e. g., for non-project work in his marketing department, instead, opportunity cost would arise.

A last cost category to be integrated is **delay penalty**, i. e., penalties that have to be paid when contracted due dates (individual due dates dd_p for the projects p and/or single due date dd for multi-project) cannot be met. Project delay can be separated in positive delay δ_p^+ (tardiness, lateness) or negative delay δ_p^- (earliness). Penalties CD_p^+ per period of earliness may, e.g., reflect that resulting products must be stored when provided too early. The penalisation of tardiness (CD_p^-) of projects is obviously reasonable and very common in project management. A prominent example is the Toll Collect project in Germany. A penalty of 250.000 € up to 500.000 € per day of delay were contracted. If no external due dates are set, the length of the critical path (CP) can be used as a surrogate. In this case, penalising tardiness is equivalent to minimising the project duration if this cost component was considered separately. Earliness of a project could not occur in such a scenario.

Cost for resource usage and consumption of jobs are not taken into account since they cannot be influenced as job duration, resource requirements and capacities are fixed parameters.

5.2 Cost orientated model for RCMPSPTTC

Parameters:

CT_{ijr}^{var}	variable cost for transferring of one unit of resource $r \in R^*$ from job $i \in J \cup \{s_0\}$ to $j \in Jr_i$
CT_{ijr}^{fix}	fixed cost for a transfer of resource $r \in R^*$ from job $i \in J \cup \{s_0\}$ to $j \in Jr_i$
CI_r	opportunity cost for one idle unit of resource $r \in R$ per period of idleness
CD_p^+, CD_p^-	tardiness and earliness penalty for project $p \in P$ per period of positive or negative delay, without index: delay penalty for multi-project, $CD_p^+, CD_p^- \geq 0$
dd_p, dd	due date of project p (if no due date is agreed $dd_p = LB1_p$ is applied), delay of multi-project

Variables:

$\bar{x}_{ijr}^k$	amount of 2nd tier resource $r \in R^*$ used or consumed during a transfer of 1st tier resources from job $i \in J \cup \{s_0\}$ to $j \in Jr_i \cup \{e_0\}$ by transfer type $k \in S$
δ_p^+, δ_p^-	positive (tardiness) and negative (earliness) delay of project $p \in P$, δ^+, δ^-: delays of multi-project
TTC	total transfer cost of the multi-project
TIC	total cost for idle resources in the multi-project
TDC	total cost caused by project delays within the multi-project or by multi-project delay

Once again, the time oriented constraints (2) – (4) with corresponding variable definitions (10) need to be introduced into the model. Moreover, constraints (14) – (30) are part of the cost orientated model as well.

- To calculate cost components TTC in (35), TIC in (36) and TDC in (37) correctly, amounts of actually supporting resources must not only be computed for non-renewable 2^{nd} tier resources (see (21)) but also for renewable supporting resources:

$$\sum_{r \in sup(q)} \mu_q^r \cdot \left\lceil \frac{\Delta_{ijr}}{\Delta_{ijr} + \varepsilon} \right\rceil \cdot x_{ijr}^k = \bar{x}_{ijq}^k \qquad \text{for } i \in J \cup \{s_0\},\ j \in Jr_i,\ q \in R,\ k \in S \tag{31}$$

- The deviations (earliness and tardiness) of the project duration(s) from the agreed due date(s) have to be measured for the multi-project approach or the single-project approach:

$$F_{e_p} - dd_p = \delta_p^+ - \delta_p^- \text{ for all } p \in P \quad \text{or} \quad F_{e_0} - dd = \delta^+ - \delta^- . \tag{32}$$

- According to the applied approach the variables measuring the positive or the negative delay must take non-negative values. The same is true for resource usage/consumption:

$$\delta_p^+, \delta_p^- \ge 0 \quad \text{or} \quad \delta^+, \delta^- \ge 0 \quad \text{for } p \in P, \quad \bar{x}_{ijq}^k \ge 0 \text{ for } i \in J \cup \{s_0\},\ j \in Jr_i,\ r \in R^*,\ k \in S \tag{33}$$

The cost oriented *objective function* minimises the sum of transfer, idle and penalty cost.

$$\text{Minimise } \Phi(*) = TTC + TIC + TDC \tag{34}$$

Total transfer cost TTC is given by the cost of all scheduled resource transfers. The variable cost of a transfer from job i to j depend on the amount x_{ijr}^k of all resources $r \in R^*$ transferred excluding amounts $\bar{x}_{ijr}^k$ of used or consumed 2^{nd} tier resources, because their cost are assumed to be a component of the transfer cost rate of the supported resource. Fixed transfer cost CT_{ijr}^{fix} arise only once when a transfer of a resource r from job i to j actually takes place. We assume fixed cost of 2^{nd} tier resources not being integrated into the fixed cost of the supported resources as it is only an accounting matter of allocating these cost to resource types.

$$TTC = \sum_{i \in J \cup \{s_0\}} \sum_{j \in Jr_i} \sum_{r \in R^*} \left(CT_{ijr}^{var} \cdot \sum_{k \in S} (x_{ijr}^k - \bar{x}_{ijr}^k) + CT_{ijr}^{fix} \cdot \sum_{k \in S} z_{ijr}^k \right) \tag{35}$$

Opportunity cost TIC for idle renewable resources arise per unit and per period of idleness. The total supply of renewable resource $r \in R$ is given by $a_r \cdot F_{e_0}$. The resource is used productively during job execution and when it supports a resource transfer of another resource $s \in sup(r)$.

$$TIC = \sum_{r \in R} CI_r \cdot \left(a_r \cdot F_{e_0} - \sum_{i \in J} u_{ir} \cdot d_i - \sum_{i \in J \cup \{s_0\}} \sum_{j \in Jr_i} \sum_{s \in sup(r)} \mu_{rs} \cdot \Delta_{ijs} \cdot x_{ijs}^k \right) \tag{36}$$

For the multi-project approach, the total delay penalty TDC is made up of the earliness or tardiness and the arranged penalty rates per period of delay. In the single-project approach the sum of weighted earliness and tardiness values is to be computed.

$$TDC = CD^+ \cdot \delta^+ + CD^- \cdot \delta^- \quad \text{or} \quad TDC = \sum_{p \in P} (CD_p^+ \cdot \delta_p^+ + CD_p^- \cdot \delta_p^-) \tag{37}$$

6 Computational Experience

In the following, we conduct two experiments. At first, we compare the managerial approaches of handling resource transfers as described in Section 2.1. Second, we examine the computational capabilities of the mathematical models defined in Section 3.2 and Section 4.2.

All tests have been performed on a computer with Intel Pentium 4 Processor and 1 GB RAM.

6.1 Comparing managerial approaches

In Section 2, we have argued that it is important to consider resource transfers in an explicit manner by a resource-transferring approach instead of neglecting or strictly restricting resource transfers. In order to support this statement, we perform a simple but meaningful experiment.

In either case, the considered scheduling problem is solved heuristically. There are two reasons: the inability of solving large instances to optimality and the common managerial use of (rather simple) heuristics. Thus, we choose priority rule based construction heuristics applying the serial and the parallel scheduling scheme. For RCMPSPTT, the parallel scheme can be adapted from the RCPSP in a straightforward manner. Yet, adapting the serial scheme is not trivial since it must be ensured that already scheduled resource transfers remain feasible when a new job is inserted. A time-based and a resource-based version of the serial scheme are applicable. In addition to the job rule of RCPSP heuristics, which builds the activity list, several further priority rules are necessary which decide how resources are made available for a job to be scheduled. For details and the classification scheme of heuristics used in Table 1 below see Krüger and Scholl (2007).

Every tested approach is applied to 100 multi-project instances of RCMPSPTT-1 (cf. Krüger and Scholl 2007). Each instance consists of five projects, which have been randomly chosen from the well-known Patterson data set, summing up to 93 to 204 jobs with 3 resource types (Patterson 1984). The common capacity a_r of any resource $r \in \{1, 2, 3\}$ in the multi-project is sampled uniformly from the interval $[\max\{\max\{u_{jr} | j \in J'\}, \min\{a_{pr} | p \in P\}\}; \sum_{p=1}^{5} a_{pr} - \max\{a_{pr} | p \in P\}]$ with a_{pr} denoting the capacity of resource r in the original project p. The interval borders ensure existence of a feasible solution and that the projects actually compete for the resources. Transfer times are generated randomly depending on minimal and maximal job durations and considering triangular inequalities. To focus on the critical transfers between projects, the transfer times between jobs of the same project are set to zero.

Three planning concepts are applied considering the managerial approaches in Section 2.1:

- **Transfer-neglecting approach (TN):** Step (1): An RCPSP instance, which is obtained by setting all transfer times to zero, is solved with an RCPSP heuristic. Step (2): A repair mechanism is used to simulate that transfer times just arise when executing the transfer-ignoring schedule. It means that a project manager must react and include the times when they occur. This is done by applying the (time based) serial scheduling scheme of Krüger and Scholl (2007) to the activity list, which contains all jobs in non-decreasing order of the starting times of the RCPSP schedule. Thus, following the originally planned order, jobs are delayed.
- **Transfer-reducing approach (TR):** Each resource unit of a type r that flows into a project p is allocated to it until the last job of p that requires r is finished. Afterwards, it can be transferred to another project. This restriction of resource transfers is observed when applying the heuristics with a rather simple modification of the scheduling schemes.
- **Transfer-using (TU):** The RCMPSPTT instances are solved as defined in Krüger and Scholl (2007) to reflect that resource transfers are allowed and planned explicitly.

In each case, the scheduling schemes can be combined with numerous priority rules. For each managerial approach, we select the best performing rule combination to make a fair comparison. Both, the single- and multi-project approach are considered using the multi-project duration increase MPDI (in %) and mean project delay MD (in time units) as performance indicator, respectively. Table 1 summarises the results based on average values for the data set.

<table>
<tr><th></th><th></th><th>Transfer-neglecting (TN)</th><th>Transfer-reducing (TR)</th><th>Transfer-using (TU)</th></tr>
<tr><td rowspan="2">multi-project</td><td>best heuristic</td><td>1. (par|MinSLK_SP(dyn)|-|-|-)
2. (tbser|ActList|minGAP|bwd|maxFlow)</td><td>(par|MinLFT_MP|minGAP|bwd|-)</td><td>(par|MinLFT_MP|minGAP|bwd|-)</td></tr>
<tr><td>∅MDI</td><td>193.09 (143%)</td><td>176.27 (131%)</td><td>135.04 (100%)</td></tr>
<tr><td rowspan="2">single-project</td><td>best heuristic</td><td>1. (par|SASP|-|-|-)
2. (tbser|ActList|minGAP|bwd|maxFlow)</td><td>(par|SASP|minGAP|bwd|-)</td><td>(par|SASP|minTT|bwd|-)</td></tr>
<tr><td>∅MD</td><td>56.48 (146%)</td><td>45.26 (117%)</td><td>38.74 (100%)</td></tr>
</table>

Table 1. Best performing heuristics and average values of performance indicators

Some further explanations are necessary:

- In the first step of applying TN, the RCPSP instance is solved by the best RCPSP heuristic (parallel scheme with dynamic minimal slack rule). This rule combination is contained in the heuristic rule set for RCMPSPTT of Krüger und Scholl (2007) by simply omitting transfer optimising rules as indicated in the classification tuple in Table 1 by '-'. Secondly, the obtained schedule is transformed into an activity list which serves as input for the (time-based) serial scheduling scheme using the best set of transfer optimising priority rules. This leads to a best possible (heuristic) repair of the transfer-neglecting schedule assuming that this ideal reaction also occurs in reality. If the repair scheme used random priorities, simulating an improvising schedule execution, we would even get ∅MDI = 253.47 and ∅MD = 73.80 .
- When applying TR, we have to face possible deadlocks due to the restricted transfer options. Thus, it sometimes occurs that two projects A and B have been already started, A still blocking a resource 1 and B still blocking a resource 2. If B also requires resource 1 and A requires resource 2, the deadlock will not be resolvable. In our experiment, this situation occurred for 18 and 12 instances in the multi- and the single-project approach, respectively. In these cases, other promising rule combinations have been additionally applied to find at least a feasible solution for the remaining 18 and 12 instances. This succeeded in each case.

The results clearly show that ignoring resource transfers (TN) is the worst that can be done. The average value of MDI is 43% and that of MD is 46% higher than the ones obtained when transfers are planned while scheduling (TU). Strictly allocating resources to projects (TR) is slightly better than ignoring transfers (TN) but also not competitive. It increases MDI on average by 31% and MD by 17% compared to the results of transfer-using scheduling (TU). To summarise, if resource transfers are possible, they should always be taken into account explicitly in project scheduling.

6.2 Experiments with the mathematical models

In order to evaluate the capability of solving the RCMPSPTT by means of our models with standard MIP-solvers, we perform experiments with the models implemented and solved by XPress MP using Mosel modelling language (version 1.6.3).

Considering the basic model RCMPSPTT-1 (cf. Section 3.2) first, we generated two data sets by ProGen (Kolisch et al. 1995). A first set contains 10 small instances with 10 jobs each (J10 basic). A second set comprises 10 instances with 20 jobs each (J20 basic).

NC	RF	RS	Instance
1.5	0.5	0.2	1, 2
		0.5	3, 4
	1.0	0.2	5, 6
		0.5	7, 8
		0.7	9, 10

Table 2. Parameter combinations for J10 and J20

The small instances are made up of n = 10 real jobs with a duration from 1 to 10. Four resource types are required at a maximum with requirements from 1 to 10. Within the set 5 different parameter combinations for network complexity (NC), resource strength (RS) and resource factor (RF) are chosen (see Table 2). For definitions of these parameters confer to Kolisch et al. (1995). For each parameter combination two instances are created. Transfer times are determined randomly considering minimal and maximal job durations as well as the triangular inequalities. The second data set uses the same parameter settings with n = 20. Obviously, the constructed instances are too small to evaluate multi-projects. Hence, only the single-project approach is tested. To simplify the analysis, project duration is used as objective, which is equivalent to the multi-project duration increase (see (12)) for the single-project approach.

The instances of J10 and J20 have been solved with XPress MP imposing a time limit per instance of TL1 = 300 s as well as TL2 = 3000 s. The findings are summarised in Table 3.

	10 jobs (J10 basic)						**20 jobs (J20 basic)**					
	Duration in days			**Computing time in s**			**Duration in days**			**Computing time in s**		
	XPress model		HF	XPress model		HF	XPress model		HF	XPress model		HF
Instance	TL1	TL2		TL1	TL2		TL1	TL2		TL1	TL2	
1	33		34	1.05		0.05	–	–	53	300	3000	0.03
2	34		34	0.11		0.02	–	(70)[a]	62	300	3000	0.03
3	31		31	1.34		0.02	43		43	18.36		0.02
4	35		35	0.72		0.02	–	–	71	300	3000	0.02
5	–	–	61	300	3000	0.02	–	–	104	300	3000	0.06
6	–	–	64	300	3000	0.02	–	–	81	300	3000	0.02
7	47		47	108.63		0.02	–	43	48	300	1180.78	0.08
8	27		27	14.88		0.02	–	–	49	300	3000	0.02
9	–	–	36	300	3000	0.02	38		40	9.59		0.03
10	(72)[a]	(42)[a]	42	300	3000	0.02	–	66	66	300	1735.27	0.03

Table 3. Results for small instances with basic model

a. Feasible solution obtained without optimality proved

For only 7 out of 10 instances with 10 jobs a feasible solution could be found within 300 s, and only six instances were solved optimally. Increasing the time limit to 3000 s has only an effect for one instance. Instances with resource factor 1.0 are found to be most complex, especially if combined with low resource strength (see instances 5 and 6), because in this case, all jobs need all resource types and resources are very scarce.

When the number of jobs rises and data set J20 is tested, the performance of XPress MP becomes even worse. Within 300 s only for two instances optimal solutions can be found. When time limit rises two more instances are solved optimally and an additional instance is solved feasibly. Applying the best (or any other reasonable) priority rule based procedure out of the heuristic framework presented in Krüger and Scholl (2007), feasible solutions can be determined for all 20 instances within negligible computation times (column HF in Table 3).

The extended model RCMPSPTT-2 (cf. Section 4.2) has been analysed by applying it to the data sets J10 and J20 after having extended them: Two non-renewable resource types are added with sufficient capacity for each project instance. Corresponding resource requirements of each job have been generated by ProGen. Transfer times for the two new resource types are generated randomly such that all triangular inequalities are true. The requirement of transfer supports, i. e., which resource type requires which supporting resource for a transfer at which amount, is determined randomly and, afterwards, altered until feasibility of the problem instance is given.

The results are even poorer than for the basic model: Only two extended J10 instances could be solved optimally and further two instances feasibly within 300 s with the extended model. No extended J20 instance could be solved within the given time limit.

The cost model has not been tested since no further insights are expected from this analysis of this even more complex model.

7 Summary and conclusion

We analysed the concept of resource flows subject to transfer times and cost as an extension of the RCPSP and the RCMPSP. First, we presented a managerial framework for handling resource transfers based on a classification of resource transfers and resource roles in those transfers. As could be demonstrated experimentally, the presented new approach of explicitly planning resource transfers instead of ignoring or unnecessarily restricting transfers is of great importance for practical project and resource planning and should be integrated in project management software, because project delays and non-negligible additional expenses arise from these transfers.

To approach these new aspects in a thorough manner, we systematically formulated several problem extensions with different resource types and roles and different objective functions. For each problem version, a mixed-integer model was developed to study its details and structure in a formal and systematic manner. Preliminary computational experiments led to the conclusion that applying standard solvers cannot be recommended. Specialised solution procedures, even heuristic approaches, can deliver better results. A first heuristic approach to tackle the basic problem has been presented by Krüger and Scholl (2007). However, further research is necessary to develop procedures for the extended problem.

References

Aldowaisan, T.; Allahverdi, A. and J.N. Gupta (1999): A review of scheduling research involving setup considerations. Omega 27/2, pp. 219-239.

Artigue, C.; Michelon, P. and S. Reusser (2003): Insertion techniques for static and dynamic resource-constrained project scheduling. European Journal of Operational Research 149/2, pp. 249-267.

Brucker, P.; A. Drexl, R. Möhring, K. Neumann and E. Pesch (1999): Resource-constrained project scheduling: Notation, classification, models, and methods. European Journal of Operational Research 112/1, pp. 3-41.

Debels, D. and M. Vanhoucke (2006): Pre-emptive resource-constrained project scheduling with setup times. Working paper 2006/391, Universiteit Gent.

Demeulemeester, E. L. and W. S. Herroelen (2002): Project scheduling: A research handbook. Kluwer Academic Publishers, Boston.

Dodin, B. and A.A. Elimam (1997): Audit scheduling with overlapping activities and sequence-dependent setup costs. European Journal of Operational Research 97/1, pp. 22-33.

Dumond, J. and V. A. Mabert (1988): Evaluating project scheduling and due date assignment procedures: An experimental analysis. Management Science 34/1, pp. 101-118.

Fendley, L. G. (1968): Towards the development of a complete multi-project scheduling system. Journal of Industrial Engineering 19/10, pp. 505-515.

Jans, R. and Z. Degraeve: Modeling industrial lot sizing problems: A review. International Journal of Production Research 46/6, pp. 1619 - 1643.

Kolisch, R. (1995): Project scheduling under resource constraints. Physica, Heidelberg.

Kolisch, R.; Sprecher, A. and A. Drexl (1995): Characterization and generation of a general class of resource-constrained project scheduling problems. Management Science 41/10, pp. 1693-1704.

Krüger, D. and A. Scholl (2007): A heuristic solution framework for the resource constrained multi-project scheduling problem with sequence-dependent transfer times. Jena Research Papers in Business and Economics (JBE) 16/2007, FSU Jena.

Kurtulus, I. and E.W. Davis (1982): Multi-project scheduling: Categorization of heuristic rules performance. Management Science 28/2, pp. 161-172.

Kurtulus, I. S. and S. C. Narula (1985): Multi-project scheduling: Analysis of project performance. IIE Transactions 17/1, pp. 58-66.

Lawrence, S. R. and T.E. Morton (1993): Resource constrained multi-project scheduling with tardy costs: Comparing myopic, bottleneck, and resource pricing heuristics. European Journal of Operational Research 64/2, pp. 168-187.

Lova, A. and P. Tormos (2001): Analysis of scheduling schemes and heuristic rules performance in resource-constrained multiproject scheduling. Annals of Operations Research 102/1-4, pp. 263-286.

Lova, A; Maroto, C. and P. Tormos (2000): A multicriteria heuristic method to improve resource allocation in multiproject scheduling. European Journal of Operational Research 127/2, pp. 408-424.

Mellentien, C.; Schwindt, C. and N. Trautmann (2004): Scheduling the factory pick-up of new cars. OR Spectrum 26/4, pp. 579-601.

Mika, M.; Waligóra, G. and J. Weglarz (2006): Modelling setup times in project scheduling. In: Jósefowska, J. and J. Weglarz (eds.): Perspectives in modern project scheduling. Springer, New York, pp. 131-163.

Neumann (2003): Project scheduling with changeover times - Modelling and applications. In: Proceedings of the international conference on industrial engineering and production management 1, Porto/Portugal, May 26-28, pp. 30-36.

Neumann, K.; Schwindt, C. and J. Zimmermann (2003): Project scheduling with time windows and scarce resources. Springer, Berlin.

Pritsker, L.J.; Watters, A.A.B. and P.M. Wolfe (1969): Multiproject scheduling with limited resources: A zero-one programming approach. Management Science 16/1, pp. 93-108.

Schwindt, C. and N. Trautmann (2000): Batch scheduling in process industries: An application of resource-constrained project scheduling. OR Spektrum 22/4, pp. 501-524.

Schwindt, C. and N. Trautmann (2003): Scheduling the production of rolling ingots: Industrial context, model, and solution method. International Transactions in Operational Research 10/6, pp. 547-563.

Yang, K. K. and C. C. Sum (1993): A comparison of resource allocation and activity scheduling rules in a dynamic multi-project environment. Journal of Operations Management 11/2, pp. 207-218.

Yang, K. K. and C. C. Sum (1997): An evaluation of due date, resource allocation, project release, and activity scheduling rules in a multiproject environment. European Journal of Operational Research 103/1, pp. 139-154.