MASTERARBEIT

Titel der Masterarbeit

A comparison of Mixed Integer Linear Programming models for the Multi-Mode Resource Constrained Project Scheduling Problem

verfasst von

Kornelia Artinger, BSc

angestrebter akademischer Grad

Master of Science (MSc)

Wien, 2015

Studienkennzahl lt. Studienblatt:	A 066 914
Studienrichtung lt. Studienblatt:	Masterstudium Internationale Betriebswirtschaft
Betreuer:	o. Univ.-Prof. Dipl.-Ing. Dr. Richard F. Hartl

ACKNOWLEDGMENT

First of all, I would like to thank my thesis supervisor O.Univ.Prof. Dipl.-Ing. Dr. Richard F. Hartl for granting me the opportunity to write this thesis at his chair and for his valuable feedback during my work.

Moreover I would like to thank Alexander Schnell, MSc for his continuous support, his critical and applicable feedback and the numerous times he helped me working my way through various mathematical and technical issues.

Another tribute goes to my colleagues and friends at the Faculty of Production and Operations Management for motivating me constantly and supporting me throughout the project.

Last but certainly not least I want to thank my parents, not just for staying supportive and optimistic throughout the whole project, but also for showing a great amount of patience and understanding.

Vienna, February 2015

Contents

List of Tables

List of Abbreviations

AoA Activity-on-Arc

AoN Activity-on-Node

DDT Disaggregated Discrete-Time

DT Discrete-Time

FCT Flow-based continuous time

MILP Mixed Integer Linear Programming

MRCPSP multi-mode Resource Constrained Project Scheduling Problem

NC Network Complexity

OOE On-Off-Event-based

RCPSP Resource Constrained Project Scheduling Problem

RF Resource Factor

RS Resource Strength

SEE Start-End-Event-based

SRCPSP Single Mode Resource Constrained Project Scheduling Problem

PSPLIB Project Scheduling Problem Library

1 Introduction

The Resource Constrained Project Scheduling Problem (RCPSP) is a well-known combinatorial optimization problem. It is designed to schedule a set of activities with given durations and resource requirements with respect to precedence relations and limited resources. Due to its numerous practical applications - for example the scheduling of production processes or timetabling - the RCPSP has been in the focus of researchers over the last decades.

It is obvious that the above mentioned problem description is broad and general and thus might not cover all aspects of practical instances. It became therefore necessary to develop not only enhanced solution methods, but also extensions to the basic formulation. One of those adaptations is the consideration of multiple performance modes. As it is often the case with real-world scenarios, activities can be performed in alternative ways influencing both the duration as well as the resources needed for the execution.
According to [Sprecher and Drexl, 1998, p. 1] modes provide

> ”...alternative combinations of resources and belonging quantities employed to fulfill the task related to the activities.”

The complexity of this problem - the so-called multi-mode Resource Constrained Project Scheduling Problem (MRCPSP) - increases, as not only the feasible - if not - optimal schedule has to be determined but also the modes in which the activities should be executed. This last choice now depends on two different trade-offs, namely time-resource and resource-resource. Whereas the first one considers the fact that a mode using a higher resource quantity to perform the activity faster, restrains the resource limitations for other activities, the second represents the possibility of resource substitution. This becomes evident if the growing need for one resource due to a mode choice lowers the need for another resource simultaneously. All of these considerations make the MRCPSP a complex and difficult problem to solve.

The literature review in the following sections will show that already a lot of different solution approaches - both heuristic and exact - have been developed to cover the various RCPSP extensions. As already the standard Single Mode Resource Constrained Project Scheduling Problem (SRCPSP) itself is - according to Blazewicz et al. [1983] - NP-hard in the strong sense, the efficiency of exact solution methods seems to be limited. According to Koné et al. [2011] at most problem instances with up to 60 activities could be solved with exact methods in the single mode case. This statement does not include latest hybrid solution methods like the one introduced by Schutt et al. [2011] which is capable of closing also some instances with 90 or 120 instances.

Despite those current limitations, exact approaches based on the solution of Mixed Integer Linear Programming (MILP) models presented in this thesis should not be neglected. As mentioned by Koné et al. [2011] when it comes to the implementation of RCPSPs in a real-world situation, MILP solvers tend to be one of the few software applications available for companies. This fact makes a standardized and efficient MILP model for the RCPSP useful across sectors and industries and therefore interesting for further research. In their paper Koné et al. [2011] provided an in-depth comparison of different MILP modeling approaches for the SRCPSP, drawing the conclusion that some of those models deliver a good performance on the analyzed benchmark problems. A summary of this paper will be provided in the literature review section. This thesis now aims at enlarging the focus of the analysis to the multi-mode case and prove whether the drawn conclusions are also valid in this respect.

The structure of this thesis is as follows: Chapter 2 describes the SRCPSP and the MRCPSP in greater detail, providing basic notations and further notes on complexity. Chapter 3 will provide the reader with a short and focused literature review on MILP formulations and the MRCPSP as well as some basic literature sources. In Chapter 4 the MILP models will be presented and then implemented and evaluated in Chapter 5. Chapter 6 draws a conclusion and provides some ideas and comments for further research.

2 Problem Description

2.1 General Problem Formulation

As already mentioned in the introductory chapter, the RCPSP consists of finding a feasible schedule of starting times for a set of activities as a result, taking into account resource and precedence constraints.

Generally stated, the problem depicts a project consisting of a set of activities $\{0, 1, ..., n+1\}$ where activities 0 and $n+1$ are the source and sink of the problem. They represent the start and the end of the project as so-called dummy activities. This differentiation led to the introduction of a second activity set by Koné et al. [2011] $A := \{1, ..., n\}$ to consider only the non-dummy activities.

Furthermore a set of k renewable resources $R = \{1, ..., m\}$ is given as well as a related resource capacity B_k. All activities require a certain amount of the renewable resources per period over their processing time, p_i, which is then represented by the variable b_{ik}. The source and sink activities have of course neither a resource requirement nor a processing time, so $p_0 = p_{n+1} = 0$ and $b_{0,k} = b_{n+1,k} = 0$.

The precedence constraints are based on a set E of activity pairs (i, j). Each of the pairs entered in the set represents a precedence relation, as for example $(i, j) \in E$ would mean that activity i has to precede activity j in a feasible schedule. In this thesis only standard precedence relations are considered. For the usage of generalized precedence constraints, the reader is referred to Brucker and Knust [2011] for introductory information.

The project spans over a given scheduling horizon $H = \{0, 1, ..., T\}$ where T is a given upper bound for the project's makespan. A valid assumption for the value of T could be the sum of all activity durations p_i.

In the literature various objective functions are considered to optimize a certain target value. Most common the aim is to minimize the project's makespan. This is also the objective function used in the context of this thesis.

Apart from the set and variable definition, also a few assumptions have to be made in order to complete the problem formulation. Once started, an activity can not be interrupted but has to be finished, preemption is not allowed. Moreover the input data is assumed to be integer and deterministically known.

For the sake of completeness it should also be mentioned that some of the assumptions made above have been relaxed by researchers over the last decades to encounter new solution methods. Some approaches allow activity splitting, therefore neglecting the non-preemption requirement. Other approaches consider multiple projects as an extension or work with different objective functions. The examples here are numerous and the reader is referred to Hartmann and Briskorn [2010] for a comprehensive description and literature review in this respect.

2.2 Multi-mode Problem Formulation

After the previous chapter provided a short summary of the general RCPSP, this one will extend the formulation to cover the multi-mode case. In the MRCPSP not just the starting times of the activities have to be determined but also the mode in which they are performed.

The modes $\{1, ..., M_i\}$ represent different alternatives for the activity's resource usage and processing time. Once chosen, a mode can not be changed anymore, the activity has to be started and finished in the same mode.

As the different modes stand for different resource combinations, it is necessary to comment further on the concept of renewable and nonrenewable resources. Whereas renewable resources are - as indicated by the name - renewed and therefore available again in each time period, nonrenewable resources are restricted with respect to the complete project planning horizon. Examples for renewable resources would be machines or manpower, nonrenewables include money (i.e. total project budget) or energy.

As put by [Brucker and Knust, 2011, p. 6]:

> "...nonrenewable resources are consumed, i.e. when an activity i is processed, the available amount R_k of a nonrenewable resource k is decreased by r_{ik}."

The concept of nonrenewable resources is introduced in this chapter because their presence is only relevant in the multi-mode case. Depending on which mode is chosen the consumption of the nonrenewable resource can differ and therefore influence the solution. In the single mode scenario, the consumption of nonrenewables is fixed and is not influencing the outcome, the capacity is either met or all schedules are infeasible.

To be most accurate there is also a distinction made for so-called doubly-constrained resources, referring for example to a monetary budget that is restricted both per period and for the total project. These resources tend to be split into one renewable and one nonrenewable resource constraint.

Compared to the formulation of the SRCPSP above, the different modes influence both the notation of the activity duration and the resource requirements. The processing time of activity i can now be denoted by p_{im} whereas the resource consumption is given by b_{ikm} to account for the influence of the mode choice.

Furthermore the different sets of resources have to be considered as well. Instead of one set of resources R, now two sets have to be introduced: R_r as the set of renewable resources and R_v as the set of nonrenewable resources. Analogously, the resource consumption of the activities is adapted to b^r_{ikm} and b^v_{ikm} respectively. The resource capacities are also aligned and denoted by B^r_k and B^v_k.

For better traceability it should be mentioned that the notation used in this thesis is derived from the one used by Koné et al. [2011] with some variations according to the resource formulations by Brucker and Knust [2011]. For the dummy activities 0 and $n+1$ it is assumed that no mode has to be chosen as the durations and resource requirements are zero by definition. In order to guarantee a consistency in the precedence relation, source activity 0 is set as the predecessor for all activities without given predecessor. The same is valid for all activities without a successor where the sink activity $n + 1$ is assigned for it.

2.2.1 Network Representation

In order to allow for a better understanding of some computational steps taken later, a short introduction into the network representation of an RCPSP will be provided. There are two main approaches used for network representation, the Activity-on-Arc (AoA) and the Activity-on-Node (AoN) version. Kyriakidis et al. [2012] describe the first one as event based method, with arcs representing activities and nodes as a starting or finishing event. In the AoN approach, on the other hand the activities are depicted by nodes whereas the arcs show the precedence relations between them. As this method seems to be more intuitive to work with, it will be used throughout this thesis. The general formulation of this network would be a precedence graph $G(V, E)$ where V is the set of activities and E as already mentioned above describes the relations between the activities.

2.3 Complexity

The complexity of a problem refers to the efficiency with which it can be solved via certain algorithms. The more complex a problem is, the harder it is to exactly solve it within reasonable time. This is why heuristics tend to be preferred in such cases as solution methods. As Brucker and Knust [2011] state, the efficiency of an algorithm can be measured by calculating the running time needed for solving the problem in a worst-case scenario with a certain input size. In this context so called "easy" and "hard" problems can be distinguished. An "easy" problem can be solved with an algorithm in polynomial time. The notion "NP-hard" on the other hand is used to describe problems that are generally hard to solve to optimality.

Blazewicz et al. [1983] proved that the SRCPSP with the aim of makespan minimization is such an NP-hard problem. In their article the authors introduced an updated classification scheme for RCPSPs with different resource constraint information and tested them regarding their complexity issues. Moreover it is shown that the problem is NP-hard in the strong sense, making it even more difficult to find an optimal solution and that is just describing the single mode case. Optimization problems are NP hard in the strong sense if the related decision problem - in this case the problem whether or not a feasible schedule with an arbitrary deadline D exists - is NP-complete in the strong sense. The term "in the strong sense" implies that even if all numerical parameters of a problem would be bounded by a polynomial of the input size, the problem would remain NP-hard and therefore be hard - if not infeasible - to solve via a pseudo-polynomial algorithm. As the MRCPSP adds another decision, namely the mode choice to the problem, the complexity increases even more. To illustrate this, Kolisch and Drexl [1997b] proved in their article the NP-completeness of the decision problem for the multi-mode case.

3 Literature review

As already mentioned, the RCPSP is a widely discussed topic resulting in a variety of articles, models and benchmarking studies. Keeping this in mind, this chapter should give a short insight into the main contributions made so far in literature. The first part is dedicated to some general categorization papers whereas the second part will introduce several solution approaches for the SRCPSP. This is followed by mentioning the most important contributions about the MRCPSP, focusing on best-performing exact and (meta)heuristic methods.

To start with the reader is referred to an article by Brucker et al. [1999]. The authors introduce a notation and classification scheme for RCPSPs in the style of similar schemes for machine scheduling. They focus on the resource and activity characteristics and the objective function as classification criteria. Furthermore the article gives an overview of solution approaches, both exact and heuristic for the RCPSP and its various extensions, accompanied by computational results. Another orientation for the reader is the survey conducted by Hartmann and Briskorn [2010]. The authors put their focus on the different variations of the RCPSP, considering special cases in terms of resource categories, activity handling and objective functions. Also here the different extensions are complemented with solution approaches and articles from the literature.

Although this thesis is examining different solution methods for the MRCPSP, several articles should be mentioned for the single-mode case as well. As already noted in the previous section, the basis for this work is the article by Koné et al. [2011], which draws a theoretical and computational comparison between different MILP models. Pritsker et al. [1969] introduced the first efficient MILP approach for the RCPSP using a time-indexed formulation. The model has been extended to the multi-mode case by Kolisch and Sprecher [1997]. In order to allow for comparative testing, they also introduced the PSPLIB which contains newly developed benchmarking sets. This library has now become frequently used for benchmark testing and is also the main source of instances used in this thesis. Artigues et al. [2003] used a different approach and introduced a flow network model supplemented by an insertion algorithm via a heuristic. Computational results have shown potential for dynamic planning.

For the sake of completeness, also some of the best-performing algorithms so far for solving the SRCPSP should be mentioned here. One state-of-the-art Branch&Bound method derived by Demeulemeester and Herroelen [1997] is currently the best performing algorithm in this field for up to 30 activities. The respective article describes the efforts to improve an older Branch&Bound version with an emphasis on improving memory usage and computation time. Additionally the new approach benefits from a new lower bound, developed by Mingozzi et al. [1995] taking into account feasible activity subsets. The use of this more efficient lower bound shows a strong impact on the memory and the computation time needed.

For more than 30 activities, an interesting hybrid approach combining Constraint Programming and Boolean satisfiability solving techniques was introduced by Schutt et al. [2011]. [Schutt et al., 2011, p. 278] use:

> "...cumulative constraints with explanation in a lazy clause generation system."

In order to get a full overview over the heuristics and metaheuristics in this area, the reader is referred to a quite recent study by Kolisch and Hartmann [2006] who categorized and evaluated the latest developments for metaheuristics, also mentioning older approaches from their former study found in Hartmann and Kolisch [2000]. The study itself gives an extensive overview over the characteristics, efficiency and capabilities of various solution approaches.

In the context of the MRCPSP several articles should be mentioned. In their paper Zapata et al. [2008] introduced several mathematical models to solve the MRCPSP, one of which uses the notion of event points to indicate start or end times of activities. The main goal was to find formulations to overcome computation problems with increasing time horizons. Although the results itself have not been promising in comparison to other models, the event-based approach has been adapted successfully by Koné et al. [2011] for the single mode case. Apart from above MILP formulations, also the already well-established Branch&Bound algorithms have been extended to the multi-mode case. The survey of Hartmann and Drexl [1998] provides a comparison of several exact algorithms in this field, reviewing different bounding techniques.

To conclude this section, we will state the best-performing algorithms so far. In terms of exact approaches, the Branch&Bound algorithm by Sprecher and Drexl [1998] is the providing the best results for activity sets with 10 and 20 activities found in the PSPLIB. Another (hybrid) exact approach was introduced by Zhu et al. [2006], who suggested a Branch&Cut procedure with an adaptive branching and bound tightening scheme. Also the use of a Genetic Algorithm for finding feasible solutions and upper bounds for the exact approach was mentioned. The approach performed excellently on instances with 20 activities and also managed to close a majority of the 30-activities instances of the PSPLIB, sometimes even improving the best performing heuristics. For instances with 30 activities, several heuristics provide good results, the local search algorithm by Kolisch and Drexl [1997a], a Tabu-Search approach by Nonobe and Ibaraki [1998] and a Simulated Annealing metaheuristic by Bouleimen and Lecocq [2003].

A very recently published survey gives a good overview over the different metaheuristics used for solving the MRCPSP. For the evaluation of the different approaches Van Peteghem and Vanhoucke [2014] generate new datasets with 50 and 100 activities which guarantee for example only efficient modes and at least one feasible solution per instance. They also give a very good examination of the influence that the different resource parameters have on the algorithms' performance. The metaheuristic that performed best in their comparison is the one by Van Peteghem and Vanhoucke [2011] dealing with a so-called scatter search algorithm. This population based metaheuristic is then accelerated with different procedures taking into account the resource scarceness parameters to guarantee a more efficient search. The method also performed very well on the instance set with 30 activities from the PSPLIB.

These articles represent of course only a fraction of the literature about the SRCPSP and the MRCPSP. The review covers those articles that have been crucial to the development of the models used in the thesis and other articles that might be of interest for the reader. As already mentioned the main research focus of this thesis is to evaluate the different MILP formulations found in Koné et al. [2011] with consideration of multiple execution modes. In order to give the reader a full understanding of the models, the differences and main characteristics, the following chapter is dedicated to present and evaluate the models in their extended versions.

4 Mixed Integer Linear Programming models – The Multi-Mode case

4.1 Introductory comments

This chapter presents and explains the mathematical models used in this work. All of the below presented approaches are extended versions of already existing formulations for solving the SRCPSP presented by Koné et al. [2011]. The models described can be roughly divided into three categories, the **Time-Indexed**

formulations, the **Flow-based formulation** and the **Event-based formulations**. All subsections will be containing a reference to the original model, a short introduction of the main characteristics, the mathematical formulation itself and a respective explanation. As the models tend to become vast and hard to oversee, a certain subdivision of the constraints in terms of resource, precedence and other relations is introduced to facilitate the understanding. The last part of this chapter contains a short review and comparative analysis of the models.

It should be mentioned here that as a preprocessing step, the earliest and latest starting times for every activity is determined. This ensures that the timing of an activity is not bound to the whole time horizon but just to its very own starting time window $[ES_i, LS_i]$. This allows us to shrink the initial time domain of the variables. The method used for the calculation of the Earliest (ES_i) and Latest (LS_i) Starting Time of activity i is based on calculating the longest path in the precedence network. Here the Bellman-Ford algorithm for computing shortest paths was used with negative arc weights (activity durations). As each activity i had different mode durations, the shortest duration was taken in order to create the largest possible starting time window. Intuitively described this algorithm is based on the relaxation of the resource constraints, only taking into account the precedence relations to derive the starting time window.

4.2 Time-indexed models

4.2.1 Discrete-time formulation (DT)

The discrete time-indexed formulation was first introduced by Pritsker et al. [1969] as an attempt to use a 0-1 formulation to model the RCPSP. Although there were already several 0-1 models in place, none of them succeeded in taking resource restrictions into consideration. The model is based on the binary decision variable x_{it} indicating if activity i starts at time t or not. In order to account for the multi-mode case this decision variable can be easily extended to x_{imt}, a variable becoming 1 if an activity i starts at time t in mode m and 0 if this is not true. The model itself has been adapted as follows: (see also Kolisch and Sprecher [1997])

Objective function

$$\min \sum_{m=1}^{M_i} \sum_{t=ES_i}^{LS_i} t \cdot x_{n+1,m,t} \tag{1}$$

Precedence constraints

$$\sum_{n=1}^{M_j} \sum_{t=ES_j}^{LS_j} t \cdot x_{jnt} \geq \sum_{m=1}^{M_i} \sum_{t=ES_i}^{LS_i} (t + p_{im}) \cdot x_{imt} \qquad \forall (i,j) \in E \tag{2}$$

Renewable resource constraints

$$\sum_{i=1}^{n} \sum_{m=1}^{M_i} b^r_{imk} \cdot \sum_{\tau=max(ES_i, t-p_{im}+1)}^{min(LS_i,t)} x_{im\tau} \leq B^r_k \qquad \forall t \in H, k \in R_r \tag{3}$$

Nonrenewable resource constraints

$$\sum_{i=1}^{n} \sum_{m=1}^{M_i} b^v_{imk} \cdot \sum_{t=ES_i}^{LS_i} x_{imt} \leq B^v_k \qquad \forall k \in R_v \tag{4}$$

Activity and mode constraints

$$\sum_{m=1}^{M_i} \sum_{t=ES_i}^{LS_i} x_{imt} = 1 \qquad \forall i \in A \cup \{n+1\} \tag{5}$$

$$x_{010} = 1 \tag{6}$$

$$\sum_{m=1}^{M_i} x_{imt} \leq 0 \qquad \forall i \in A \cup \{n+1\}, t \in H \setminus \{ES_i, LS_i\} \tag{7}$$

Decision variable definitions:

$$x_{imt} \in \{0,1\} \qquad \forall i \in A \cup \{n+1\}, t \in H, m = 1, ..., M_i \tag{8}$$

The objective function (1) minimizes the makespan of a project, represented by the point in time t at which the last (sink) activity without any duration is started within its starting time window ($x_{n+1} = 1$).

Constraints (2) ensure that two activities (i, j) that are part of the precedence matrix E stick to their precedence relationship. The point in time t where activity j starts needs to be bigger or equal to the finishing time t of activity i (calculated by summing up the starting time and the duration of activity i in mode m).
Constraints (3) and (4) guarantee that the resource consumption of an activity does not exceed the capacity of the available resources. Whereas (3) considers the renewable resource capacity by taking into account the resource consumption per resource k and time period t, (4) indicates the nonrenewable situation by summing up the consumption over the entire time horizon.
Activity and mode constraints (5) ensure that each activity is only started once in exactly one mode, whereas constraints (6) set the starting time of the source activity to 0. Constraints (7) can be used to narrow the use of the decision variables explicitly to their start time windows by setting the variable in at any point in time outside those windows to 0. Constraints (9) characterize the decision variable as binary.

4.2.2 Disaggregated discrete-time formulation (DDT)

The second time-indexed model attempts to use a different formulation of the above precedence constraints. The model was introduced by Christofides et al. [1987] and is - as indicated by the name - disaggregating the restriction on the precedence relationship.
The below formulation used has been adapted from the model developed by Artigues [2013] and extended to the multi-mode case.

$$\sum_{m=1}^{M_i} \sum_{\tau=ES_i}^{t-p_{im}} x_{im\tau} - \sum_{n=1}^{M_j} \sum_{\tau=ES_j}^{t} x_{jn\tau} \geq 0 \qquad \forall (i,j) \in E, t \in \{ES_i, LS_i\} \tag{9}$$

This formulation in turn requires more constraints than the Discrete-Time (DT) formulation with the same amount of binary decision variables.

4.3 Flow-based continuous-time model (FCT)

The flow-based approach elaborated by Artigues et al. [2003] moves away from using time as an index on to a continuous time formulation. Instead of viewing renewable resources as being available in a certain capacity per time period, [Artigues et al., 2003, p. 250] approach them

> "...by defining each resource k as the union of R_k resource units..."

where R_k stands for the resource capacity. In other words, each of the units in this union, can only be allocated to one activity i at a time and is after the completion of this activity passed on to another activity that requires it. The source and sink activities are here also functioning as resource source and sink for the flow model, by setting $\tilde{b}^r_{0,k,1} = \tilde{b}^r_{n+1,k,1} = B^r_k$ and $\tilde{b}^r_{ikm} = b^r_{ikm}$ for $i \neq 0, n+1$ is an adapted notation for the resource consumption.

To make this model work, a wider range of decision variables is needed. A continuous starting-time variable S_i, to indicate the starting time of each activity, a sequential binary variable x_{ij} to state if a precedence relationship exists between i and j, a continuous flow variable f_{ijk} to represent the amount of renewable resource k transferred from activity i to activity j and a binary mode variable y_{im}, to show the chosen mode of activity i. This last decision variable has been added to account for the multi-mode case. The model has been implemented as follows:

Objective function:

$$\min S_{n+1} \tag{10}$$

Precedence constraints:

$$S_j - S_i \geq -M_{ij} + \sum_{m=1}^{M_i} (p_{im} + M_{ij}) \cdot x_{ij} \cdot y_{im} \qquad \forall (i,j) \in (A \cup \{0, n+1\})^2 \tag{11}$$

Renewable resource constraints:

$$f_{ijk} \leq x_{ij} \cdot \min(\sum_{m=1}^{M_i} \tilde{b}^r_{ikm} \cdot y_{im}, \sum_{n=1}^{M_j} y_{jn} \cdot \tilde{b}^r_{jkn})$$
$$\forall (i,j) \in (A \cup \{0\} \times A \cup \{n+1\}), k \in R_r \tag{12}$$

$$\sum_{j \in A \cup (0,n+1)} f_{ijk} = \sum_{m=1}^{M_i} \tilde{b}^r_{ikm} \cdot y_{im} \qquad \forall i \in A \cup \{0, n+1\}, k \in R_r \tag{13}$$

$$\sum_{i \in A \cup (0,n+1)} f_{ijk} = \sum_{n=1}^{M_j} \tilde{b}^r_{jkn} \cdot y_{jn} \qquad \forall j \in A \cup \{0, n+1\}, k \in R_r \tag{14}$$

$$f_{n+1,0,k} = B^r_k \qquad \forall k \in R_r \tag{15}$$

Nonrenewable resource constraints:

$$\sum_{i \in A} \sum_{m=1}^{M_i} b^v_{ikm} \cdot y_{im} \leq B^v_{ik} \qquad \forall k \in R_v \tag{16}$$

Activity and mode constraints:

$$x_{ij} + x_{ji} \leq 1 \qquad \forall (i,j) \in (A \cup \{0, n+1\})^2, i < j \tag{17}$$

$$x_{ik} \geq x_{ij} + x_{jk} - 1 \qquad \forall (i,j,k) \in (A \cup \{0, n+1\})^3 \tag{18}$$

$$x_{ij} = 1 \qquad \forall (i,j) \in T(E) \tag{19}$$

$$\sum_{m=1}^{M_i} y_{im} = 1 \qquad \forall i \in (A \cup \{0, n+1\}) \tag{20}$$

Decision variable definitions:

$$f_{ijk} \geq 0 \qquad \forall (i,j) \in (A \cup \{0, n+1\})^2, k \in R \tag{21}$$

$$ES_i \leq S_i \leq LS_i \qquad \forall i \in (A \cup \{n+1\}) \tag{22}$$

$$S_0 = 0 \tag{23}$$

$$x_{ij} \in \{0, 1\} \qquad \forall (i,j) \in (A \cup \{0, n+1\}^2) \tag{24}$$

$$y_{im} \in \{0, 1\} \qquad \forall i \in (A \cup \{0, n+1\}) \tag{25}$$

The objective function (10) minimizes the starting time of the sink activity, therefore minimizing the total makespan.

The precedence constraints (11) indicate that if i precedes j ($x_{ij} = 1$), the starting time of j, S_j, needs to be bigger or equal to the starting time of i plus the mode dependent activity duration. This constraint shows the flaw of this formulation as the introduction of a big-M variable is necessary. M_{ij} is calculated as the difference between ES_i and LS_j to keep it as low as possible. Apart from that the multiplication of the two decision variables x_{ij} and y_{im} makes additional linearization necessary. This was achieved by simply introducing a new binary variable $z_{ijm} = x_{ij} \cdot y_{im}$ and several additional constraints to account for these non-linearities:

$$z_{ijm} \leq y_{im} \qquad \forall (i,j) \in A^2, m \in 1...M_i$$

$$\sum_{m=1}^{M_i} z_{ijm} \leq x_{ij} \qquad \forall (i,j) \in A^2$$

$$z_{ijm} \geq x_{ij} + y_{im} - 1 \qquad \forall (i,j) \in A^2, m \in 1...M_i$$

The constraints (12) - (15) regulate the flow of renewable resources. (12) sets the pace for the flow, by linking f_{ijk} to the precedence variable x_{ij}. The flow variable is only positive if a link between activities i and j exists. If yes, the flow is limited to the minimum of resources needed by either activity. Also here a linearization is needed in the calculation phase. This is continued by constraints (13) and (14) which guarantee that the flow from activity i to all succeeding activities is exactly the resource requirement of i in the chosen mode and that the flow from all preceding activities i to j is exactly the resource requirement of j. (15) on the other hand sets the resource flow from the sink to the source activity to the total capacity. This guarantees the completion of the flow cycle. In comparison the nonrenewable resource constraint (16) is straightforward, limiting the total consumption of each resource to the total capacity.

Constraints (17) to (20) are activity and mode related. (17) should avoid any cycles between activities i and j, whereas (18) guarantees the transitivity. If activities i and j and activities j and k are connected, then there is also a link between i and k. Constraint (19) sets all possible connections in the transitive closure TE to 1, opening all possible links for a resource flow. The transitive closure was calculated with the help of the Floyd-Warshall algorithm in the preprocessing phase. Constraint (20) ensure that only one mode per activity is chosen. The remaining constraints (21) to (25) characterize the decision variables.

4.4 Event-based continuous-time models

A relatively new approach is the usage of so-called events as main indexation. An event corresponds to the start or the end time of an activity. When considering a left-shifted schedule without any time lags between the activities, it is obvious that the start time of an activity is either 0 or the end time of a preceding activity. This reduces the number of events accordingly to the number of non-dummy activities plus one. This event set $\mathcal{E} = \{0, ..., n\}$ remains independent from the time horizon, making it an attractive approach with an increasing time horizon. Koné et al. [2011] distinguish between two different methods in this context, the Start-End-Event-based (SEE) formulation and the On-Off-Event-based (OOE) formulation.

4.4.1 Start-end Event-based formulation (SEE)

The SEE formulation is based on the work of Zapata et al. [2008], but was developed further by Koné et al. [2011]. It involves two sets of binary variables and two sets of continuous ones. The binary variables x_{iem} indicate if an activity i is started at an event e in a certain mode m, the variables y_{iem} indicate the ending of the same. The set t_e represents the event date as each event gets attributed a point in the time schedule. Last but not least, r_{ek} is the renewable resource requirement after each event e.
Objective function:

$$\min t_n \tag{26}$$

Precedence constraints:

$$\sum_{m=1}^{M_i} \sum_{e'=e}^{n} y_{ie'm} + \sum_{n=1}^{M_j} \sum_{e'=0}^{e-1} x_{je'n} \leq 1 \qquad \forall (i, j) \in E, e \in \mathcal{E} \tag{27}$$

Renewable resource constraints:

$$r^r_{0k} = \sum_{i\in A}\sum_{m=1}^{M_i} b^r_{ikm}\cdot x_{i0m} \qquad \forall k\in R_r \tag{28}$$

$$r^r_{ek} = r^r_{e-1,k}+\sum_{i\in A}\sum_{m=1}^{M_i} b^r_{ikm}\cdot x_{iem}-\sum_{i\in A}\sum_{m=1}^{M_i} b^r_{ikm}\cdot y_{iem} \qquad \forall e\in\mathcal{E}, e\geq 1, k\in R_r \tag{29}$$

$$r^r_{ek}\leq B^r_k \qquad \forall e\in\mathcal{E}, k\in R_r \tag{30}$$

Nonrenewable resource constraints:

$$\sum_{i\in A}\sum_{m=1}^{M_i}\sum_{e\in\mathcal{E}} b^v_{ikm}\cdot x_{iem}\leq B^v_k \qquad \forall k\in R_v \tag{31}$$

Activity and mode constraints:

$$\sum_{m=1}^{M_i}\sum_{e\in\mathcal{E}} x_{iem}=1 \qquad \forall i\in A \tag{32}$$

$$\sum_{m=1}^{M_i}\sum_{e\in\mathcal{E}} y_{iem}=1 \qquad \forall i\in A \tag{33}$$

$$\sum_{e\in\mathcal{E}} x_{iem}=\sum_{f\in\mathcal{E}} y_{ifm} \qquad \forall i\in A, m\in 1...M_i \tag{34}$$

Timing constraints:

$$t_0=0 \tag{35}$$

$$t_f\geq t_e+p_{im}\cdot x_{iem}-p_{im}\cdot(1-y_{ifm}) \qquad \forall (e,f)\in\mathcal{E}^2, f\geq e, i\in A, m\in 1...M_i \tag{36}$$

$$\sum_{m=1}^{M_i}\sum_{e'=0}^{e-1} y_{ie'm}\leq e\cdot(1-\sum_{m=1}^{M_i} x_{iem}) \qquad \forall e\in\mathcal{E}, i\in A, e>0 \tag{37}$$

$$t_{e+1}\geq t_e \qquad \forall e\in\mathcal{E}, e<n \tag{38}$$

$$\sum_{m=1}^{M_i} ES_i\cdot x_{iem}\leq t_e\leq\sum_{m=1}^{M_i} LS_i\cdot x_{iem}+LS_{n+1}\cdot(1-\sum_{m=1}^{M_i} x_{iem}) \qquad \forall i\in A, e\in\mathcal{E} \tag{39}$$

$$ES_{n+1}\leq t_n\leq LS_{n+1} \tag{40}$$

$$\sum_{m=1}^{M_i}(ES_i + p_{im}) \cdot y_{iem} \leq t_e \leq \sum_{m=1}^{M_i}(LS_i + p_{im}) \cdot y_{iem} + LS_{n+1} \cdot (1 - \sum_{m=1}^{M_i} y_{iem}) \qquad \forall i \in A, e \in \mathcal{E} \quad (41)$$

Decision variable definitions:

$$t_e \geq 0 \qquad \forall e \in \mathcal{E} \quad (42)$$

$$r_{ek} \geq 0 \qquad \forall e \in \mathcal{E}, k \in R_r \quad (43)$$

$$x_{iem} \in \{0,1\}, \qquad y_{iem} \in \{0,1\} \qquad \forall i \in A \cup \{0, n+1\}, e \in \mathcal{E}, m = 1...M_i \quad (44)$$

The objective function (26) is again minimizing the makespan, this time by minimizing the date of the last event n.
Precedence constraints (27) indicate that for two activities (i, j) in a precedence relationship, j cannot start at at event e before which i has ended.

The renewable resource consumption is regulated by three constraints. As already stated r_{ek} represents the resource consumption of resource k right after event e. Therefore (28) is defining the resource consumption at event 0 by summing up the resource needs of all activities starting right after event 0. (29) does the same for all upcoming events by evaluating the consumptions of the last event adding all the activities starting at the event and subtracting the ones ending. Constraint (30) is then limiting the total amount of r_{ek} to the total capacity. In comparison the nonrenewable resource consumption is summing up the resource needs over all events and restricting it to the total capacity.

Constraints (32) and (33) indicate that each activity only starts and ends once at one event in one mode and constraints (34) make sure that this mode is the same at start and end events.

As the events need to be aligned with some points in the time horizon, additional timing constraints are needed. (35) clarify that the first event date is 0, whereas (36) indicate that if an activity starts at event e and ends at event f, the date of f needs to be at least bigger than the date of e plus the respective activity duration. Constraints (37) are very important and have been explicitly mentioned in Artigues et al. [2013], their main purpose is to make sure that an activity i can not start at an event e and end at an event $e' < e$. Constraints (38) put the events in ascending order. Constraints (39) to (41) restrict the event dates to the specific starting time windows for starting respectively ending events. (42) to (44) define the decision variables.

4.4.2 On-off Event-based formulation (OOE)

In the OOE model the approach is simpler, one set of binary variables z_{iem} indicate whether or not an activity i is in process at a certain event e in the chosen mode m. In other words the variable is showing the status of the activity, whether it is active (on) or inactive (off). Of course this reduces the number of events to the number of non-dummy activities. The upper bound of the events is denoted by v. The second binary variable set y_{iem} states if an activity i starts at event e in a mode m, marking the unique starting event. This additional variable is necessary to determine the nonrenewable resource consumption. $Cmax$ is a continuous variable representing the makespan of the project, while t_e again shows the date of event e.
Objective function:

$$\min \qquad C_{max} \tag{45}$$

Precedence constraints:

$$\sum_{m=1}^{M_i} z_{iem} + \sum_{n=1}^{M_j} \sum_{e'=0}^{e} z_{je'n} \leq 1 + (1 - \sum_{m=1}^{M_j} z_{iem}) \cdot e \qquad \forall (i,j) \in E, e \in \mathcal{E} \tag{46}$$

Renewable resource constraints:

$$\sum_{i=1}^{n} \sum_{m=1}^{M_i} b^r_{ikm} \cdot z_{iem} \leq B^r_k \qquad \forall e \in \mathcal{E}, k \in R_r \tag{47}$$

Nonrenewable resource constraints:

$$\sum_{i=1}^{n} \sum_{e \in \mathcal{E}} \sum_{m=1}^{M_i} b^v_{ikm} \cdot y_{iem} \leq B^v_k \qquad \forall k \in R_v \tag{48}$$

Activity and mode constraints:

$$\sum_{m=1}^{M_i}\sum_{e\in\mathcal{E}} z_{iem} \geq 1 \qquad \forall i \in A \tag{49}$$

$$\sum_{m=1}^{M_i}\sum_{e'=0}^{e-1} z_{ie'm} \leq \sum_{m=1}^{M_i} e\cdot(1-(z_{iem}-z_{i,e-1,m})) \qquad \forall e \in \mathcal{E}\setminus\{0\} \tag{50}$$

$$\sum_{m=1}^{M_i}\sum_{e'=e}^{v} z_{ie'm} \leq \sum_{m=1}^{M_i}(n-e)\cdot(1+(z_{iem}-z_{i,e-1,m})) \qquad \forall e \in \mathcal{E}\setminus\{0\} \tag{51}$$

$$\sum_{e\in\mathcal{E}}\sum_{m=1}^{M_i} y_{iem} = 1 \qquad \forall i \in A \tag{52}$$

$$y_{iem} \geq z_{iem} - z_{i,e-1,m} \qquad \forall i \in A, e \in \mathcal{E}, m = 1...M_i \tag{53}$$

Timing constraints:

$$t_0 = 0 \tag{54}$$

$$C_{max} \geq t_e + \sum_{m=1}^{M_i}(z_{iem}-z_{i,e-1,m})\cdot p_{im} \qquad \forall e \in \mathcal{E}, i \in A \tag{55}$$

$$ES_{n+1} \leq C_{max} \leq LS_{n+1} \qquad \forall e \in \mathcal{E}, i \in A \tag{56}$$

$$t_{e+1} \geq t_e \qquad \forall e \neq n-1 \in \mathcal{E} \tag{57}$$

$$t_f \geq t_e + ((z_{iem}-z_{i,e-1,m})-(z_{ifm}-z_{i,f-1,m})-1)\cdot p_{im}$$
$$\forall (e,f,i) \in \mathcal{E}^2 xA, f > e, m = 1...M_i \tag{58}$$

$$\sum_{m=1}^{M_i} ES_i \cdot z_{iem} \leq t_e \leq LS_i \cdot \sum_{m=1}^{M_i}(z_{iem}-z_{i,e-1,m}) + LS_n\cdot(1-\sum_{m=1}^{M_i}(z_{iem}-z_{i,e-1,m}))$$
$$\forall e \in \mathcal{E}, i \in A \tag{59}$$

Decision variable definition:

$$t_e \geq 0 \qquad \forall e \in \mathcal{E} \tag{60}$$

$$z_{iem} \in \{0, 1\} \qquad \forall i \in A \cup \{0, n+1\}, e = 0...v, m = 1...M_i \tag{61}$$

$$y_{iem} \in \{0, 1\} \qquad \forall i \in A \cup \{0, n+1\}, e = 0...v, m = 1...M_i \tag{62}$$

In order to also account for $e = 0$, in the preprocessing phase also an process variable $z_{i,-1,m} = 0$ for $e = -1$ is considered. The model minimizes the previously defined variable Cmax (45), which is simply representing the makespan.

Precedence constraints (46) state that if activity i precedes activity j, j cannot be active ("on") at any event before e, if i is still active at e.
The renewable resource constraints (47) are putting the resource consumption of each resource at each event in perspective to the available capacity. For the non-renewable resources, only the event where i is starting is relevant, represented by the variable y_{iem} in the relevant mode.

Constraints (49) make sure that each activity is at least active at one event. Constraints (50) and (51) guarantee the non-preemption of an activity i. (50) is stating that if i is starting at event e ($z_{iem} = 1; z_{ie-1m} = 0$) then it can not start at any event before e. (51) is referring to the opposite, if i ends at e ($z_{iem} = 0; z_{ie-1m} = 1$) then it cannot be active at any event after e. Constraints (52) request that each activity starts at exactly one event, whereas (53) is defining the starting point, by linking variables z_{iem} and y_{iem}.

Again the first event date is defined by 0 (54) and Cmax is set to be bigger or equal to the event date of each activity plus its duration (55). (56) limits Cmax further to the starting time window of the last activity and (57) order the event dates. (58) is defining similarly to the SEE model, that the start and end date of an activity i need to be at least separated by the activity duration. (59) on the other hand put the event date in perspective of starting time windows and constraints (60) - (62) define the decision variables.

4.5 Model comparison and analysis

When comparing the models, the first assumption would be that the Time-Indexed formulations have a clear advantage in terms of the amount of decision variables needed. They only use one set of binary decision variables whereas all other models, especially in the multi-mode case need at least two sets of binaries and some sets of continuous decision variables. However, this special feature of the DT and Disaggregated Discrete-Time (DDT) models becomes problematic with increasing time horizons. The number of the time-indexed decision variables increase of course proportionally to the time horizon, a feature the other models do not share. The Flow-based continuous time (FCT) and the event-based formulations do not have any dependency on the time frame within their decision variables. This is a clear advantage over the less-extensive time-indexed formulations when the time horizon is large.

The flow-based formulation involves fewer binary decision variables than the event-based models, but on the other hand has to consider the big-M constant, calculated individually per activity combination in the preprocessing and the need for additional linearization constraints in the model implementation.

In comparison, the event-based models do not have to deal with big-M constants, the OOE formulation involving fewer binary decision variables than the SEE model. On the downside the nonrenewable resource constraint now needs an additional binary variable to clarify the starting event. From this short characterization it is now clear, that the extend of the time horizon has a crucial influence on the performance of the models, as well as the number of activities and the precedence relations.

These properties have been tested for the Single-Mode case by Koné et al. [2011] on various instance sets. To account for more settings they also introduced new sets with a very high processing time range to align the examples with reality. Not surprisingly the time-indexed models performed very good on the instance sets with low processing time ranges and low scheduling horizon, followed by FCT and OOE approaches. Within the new sets, these two formulation outperform the remaining ones, although they were not able to solve all included instances.

It is the scope of this paper to evaluate if these results can be upheld for the multi-mode case or if the introduction of additional complexity and variables lead to a shift in the outcome. Especially the performance of the OOE approach could lose due to the introduction of additional binary decision variables. Apart from that the time restriction on the calculation run will be of crucial importance when it comes to finding an optimal solution.

5 Computational Analysis

In order to evaluate the different models presented in the last section, this chapter is dedicated to the actual implementation. For a better understanding of the process, the first part will be discussing the analytical framework of the analysis, including the instances' characteristics as well as the computational settings and some remarks on the instance sets. The second part will present the results and discuss their meaning also in dependence of parameter variations.

5.1 Analytical Framework

5.1.1 Parameter settings

The testing was performed on an HP Notebook PC with a 2.4 GHz processor and 4 GB RAM, running on Windows 7. The models were coded in Python 2.7 using the PuLP module with a CPLEX solver interface. Restrictions of time or node size as well as other parameters were modeled in the CPLEX environment.

Due to the high complexity the modeling took advantage of the node file writing possibility offered by CPLEX. Especially for the larger instances this was crucial due to exceedingly high memory requirements. This feature enables the program to reduce the amount of memory space needed while exploring the search space by saving parts of the nodes in the tree in files on the hard drive. Apart from that also an upper cutoff level was set to the value of an upper bound on the project makespan or to the maximal project horizon to restrict the search space.

5.1.2 Discussion of set characteristics

As already mentioned in an earlier chapter, the instance sets used in this thesis were taken from the PSPLIB, due to the fact that those sets are widely used and well-reputed and are therefore favorable for a first computational study of the new models. Kolisch and Sprecher [1997] generated those instances by combining and varying three different sets of parameters:

- Fixed parameters
- Base parameters and
- Variable parameters.

Fixed parameters are - as indicated by the name - fixed in advance and are the **same for all benchmark sets**. They are not varied at all during the set generations and include basic information like probabilities and deviation tolerances.

Base parameters are **varied once per benchmark set** and include for example the number of activities, resources, predecessors and successors as well as the activity duration range and the so called Network Complexity (NC). NC is a parameter that indicates the average number of non-redundant arcs per activity node. A general assumption is that the higher this value is, the easier the problem is to solve. As this value is set per benchmark set, it is not possible or in scope of this thesis to prove this assumption by parameter variation.

Variable parameters on the other hand are

> "...systematically **varied within** each benchmark set." [Kolisch and Sprecher, 1997, p. 209]

These variable parameters are the Resource Factor (RF) and the Resource Strength (RS). They have shown to affect computational results significantly and should therefore be explained in more detail.

The RF of a certain resource represents the average requirements of this specific resource within the problem, whereas the RS shows the strength of the related resource constraints. Or as put by [Kolisch and Drexl, 1997a, p. 995]

> ”The resource factor reflects the density of the coefficient matrix... . The resource strength measures the degree of resource-constrainedness in the interval [0,1].”

To get a bit more into detail for these two important factors, we will take a closer look at the set up of them provided by Kolisch and Sprecher [1997]. First, the RF for each resource category τ where τ represents renewable and nonrenewable resource categories is calculated as follows:

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

The formula basically reflects the average amount of a certain resource capacity τ used across all non-dummy activities and all modes available. The variable Q_{jm} is the amount of combinations (j, m, r), for which the respective $b_{jmr}^r > 0$. The RF has to be determined by using a resource factor deviation tolerance ϵ_{RF} as a controlling instrument, implying

$$RF_\tau \in [(1 - \epsilon_{RF}) \cdot RF_\tau; (1 + \epsilon_{RF}) \cdot RF_\tau]$$

The value of ϵ in the calculations is as mentioned above a fixed variable, valid for all benchmark instances.

The calculation of the RS based on the determination of the resource availability K_r^τ [Kolisch and Sprecher, 1997, p. 209] is presented

> ”...as a convex combination of a minimum and maximum level K_r^{min} and K_r^{max}, $r \in \tau$...”

where RS functions as scaling parameter.

$$K_r^\tau = K_r^{min} + round(RS_\tau \cdot (K_r^{max} - K_r^{min}))$$

This indicates that with an RS of 0, the problem only deals with the smallest resource availability possible. The minimum and maximum availability levels for nonrenewables are determined by aggregating the respective consumptions.

$$K_r^{min} = \sum_{j=2}^{J-1} min_{m=1}^{M_J}\{k_{jmr}^v\}$$

$$K_r^{max} = \sum_{j=2}^{J-1} max_{m=1}^{M_J}\{k_{jmr}^v\}$$

For the renewables the determination is as follows:

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

K_r^{max} for renewables on the other hand is [Kolisch and Sprecher, 1997, p. 209]

> ”...determined by the peak per-period usage of resource r in the resource dependent earliest start schedule.”

It is generally observed that the RF and the hardness of a problem are positively correlated, meaning that the higher the RF or the denser the coefficient matrix, the harder the problem is to solve. This is easier to understand when one considers the fact that RF = 1 means that every activity in every mode would require at least one unit of each resource, whereas a lower RF also allows for resource requirements of 0, leading to a less intense scheme. The opposite indication is valid for the RS, the closer the factor is to 0, the harder the problem is, indicating more active resource constraints that need to be considered in the optimization process.

To generate the sets with the above-mentioned parameters, a full factorial test design was executed including 10 replications. For the sets under review in this thesis, **J10** and **J20**, this was resulting in a total of 640 instances each, though not all of them are feasible. The set names J10 and J20 indicate already the number of activities in each set. Whereas J10 comprises 536 feasible instances with 10 non-dummy activities, J20 includes 554 feasible instances with 20 non-dummy activities. Both sets have two renewable and two nonrenewable resources available as well as three different modes for choice. The activity durations p_{im} are settled in the range from 1 to 10. Across all models the time restriction for the optimization procedure was set to 300 seconds for the J10 instances and to 1800 seconds for the J20 instances. CPLEX ultimately has a certain tolerance when it comes to the time limitation. It is for example possible that an instance of the J20 set has a solution time of 1805 seconds, increasing the solution time to a value above the actual time limit.

Before starting any analyses, below list shows an overview of the parameters used in the tables below and their respective abbreviations in the table headings:

- **Set:** Name of the set under review
- **N:** Number of instances in a certain set or subset
- **AvT:** Average solution time needed (in seconds)
- **AvN:** Average node amount of the solution
- **AvGap:** Average gap in percent between the makespan and the lower bound
  Remark: The lower bound was determined by calculating the makespan without considering the resource requirements (= Earliest Starting Time)
- **Model:** Name of the model under review
- **Feasible #/%:** Number/Percentage of feasible (integer) solutions found
- **Best:** Number of solutions found that match the reference best solution from literature
- **Opt:** Number of instances which could be solved to proven optimality within the given time limit

It is important to note here that the parameters like solution time and node amount span their average only across the actual feasible solutions found and not across the whole parameter set. This fact should be made obvious to the reader as it might not be clear from below tables. Also results have been rounded for a more readable presentation. This rounding has been performed after the actual

calculation and has therefore no influence on the results themselves.
The above mentioned factors RF and RS are set for both resource types and named accordingly **RFR** and **RSR** for the renewable and **RFN** and **RSN** for the nonrenewable resource set. The values of these parameters are different for both instance sets and can be found in Table 1 and Table 2 accompanied by the value of NC. The influence of these parameters will be tested accordingly in the analysis conducted below.

RFR	0.5		1.0	
RSR	0.2	0.5	0.7	1.0
RFN	0.5		1.0	
RSN	0.2	0.5	0.7	1.0
NC	1.5			

Table 1: Variable parameter setting and NC for J10 instance set

RFR	0.5		1.0	
RSR	0.25	0.5	0.75	1.0
RFN	0.5		1.0	
RSN	0.25	0.5	0.75	1.0
NC	1.8			

Table 2: Variable parameter setting and NC for J20 instance set

According to the above explanation of the parameters, their setting will thoroughly influence the outcome. To illustrate this in a very general context, two different sub-instance sets of J10 and J20 are introduced below to account for two possible settings of RF and RS. Both subsets - denoted 10 and 37 - comprise 10 instances with the same NC of 1.5, differing only in their setting of the resource related parameters. While the examples in J1010 (J2010 respectively) have all parameters set to 0.5, the formulation of J1037 (and J2037) is more extreme, having an RF of 1.0 and an RS of 0.2 (0.25 respectively for J20) for all resource types.

As mentioned above a high RF value and a low RS value would indicate a harder instance to solve. The first assumption is therefore that the instances numbered 37 are more difficult to solve than those of subset 10, which proves to be true when considering below results. The second assumption here is that these differences are going to be more obvious for the J20 set due to a higher number of jobs.

Table 3 - 6 show the average outcome of the 10 instances per set in terms of average solution time in seconds, average node amount and average solution gap as well as the percentage of best bound solutions and the number of optimal solutions found per model approach.

Model	Best	Opt	AvT	AvN	AvGap
DT	100%	100%	0.66	160	
DDT	100%	100%	1.63	7	
FCT	100%	100%	0.59	20	20.3%
SEE	100%	80%	95.29	1,499	
OOE	100%	100%	50.79	2,176	

Table 3: Results overview J1010

Model	Best	Opt	AvT	AvN	AvGap
DT	100%	100%	49.64	17,716	155.63%
DDT	100%	100%	27.06	908	155.63%
FCT	100%	100%	8.01	1,939	155.63%
SEE	80%	20%	264.72	26,980	157,05%
OOE	90%	30%	246.79	5,804	156,48%

Table 4: Results overview J1037

Model	Best	Opt	AvT	AvN	AvGap
DT	100%	100%	2.39	354	2.7%
DDT	100%	100%	20.06	6	2.7%
FCT	100%	100%	16.20	637	2.7%
SEE	100%	80%	674.60	20,198	2.7%
OOE	50%	50%	1,060.64	46,778	6.3%

Table 5: Results overview J2010

Model	Best	Opt	AvT	AvN	AvGap
DT	20%	10%	1,723.08	108,671	103.0%
DDT	0%	0%	1,800.86	10,038	116.8%
FCT	10%	0%	1,801.11	26,069	119.7%
SEE	0%	0%	1,803.06	33,702	121.4%
OOE	0%	0%	1,801.27	95,989	121.8%

Table 6: Results overview J2037

As can be clearly seen the optimization of instance set J1010 requires considerably less time and nodes to find a feasible solution. Apart from that CPLEX could find the best reference solution within the given time frame for almost all instances. Also when considering the number of optimal solutions found, most problems are solve to optimality within the given time horizon. The average solution gap between the makespan found and the actual lower bound is vastly different, and exceedingly higher in the J1037 approaches.

The same observation made for J10 can also be made for the J20 instances. With nearly none of the problems from the set J2037 the reference best solution could be found. The average node size - although already very high for the event based models in the easy subset 10 - increases dramatically across all models. Nevertheless the difference seems to be most obvious in the time-indexed models, especially in the DT model. Same is true for the average solution gap, although here the differences across models are not as obvious as with the node size.

This is of course only a rough overview including averages and only specifically chosen instance sets, but it is clearly showing the huge influence of the variable parameters, not just on one but on all models. It therefore makes sense to also take these parameters into consideration when conducting a more detailed analysis in the second part of this chapter.

5.1.3 Comments on PSPLIB

Although the benchmark instances introduced in Kolisch and Sprecher [1997] are widely used, also several downsides of the settings have been recognized. In order to provide the reader with all information needed, also those issues should be mentioned here to contribute to future analyses.

First of all the restriction of activity durations to a value between 1 and 10 is very limiting when this feature should be evaluated in terms of its influence on optimality. Due to this restriction, Koné et al. [2011] introduced several new instance sets with a wider variety in the duration range. This in turn intends to put additional pressure on the time-indexed formulations and adds another dimension to the analytical framework.

Another downside is that out of the provided instance sets not all have an actual feasible solution or provide inefficient mode alternatives in the first place. Apart from that it is - due to the nature of the instances - not really possible to achieve any major improvements by an algorithm as most solution procedures can indeed solve these problems to optimality.

These shortcomings were mentioned explicitly by Van Peteghem and Vanhoucke [2014] and triggered the introduction of new instance sets, therefore broadening the scope of future testing. Despite the problematic aspects the PSPLIB instance sets are still well-reputed and ideal for an initial testing of the new models.

5.2 Results and analysis

5.2.1 Remarks on the analysis

The main objective of below numerical analysis is to show whether or not one or all of the introduced models show the potential for solving the MRCPSP compared to the current best solution. Apart from that general comparison it is also analysed which of the models are performing best under which circumstances. Last but not least comparisons should be drawn to the SRCPSP results presented by Koné et al. [2011]. It will be interesting to see whether their results are also true for the multi-mode case or if the influence of the mode number changes the outcome.

In order to answer the above questions, a series of tests with the models was performed and analysed according to a unified scheme. For each problem set two different upper bounds where used. The first approach uses a tighter upper bound which equals the best known upper bound plus 4%, in the latter this subversion is denoted by the extension "0". The second version identified by the extension "1" used the above mentioned maximum sum of all activity durations as the upper bound, granting a higher search space and probably a higher complexity.

Before starting the analysis, Table 7 show the reference best solutions found in literature for the two sets under review in terms of average solution time and - if available - the average node amount.

Set	N	Author	AvT (in sec)	AvN
J10	536	Sprecher and Drexl [1998]	0.14	
J20	554	Sprecher and Drexl [1998]	198.74	
		Zhu et al. [2006]	32.06	238

Table 7: Reference Best Solution

5.2.2 Analysis J10

Table 8 shows the results of the calculation for the instance set J10 and the shorter time horizon of the best bound + 4%.

Model	Feasible		Best	Opt	AvGap	AvT	AvN
	#	%					
DT	536	100%	536	536	32.2%	0.69	196
DDT	536	100%	536	536	32.2%	0.37	20
FCT	536	100%	536	536	32.2%	2.00	283
SEE	534	99.6%	522	404	32.1%	109.8	4,767
OOE	536	100%	533	461	32.3%	84.76	3,954

Table 8: Result overview J10_0

The first obvious result is that almost all models can find the best solutions provided by literature. Also most of them found the optimal solution, using of course less than 300 seconds. The event-based models show as expected a slight deviation. Of those instances where an integer solution was found, most of the models can find an equal number of optimal solutions and show a quite similar solution gap to the lower bound.

The greatest differences can be observed in the solution time and the node amount. While the time-indexed formulations and the flow-based model show average solution times of under 5 seconds, the event-based models take significantly more time to reach a similar result and an even higher amount of nodes.

When only comparing the time indexed formulations, the DDT model performs better than the DT which coincides with the basic findings of Koné et al. [2011]. The same is valid for the (slightly) better performance of the OOE formulation compared to the SEE model when considering solution time and node size within the event-based models.

The findings valid for the above table are in general also true for the analysis provided in Table 9 with a larger time horizon.

Model	**Feasible**		**Best**	**Opt**	**AvGap**	**AvT**	**AvN**
	#	%					
DT	536	100%	536	536	32.2%	2.75	820
DDT	536	100%	536	536	32.2%	3.52	50
FCT	536	100%	536	536	32.2%	3.45	437
SEE	536	100%	524	386	32.5%	119.24	4,764
OOE	536	100%	531	459	32.3%	86.42	4,074

Table 9: Result overview J10_1

Here the time-indexed formulations perform better in terms of timing and node size, followed by the flow-based model. The DT model is slightly faster than the DDT but still using more nodes to find the same solution and also more than the FCT model which only needs one second more on average for the solution.

The event-based models are still at the end in terms of performance and finding the best solution, with the OOE model still slightly better. What is more obvious is that it gets more difficult for the event-based versions to find the best solution in below 300 seconds, leading to an increasing difference between the values of the variables Best and Opt per model.

What is really interesting to investigate is the change in performance of the different models when only the time horizon is extended, all other parameters being equal. Although in fact the event-based models are still found at the end of the performance range, the extension of the time horizon has not significantly changed the solution time or the amount of nodes needed.

As expected changes in the time horizon have no or just minimum influence on the performance of these models, which seems to be a clear advantage.

Comparing this result to the changes of the time-indexed models, it looks quite different. Although they are still outperforming in both situations, the changes are obvious. The solution time needed almost tripled for both and the nodes needed in the DT model increased about four times. The FCT also showed increases in timing and node amount, but seems to be less sensitive towards time changes.

After that general analysis, the next tables show a more detailed view on the different models when it comes to varying settings of RS and RF. Those results are divided into three tables per model approach for a better overview. The detailed analysis is only presented for the subversion of the calculation using the larger time horizon in order to capture the changes better. Table 10 is showing the detailed results for the time-indexed models.

Parameter		N	DT					DDT				
			Best	Opt	AvGap	AvT	AvN	Best	Opt	AvGap	AvT	AvN
RFR	0.5	259	259	259	26.4%	0.54	154	259	259	26.4%	1.43	2
	1.0	277	277	277	37.8%	4.81	1,442	277	277	37.8%	5.48	94
RSR	0.2	119	119	119	76.0%	10.59	3,350	119	119	76.0%	10.52	213
	0.5	139	139	139	25.4%	0.78	216	139	139	25.4%	2.07	3
	0.7	138	138	138	19.4%	0.47	53	138	138	19.4%	1.38	5
	1.0	140	140	140	14.5%	0.29	24	140	140	14.5%	1.13	1
RFN	0.5	232	232	232	18.7%	2.00	487	232	232	18.7%	2.96	35
	1.0	304	304	304	42.6%	3.32	1,074	304	304	42.6%	3.95	61
RSN	0.2	78	78	78	87.7%	7.26	2,652	78	78	87.7%	5.25	120
	0.5	151	151	151	33.2%	2.68	883	151	151	33.2%	4.50	82
	0.7	156	156	156	20.1%	1.46	436	156	156	20.1%	3.18	20
	1.0	151	151	151	15.3%	1.82	206	151	151	15.3%	2.01	12

Table 10: Detailed results J10_1_Time indexed

For the detailed analysis, one is firstly referred to the two parameter sections regarding the renewable resources. As expected the higher the RF the harder the problem is to solve. This comes clear when comparing the average timing and node amount. It is also worth noting that the DDT model performs slightly worse in terms of solution time in the harder cases than the DT model. The influence of the parameters becomes even clearer when looking at the RS. While the smaller coefficient leads to a comparably high use of time and nodes, the influence diminishes with rising coefficient.

The largest jump in terms of solution time and nodes can be found when moving from the smallest to the second-smallest coefficient level the following stages are not as intense. It is quite interesting to see that although the DDT uses less nodes than the DT model it takes a bigger amount of solution time. This can be related to the fact that on the one hand, the DDT model uses more constraints due to the different precedence constraint. This in turn takes up more time to solve. On the other hand the precedence constraint of the DDT model is stronger than the DT constraint - or put differently - it is implying the statement of the DT constraint. This in turn reduces the search space for the DDT model and thus decreases the node size needed to explore and find a solution.

When considering the nonrenewable section the above assumptions also prove to be correct, although their influence is not as intense as it is in the renewable part. Nevertheless the changes in the results are still significant enough to prove the assumptions to be true.

When comparing the makespans found to the results from literature both models could compete across the parameter sets, providing all instances found in literature and all of them within the given time limit.

Table 11 show the analysis for the flow-based approach.

Parameter		N	FCT				
			Best	**Opt**	**AvGap**	**AvT**	**AvN**
RFR	0.5	259	259	259	26.4%	0.56	42
	1.0	277	277	277	37.8%	6.15	806
RSR	0.2	119	119	119	76.0%	13.17	1,624
	0.5	139	139	139	25.4%	1.35	257
	0.7	138	138	138	19.4%	0.46	35
	1.0	140	140	140	14.5%	0.22	2
RFN	0.5	232	232	232	18.7%	4.05	489
	1.0	304	304	304	42.6%	2.99	397
RSN	0.2	78	78	78	87.7%	1.47	287
	0.5	151	151	151	33.2%	3.74	415
	0.7	156	156	156	20.1%	3.78	510
	1.0	151	151	151	15.3%	3.84	460

Table 11: Detailed results J10_1_Flow-based

The results back up the assumption made for the renewable resources. However within the nonrenewables, the findings are quite surprising. Despite the results from Table 10, here the average timing and node size for increasing parameters of RFN decrease. This difference is also true for the RSN, less nodes and time is used in the supposedly harder levels. The differences are not that big or intense, still leading to the conclusion that the parameter settings for the nonrenewables are not as influential within the FCT model. These results are unexpected and will be investigated further in the J20 analysis if they are also observed there. However, as with the time-indexed models all instances could find a reference solution within the given time frame.

Last but not least, Table 12 provides the results for the event-based models.

Parameter		N	SEE					OOE				
			Best	**Opt**	**AvGap**	**AvT**	**AvN**	**Best**	**Opt**	**AvGap**	**AvT**	**AvN**
RFR	0.5	259	259	228	26.4%	77.79	1,526	259	253	26.4%	44.56	2,227
	1.0	277	265	158	38.2%	157.99	7,791	272	206	37.9%	125.57	5,802
RSR	0.2	119	109	35	76.9%	249.89	12,859	117	77	76.1%	169.35	6,236
	0.5	139	138	94	25.5%	142.96	4,783	136	115	25.6%	104.16	5,851
	0.7	138	137	120	19.4%	65.51	1,956	138	128	19.4%	55.96	2,908
	1.0	140	140	137	14.5%	33.63	633	140	139	14.5%	28.37	1,623
RFN	0.5	232	228	167	18.9%	103.21	4,025	232	204	18.7%	65.56	3,049
	1.0	304	296	219	42.9%	131.47	5,328	299	255	42.7%	102.35	4,857
RSN	0.2	78	76	61	87.8%	141.97	5,993	76	64	87.8%	131.88	4,061
	0.5	151	148	96	33.4%	151.02	5,917	149	120	33.3%	118.43	6,271
	0.7	156	151	112	20.3%	107.33	4,928	156	135	20.1%	74.30	3,976
	1.0	151	149	117	15.5%	88.00	2,807	150	140	15.4%	43.46	1,986

Table 12: Detailed results J10_1_Event based

It is quite clear, that the results are mostly in line with the previous time-indexed formulation results. The only difference is that the values of Best and Opt differ across the parameters, indicating that not all of the best/optimal solutions found could be detected within the 300 seconds limit.

What is now left to analyse for this set is if the influence of the parameters differ in its extent among the different model approaches. As already detected in the general analysis, the OOE model performs faster than the SEE approach. The differences in the node amount used is not distinct and differs across the parameter levels. As expected more of the reference solutions were found for the easier parameter instances, with the SEE model showing a better performance. Interestingly of those that actually found the best solution, the OOE approach could solve more instances to optimality. The differences are quite big when considering the harder instances in the renewable section, they will be further analysed within the J20 instance set.

When taking a look at the changes in the different RFR levels it seems as if the greatest influence on the solution timing can be found within the DT and FCT models, showing an 8 to 10 times higher time value. However when it comes to the node size, the DDT model has the worse effects to bear. The event-based models seem in comparison to be the least affected. When considering the RSR section, the most obvious relative differences in timing between the lowest and highest parameter level can again be found within the DT and FCT model. Although the event based models need significantly more solution time, the changes within the parameter settings for the renewables are not that intense. The same is true for the RSR, while the time indexed and flow-based models show up to an 800 times larger node amount for the harder cases, the event-based models show a maximum of 20 times higher nodes sizes.

Apart from the above-mentioned specialty of the FCT model, the other approaches reveal now a higher variability amongst the nonrenewable settings in the RFN parameter set. When observing the development within the RSN adaptations, the event-based models again show a smaller vulnerability to changes in the settings than the time-indexed formulations.

To sum up, it seems as if changes in the setting of renewable resource have an influence on all of the models in the same way, but show the greatest changes when it comes to the time-indexed and the flow-based formulations. The following section will show if this higher sensitivity can also be found with a higher number of activities.

5.2.3 Analysis J20

The analysis of J20 is built up in the exact same way as the one of J10, starting with a general overview of the results for the two different time horizons, moving on to the more detailed results of parameter influences.

Table 13 shows the results when using a smaller time horizon, proving again the dominance of time-indexed formulations over event-based models when it comes to finding feasible and best solutions.

Model	**Feasible**		**Best**	**Opt**	**AvGap**	**AvT**	**AvN**
	#	%					
DT	543	98.0%	540	534	15.5%	66.03	15,789
DDT	552	99.7%	548	544	16.8%	45.81	1,995
FCT	499	90.1%	481	468	12.8%	170.9	6,204
SEE	339	61.2%	277	219	5.3%	720.1	12,178
OOE	347	62.6%	284	219	6.1%	781.0	25,714

Table 13: Result overview J20_0

Interestingly – when observing the results – the before very clear dominance of the time-indexed models over the other approaches diminishes when looking at the best/optimal solutions found. Although they still find an integer solution in the majority of the cases, still dominating the event-based solutions, the share of the optimal solutions found is decreasing. When in turn looking at the event-based models, they could not always find a feasible solution, in the majority of the cases where they did, the optimum was found, leading also to a lower average solution gap. None of the models could find all feasible solutions, although the DDT model got quite close. The average solution time was again highest within the event-based formulations, followed – with a large gap – by the FCT. The DT model seems to lose partly its efficiency, needing both higher solution time and an even higher node amount than the DDT model and also finds less feasible solutions. Also compared to the FCT approach, the node amount used by the DT is quite high.

Table 14 shows the results for the larger time horizon.

Model	**Feasible**		**Best**	**Opt**	**AvGap**	**AvT**	**AvN**
	#	%					
DT	554	100%	533	524	17.3%	147.1	13,872
DDT	554	100%	530	527	17.6%	153.8	1,073
FCT	553	99.8%	471	457	19.4%	412.6	7,343
SEE	542	97.8%	297	223	23.0%	1,162	27,661
OOE	553	99.8%	295	219	22.2%	1,200	55,753

Table 14: Result overview J20_1

It is first to notice that in the setting with a larger time horizon, more feasible solutions could be found, although the share of the best solutions found increases only for the event-based models. The same is true for the optimal makespans found. Only the event-based approaches seem to profit from the larger time horizons, finding more optimal solutions within the given time frame of 1800 seconds. Solution time of course increases, as well as the node amounts to a certain extent. Unexpectedly the node amount value did not rise within the time-indexed models but decreased to a small extent. The node amount used in the event-based formulations more than doubled.

After that broad overview, the more detailed analysis below, will show if the propositions held above are also true for a larger number of activities.

A quick glance at Table 15 shows the similarity of the results found within set J10 when investigating the time-indexed models.

Parameter		N	DT					DDT				
			Best	Opt	AvGap	AvT	AvN	Best	Opt	AvGap	AvT	AvN
RFN	0.5	244	240	236	10.0%	94.17	10,105	239	238	10.1%	110.52	827
	1.0	310	293	288	23.1%	188.78	16,838	291	289	23.6%	187.86	1,267
RSN	0.25	75	66	63	57.0%	398.59	27,269	63	63	59.0%	352.96	1,826
	0.5	159	150	145	16.6%	188.94	19,691	149	149	16.8%	200.19	1,543
	0.75	160	158	157	8.8%	80.43	10,413	158	156	8.8%	101.66	704
	1.0	160	159	159	8.0%	54.31	5,271	160	159	7.9%	66.48	624
RFR	0.5	271	271	271	13.5%	28.87	3,569	271	271	13.5%	29.54	46
	1.0	283	262	253	21.0%	260.32	23,739	259	256	21.6%	272.79	2,057
RSR	0.25	140	120	112	40.5%	518.24	48,908	117	114	41.7%	502.42	3,805
	0.5	141	140	139	12.8%	54.20	4,453	140	140	12.8%	73.95	428
	0.7	143	143	143	10.2%	6.24	1,133	143	143	10.2%	17.47	8
	1.0	130	130	130	5.1%	3.13	373	130	130	5.1%	14.93	4

Table 15: Detailed results J20_1_Time indexed

Again the higher the RF value or the lower the RS value, the higher is the solution time and node amount needed to find a solution. The most remarkable result is that the node amount used by the DT model is significantly higher, although its solution time value can be found among the ranges of the DDT results. The DT model finds more solutions referenced by literature, though the difference is not that significant. But also in both approaches more optimal solutions could be found with the assumed easier parameter. This coincides with the results found in the analysis of J10.

The results in Table 16 concerning the flow-based approach also confirm the main settings found in the J10 cases.

Parameter		**N**	**FCT**				
			Best	**Opt**	**AvGap**	**AvT**	**AvN**
RFN	0.5	244	212	206	11.8%	370.04	7,914
	1.0	310	259	251	25.3%	446.22	6,893
RSN	0.25	75	61	59	58.7%	488.38	7,157
	0.5	159	134	130	19.5%	434.53	6,601
	0.75	160	136	131	10.8%	392.66	6,748
	1.0	160	140	137	9.5%	375.72	8,761
RFR	0.5	271	261	257	13.4%	179.44	2,554
	1.0	283	210	200	25.0%	635.06	11,912
RSR	0.25	140	65	55	48.6%	1,252.03	18,852
	0.5	141	133	129	13.1%	355.17	9,438
	0.75	143	143	143	10.2%	22.72	726
	1.0	130	130	130	5.1%	6.24	45

Table 16: Detailed results J20_1_Flow-based

The parameters in the nonrenewables section again show a slightly different behavior than expected and only small differences in solution node amount between the extremes. This confirms the finding within the analysis for the J10 set for the node amount, but not for the timing. The solution values behave more as expected being higher for the harder parameter settings, as well as the values of best and optimal solutions found being lower.

However those findings are not significant enough to completely ignore the differences when comparing to the other models. One reason for this insensitivity to the set up of the nonrenewable resources might be that the nonrenewable resource constraint for this approach is far simpler than the ones found in the time-indexed and event-based approaches. It only considers activities and modes not taking into account timing and event indices like the other models. Changes in the nonrenewable resource matrix might therefore be less crucial for the result. This is an interesting feature of this model that might be useful for specific problem settings and should be evaluated further with a larger problem set.

For the renewable cases on the other hand the results are in coherence with the results found in other models.

The outcome of the event based models in Table 17 shows again the expected development of values across parameter settings.

Parameter		N	SEE					OOE				
			Best	Opt	AvGap	AvT	AvN	Best	Opt	AvGap	AvT	AvN
RFN	0.5	244	167	135	14.3%	933.81	21,178	165	130	13.2%	986.43	44,342
	1.0	310	130	88	29.9%	1,342.94	32,813	130	89	29.4%	1,368.15	64,763
RSN	0.25	75	9	0	67.2%	1,801.44	42,691	10	0	68.4%	1,801.39	70,903
	0.5	159	73	45	24.0%	1,378.06	32,271	62	36	22.4%	1,513.19	74,845
	0.75	160	107	87	12.7%	934.62	21,175	107	90	11.9%	945.45	45,331
	1.0	160	108	91	12.3%	883.05	22,751	116	93	10.8%	862.42	40,220
RFR	0.5	271	184	129	16.1%	1,034.32	23,722	180	125	16.6%	1,079.64	49,609
	1.0	283	113	94	29.6%	1,284.61	31,458	115	94	27.7%	1,315.12	61,657
RSR	0.25	140	20	2	56.8%	1,780.84	37,517	35	6	50.4%	1,738.23	70,403
	0.5	141	70	48	18.5%	1,295.73	36,434	64	46	17.8%	1,320.56	67,522
	0.75	143	101	80	12.4%	955.44	23,919	95	75	13.3%	998.36	48,840
	1.0	130	106	93	6.2%	633.92	12,559	101	92	6.8%	714.37	34,927

Table 17: Detailed results J20_1_Event based

It can be observed that in the nonrenewable cases the timing of both models is almost the same, whereas the node amount needed is almost twice as high in the OOE formulation. In the renewables section these findings are confirmed as well. What is obvious is that the differences in the solution time for the two models are diminishing with the SEE model performing slightly better. On the other hand the node amounts needed by the OOE approach are now almost double the size on average than the ones used by the SEE. In general it seems as if the OOE model is more favourable for the J10 set in terms of solution time due to its fewer resource constraints. But as the activity and with it the event size increases this advantage diminishes. This might in part be due to the fact that the constraints in the activity/mode section of the OOE are – in comparison – not only more numerous, but also bound to both – the activity amount and the events. This could be sufficient enough to make up for the smaller amount of resource constraints. It would be interesting to further observe this development with more activities to investigate whether this discrepancy is continuing.

When considering the amount of best and optimal solutions found, the results are ambiguous. No real dominance can be found as both models find a comparable amount for each parameter set. However the assumption that more best and optimal solutions can be found for easier parameter instances is again proven.

When comparing the extent of the influence the parameter changes have, the differences in the RFN are almost non-existent, while the influence of the RSN is higher when taking a look at the time-indexed formulations. The same statement is true in the case of renewable resources, having for example about 170 to 200 times larger solution times for the lower RSR instances than for the highest ones. It seems as if both the time-indexed and the flow based models show a higher vulnerability to changes in the parameter settings, both in timing and node amount.

5.2.4 General analysis

To sum up the results above, not all of them came as expected. It is of course not surprising that all models could solve all J10 instances at least to feasibility with a lot less solution time and node amount needed compared to the J20 set. It can also be stated that the differences in terms of feasible and best solutions found where only marginal in the J10 cases. This is not true for the comparison of the optimal solutions found. Here the event-based models show a worse result in terms of the share of both of these variables, though still not to a great extent.

Major differences across models could only be observed in the solution time and node size differences. This is true for both horizons used in this respect. The main findings inside J10 were that when going from a low to a higher time horizon, the deviation has been most obvious on the DT model, where both solution time and node size increased significantly. With the FCT model staying in the middle, the event-based models showed the least vulnerability to the change in horizon.

This is also partly true for the J20 set, where the differences are far more obvious also in terms of the average solution gap. The increases in timing and node amount from the smaller to the bigger horizon can also be found here. The time indexed models are still fast, the event-based still rank last in terms of solution time and node size, although the differences in the average solution gap are not as big among the models. The OOE model not only loses its dominance towards the SEE version in terms of node size and average solution time for the larger horizon, but also in the numbers of optimal solutions found. Compared to the J10 set also the DT model seems to lose pace compared to the DDT approach.

All in all those general observations show a clear tendency towards three models in particular for each subset, the DDT model for the time-indexed version, the FCT model as best performing (and quite stable) time-continuous formulation and surprisingly the SEE approach in the event-based setting. It seems as if the increase in the time horizon had no significant influence on the ranking among the models, though the problems of course got more complex to solve. This has to be tested further with larger problem instances, J20 still is not the most representative set for this testing, showing only tendencies, but still some open issues and inconsistent model behaviour.

When taking a look at the more detailed results in the parameter analysis, the outcome is actually consistent across the sets and within the models. The results show the same expected and unexpected model behaviour across all sets, delivering more carved-out results of course in the J20 cases.

When comparing the results to the observations made by Koné et al. [2011], some similarities can be found in the multi-mode case. The time-indexed models perform best under the easiest circumstances, showing an inevitable decrease in performance when additional time and activity settings are added. This deterioration is as mentioned only obvious in the high relative increases in solution time and might become more evident in a more complex environment. The FCT model is throughout the sets the best performing time-continuous formulation showing neither the best nor the worst performance. Its ignorance towards the settings of nonrenewable parameters has not been shown or been tested in the article so far.

Despite the observations of Koné et al. [2011], the OOE model is not the best performing event-based model in the multi-mode setting. This can be due to the fact that the additional constraints added to make up for the multi-mode settings make the model more complex. It is also possible that the lack of the preprocessing attributed to the model in Koné et al. [2011] is responsible for the worse performance. It will be a question for further research if preprocessing can accelerate the performance of either the event-based formulations.

The dominance within the time-indexed models is not as straightforward in this work. Tendencies are pointing towards a better performance of the DDT model, but this has to be investigated further with more complex settings like more activities or – as introduced by Koné et al. [2011] – a higher variability within the activity durations.

5.2.5 Preprocessing

The preprocessing steps to determine Earliest and Latest Starting Time Windows which were described in chapter 2 are the most straightforward but also most general calculation method. Also in this context, improvements in the preprocessing phase could be made by a more sophisticated method. Koné et al. [2011] describe an alternate way by introducing precedence and constraint propagation and a parallel schedule scheme heuristic for the starting time window calculation. As the method is not used here, it will not be described in more detail, but should be mentioned nevertheless for the interest of the reader.

Apart from the time window preprocessing that is basically only influencing the performance of the time-indexed formulations DT and DDT, Koné et al. [2011] also mention several preprocessing techniques for the OOE model aiming at reducing the number of events needed. Their solutions range from the removal of events due to precedence requirements on to the reduction of search space through a heuristic variable setting. The preprocessing steps can also here influence the performance of the algorithm and according to the authors there is the possibility of enhancing the used heuristics.

6 Conclusion and further research

As indicated already in this thesis, the analyses conducted here are dedicated to preliminary testing of the five introduced models. The extent of this paper only allowed for the initial comparison of the approaches on two different sets provided by literature. The scope however was to prove whether or not the use of the MILP models for the MRCPSP is useful and if yes, under which circumstances.

The findings of the analyses show promising but inconclusive results. This is on the one hand due to the fact that the sets chosen are not suitable for an in-depth research as they do not support any major improvements through differing algorithms. On the other hand, there is still room for improvement within the model set up, leaving space for better performances.

The main points to target for future research are – among others – preprocessing techniques on the one hand and a differing range of problems sets on the other hand. The preprocessing steps in this thesis are basic and Koné et al. [2011] proposes other approaches for time-window preprocessing and steps for the improvement of the OOE model. When considering the performance within the event-based models, also preprocessing steps for the SEE approach are useful and allow for a fairer comparison of the two.

It should also not be neglected that – although never taking the lead – the FCT model performed well and pretty stable across all sets. This behaviour together with the outstanding insensibility towards changes in the nonrenewable resource parameter set changes, makes this model very promising for future research and adaptations.

The broadening of the scope in terms of instance sets is another important chapter. As already pointed out when discussing the downside of the PSPLIB, a greater variation within the activity duration range will challenge the dominance of the time-indexed models, together with a larger activity amount and probably a higher mode amount for choice. It will be of utmost interest to see how the models compete in a more complex setting.

To sum up, the findings of this paper definitely allow for further research in this topic, also in the multi-mode case. So far the models showed different capabilities in different settings. It will be a challenge for the future research to see whether one of these models can be outstanding with the right preprocessing and – as already shown by Zhu et al. [2006] and the Branch and Cut method – using hybrid methods and heuristic improvements.

References

C. Artigues. A note on time-indexed formulations for the resource-constrained project scheduling problem. 2013.

C. Artigues and F. Roubellat. A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes. *European Journal of Operational Research*, 127(2):297–316, 2000.

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

C. Artigues, P. Brucker, S. Knust, O. Koné, P. Lopez, and M. Mongeau. A note on event-based milp models for resource-constrained project scheduling problems. *Computers & Operations Research*, 40(4):1060–1063, 2013.

J. Blazewicz, J. Lenstra, and A. Rinnooy Kan. Scheduling subejct to resource constraints: classification and complexity. *Discrete Applied Mathematics*, 5: 11–24, 1983.

K. Bouleimen and H. Lecocq. A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research*, 149(2):268–281, 2003.

P. Brucker and S. Knust. *Complex scheduling*. Springer, 2011.

P. Brucker, D. A., R. Moehring, K. Neumann, and E. Pesch. Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112:3–41, 1999.

N. Christofides, R. Alvarez-Valdés, and J. M. Tamarit. Project scheduling with resource constraints: A branch and bound approach. *European Journal of Operational Research*, 29(3):262–273, 1987.

E. Demeulemeester and W. Herroelen. New benchmark results for the resource-constrained project scheduling problem. *Management Science*, 43(11):1485–492, 1997.

S. Hartmann and D. Briskorn. A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1):1–14, 2010.

S. Hartmann and A. Drexl. Project scheduling with multiple modes: A comparison of exact algorithms. *Networks*, 32(4):283–297, 1998.

S. Hartmann and R. Kolisch. Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 127(2):394–407, 2000.

R. Heilmann. A branch-and-bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags. *European Journal of Operational Research*, 144(2):348–365, 2003.

M. L. Hetland. Python algorithms. *Mastering Basic Algorithms in the Python Language, Apress*, 2010.

IBM. Running out of memory.

R. Kolisch and A. Drexl. Local search for nonpreemptive multi-mode resource-constrained project scheduling. *IIE transactions*, 29(11):987–999, 1997a.

R. Kolisch and A. Drexl. Local search for nonpreemptive multi-mode resource-constrained project scheduling. *IIE transactions*, 29(11):987–999, 1997b.

R. Kolisch and S. Hartmann. Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research*, 174(1):23–37, 2006.

R. Kolisch and A. Sprecher. Psplib-a project scheduling problem library: Or software-orsep operations research software exchange program. *European Journal of Operational Research*, 96(1):205–216, 1997.

O. Koné, C. Artigues, P. Lopez, and M. Mongeau. Event-based milp models for resource-constrained project scheduling problems. *Computers & Operations Research*, 38(1):3–13, 2011.

T. Kyriakidis, G. Kopanos, and M. Georgiadis. Milp formulations for single-and multi-mode resource-constrained project scheduling problems. *Computers & Chemical Engineering*, 36:369–385, 2012.

A. Mingozzi, V. Maniezzo, S. Ricciardelli, and L. Bianco. An exact algorithm for the resource constrained project scheduling problem based on a new mathematical formulation. *Management Science*, 44(5):714–729, 1995.

K. Nonobe and T. Ibaraki. A tabu search approach to the constraint satisfaction problem as a general problem solver. *European Journal of Operational Research*, 106(2):599–623, 1998.

A. Pritsker, L. Waiters, and P. Wolfe. Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science*, 16(1):93–108, 1969.

M. Sabzehparvar and S. Seyed-Hosseini. A mathematical model for the multi-mode resource-constrained project scheduling problem with mode dependent time lags. *The Journal of Supercomputing*, 44(3):257–273, 2008.

A. Schutt, T. Feydy, P. J. Stuckey, and M. G. Wallace. Explaining the cumulative propagator. *Constraints*, 16(3):250–282, 2011.

A. Sprecher and A. Drexl. Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm. *European Journal of Operational Research*, 107(2):431–450, 1998.

A. Sprecher, S. Hartmann, and A. Drexl. An exact algorithm for project scheduling with multiple modes. *OR Spektrum*, 19:195–203, 1997.

V. Van Peteghem and M. Vanhoucke. Using resource scarceness characteristics to solve the multi-mode resource-constrained project scheduling problem. *Journal of Heuristics*, 17(6):705–728, 2011.

V. Van Peteghem and M. Vanhoucke. An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances. *European Journal of Operational Research*, 235(1): 62–72, 2014.

J. Zapata, B. Hodge, and G. Reklaitis. The multimode resource constrained multiproject scheduling problem: Alternative formulations. *American Institute of Chemical Engineers Journal*, 54(8):2101–2119, 2008.

G. Zhu, J. F. Bard, and G. Yu. A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem. *INFORMS Journal on Computing*, 18(3):377–390, 2006.

ABSTRACT

The Resource-Constrained Project Scheduling Problem (RCPSP) is a a widely known problem for which various extensions have been introduced to make it more applicable to real-world situations. One of these extensions is the introduction of the multiple performance modes into this problem to examine the different trade-offs in duration and resource consumption.

This thesis aims at extending and testing five different exact Mixed Integer Linear Programming models on this complex problem, each of which has different advantages and disadvantages provided by its setting. The solution approaches have already been tested in the Single Mode case and found to be promising for further research.

In this context the models are now tested on well-known Multi Mode benchmarking instances with parameter variation and compared regarding their solution times, the node amount used to get results and the actual number of optimal solutions found. The outcome does not fully coincide with the one found in the Single-Mode comparison, but similarities are apparent. The analysis also shows promising results, indicating that further research is needed to fully grasp the features of the different models.

ZUSAMMENFASSUNG

Das Resourcenbeschraenkte Projektplanungsproblem (RCPSP) ist ein bekanntes Problem, fuer das bereits verschiedene Erweiterungen eingefuehrt wurden, um es besser an reale Verhaeltnisse anzupassen. Eine dieser Erweiterungen ist die Einfuehrung multipler Modi, um die verschiedenen Auswirkungen auf die Dauer und den Ressourcenverbrauch der einzelnen Aktivitaeten zu untersuchen.

Diese Masterarbeit hat zum Ziel fuenf verschiedene Gemischt-Ganzzahlig Lineare Programmmodelle auf diese komplexe Situation auszuweiten und zu testen. Jedes dieser Modelle hat verschiedene Vor- und Nachteile durch die verwendeten Variablen und Indices. Die Loesungsansaetze wurden bereits in der urspruenglichen Version getestet und als vielversprechend fuer weitere Forschungen bewertet.

In diesem Zusammenhang werden diese Modelle nun an bekannten multimodalen Benchmarking Instanzen mit einer Variation der Parameter getestet und bezueglich der Loesungszeit, der verwendeten Menge an Knoten und der tatsaechlichen Anzahl an gefundenen optimalen Loesungen verglichen. Die Resultate entsprechen nicht voellig den bisher ermittelten Ergebnissen in der urspruenglichen Version, Gemeinsamkeiten sind aber durchgehend vorhanden. Die Analyse zeigt bereits vielversprechende Ergebnisse und verweist auf eine vertiefende Forschung, um die verschiedenen Aspekte der einzelnen Modelle voll zu erfassen.

CURRICULUM VITAE

Personal information:

Name:	Kornelia Artinger
Date of birth:	14.11.1986
Nationality:	Austria

Education:

09/2010 - present:	Master Degree Programme International Business Administration, University of Vienna *Specialisations: Supply Chain Management, International Management*
01/2010 - 06/2010:	Exchange Semester at the Copenhagen Business School, DK
09/2006 - 08/2010:	Bachelor Degree Programme International Business Administration, University of Vienna
2001 - 2006:	Vienna Business School, Floridsdorf, Vienna

Working experience:

09/2014 - present:	DB Schenker Austria European Land Transport, System Freight CEE
08/2013 - 07/2014:	Henkel Central Eastern Europe GmbH Internship Processes & Operations Purchasing CEE
07/2012 - 09/2012:	Hamburg Sued, Hamburg, Germany Internship Region Europe, Trade Management
03/2012 - 01/2013:	University of Vienna, Department of Business Administration Student Assistant at the Chair of Production and Operations Management
10/2011 - 01/2012:	University of Vienna, Department of Business Administration Student Assistant at the Chair of Supply Chain Management
09/2008 - 12/2009:	University of Vienna, Department of Business Administration Student Assistant at the Division of eBusiness
03/2008 - 07/2008:	University of Vienna, Department of Business Administration Tutor at the Division of eBusiness

Additional skills:

German	Native language
English	(Business) Fluent
Italian:	Proficient
Russian	Moderate
French	Basic skills
IT Skills:	Proficient knowledge in SAP (SAP ERP, SAP APO), Xpress-MP and Python, Intermediate skills in using Latex, Basic knowledge in SPSS