A comparison of sequencing formulations in a constraint generation procedure for avionics scheduling

Department of Mathematics, Linköping University

Jessika Boberg

LiTH-MAT-EX–2017/18–SE

Credits: 16 hp
Level: G2
Supervisors: Emil Karlsson, Elina Rönnberg,
Department of Mathematics, Linköping University
Andreas Stenberg,
Saab AB
Examiner: Elina Rönnberg,
Department of Mathematics, Linköping University
Linköping: December 2017
Abstract

This thesis compares different mixed integer programming (MIP) formulations for sequencing of tasks in the context of avionics scheduling. Sequencing is a key concern in many discrete optimisation problems, and there are numerous ways of accomplishing sequencing with different MIP formulations.

A scheduling tool for avionic systems has previously been developed in a collaboration between Saab and Linköping University. This tool includes a MIP formulation of the scheduling problem where one of the model components has the purpose to sequence tasks. In this thesis, this sequencing component is replaced with other MIP formulations in order to study whether the computational performance of the scheduling tool can be improved.

Different scheduling instances and objective functions have been used when performing the tests aiming to evaluate the performances, with the computational times of the entire avionic scheduling model determining the success of the different MIP formulations for sequencing. The results show that the choice of MIP formulation makes a considerable impact on the computational performance and that a significant improvement can be achieved by choosing the most suitable one.

**Keywords**: Scheduling, avionics scheduling, multiprocessor scheduling, mixed integer programming.

**URL for electronic version**: http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-143274
I would like to thank my supervisors Elina, Emil and Andreas. Their support and guidance throughout this thesis have taken it to new heights and I could not have done it without their enthusiasm and encouragement.
Contents

1 Introduction .................................................. 1
  1.1 Background .......................................... 1
  1.2 Purpose ............................................. 2
  1.3 Method ............................................. 2
  1.4 Limitations ......................................... 2
  1.5 Chapter outline ................................... 2

2 An avionics scheduling problem ......................... 3
  2.1 System model ....................................... 3
    2.1.1 AM ........................................... 5
    2.1.2 CM ........................................... 5
    2.1.3 Precedence relations ......................... 5
    2.1.4 Communication network ....................... 5
  2.2 Constraint generation ............................... 6
    2.2.1 $\alpha$-model ................................ 7
    2.2.2 $\beta$-model ................................ 7
  2.3 Sequencing component .............................. 8
    2.3.1 MIP formulation for AM tasks ............... 8
    2.3.2 MIP formulation for CM tasks ............... 9
    2.3.3 Generated sequences ......................... 10

3 MIP formulations for sequencing ....................... 11
  3.1 Completion time variables ......................... 12
  3.2 Assignment and positional date variables .......... 12
  3.3 On/off event-based formulation ................... 14
  3.4 Other formulations ................................ 15
    3.4.1 Linear ordering variables ................... 15
    3.4.2 Time index variables ......................... 16
    3.4.3 Start/End event-based ....................... 16

Boberg, 2017. vii
Chapter 1

Introduction

1.1 Background

The concept of sequencing is fundamental in many optimisation problems and is a key concern in several problem areas such as job shop scheduling, vehicle routing and multiprocessor scheduling. As such it has been studied a great length and there exist numerous ways of accomplishing sequencing with different mixed-integer problem (MIP) formulations, each with different characteristics.

This thesis is part of an ongoing research collaboration between Saab AB and Linköping university, developing solution methods for multiprocessor scheduling problems that are relevant for the development of future avionic systems (the electronic devices of an aircraft). The project has created a scheduling tool based on MIP formulations and constraint generation, aiming to find a feasible schedule or provide information if no such exists. An important component of this tool is the mathematical model for sequencing of tasks. This thesis will study different sequencing formulations within the scheduling tool and evaluate which formulation that achieves the best computational performance in this context.

Avionic systems are often large and complex, and new software functionality is added iteratively, which is why the scheduling tool needs to be efficient. The instances aimed to be scheduled are of industrial relevance, provided by Saab and contain up to 20 000 tasks. In this thesis, the mathematical model implemented in the scheduling tool, presented in [1], is considered to be given and the background of this model will therefore not be discussed in detail.
1.2 Purpose

The purpose of this bachelor thesis is to study if the computational performance of a scheduling tool can be improved by using different sequencing formulations.

1.3 Method

Commonly used mathematical models for sequencing will be adapted to the constraint generation procedure which the scheduling tool is based on, and the sequencing component of the tool will then be replaced by the adapted models. Different objective functions and scheduling instances of industrial relevance will be used when performing the tests in order to evaluate the performances. It is primarily the computational times of the entire avionics scheduling model that will be used to determine how well the different mathematical models for sequencing performs in this context.

1.4 Limitations

The duration of the project is set to a total of 10 weeks, which limits the number of formulations that will be studied. Since the thesis aims to compare the formulations with an already existing model as a base, the result of this thesis can only be considered valid in this context. Which MIP formulation that performs best for other scheduling problems or avionic systems remains to be further investigated.

1.5 Chapter outline

In Chapter 2, the scheduling tool and the existing avionics scheduling model is described. Chapter 3 is a literature review of the MIP formulations that have been studied. Chapter 4 suggests how the MIP formulations in Chapter 3 can be adapted to the avionics scheduling model in Chapter 2 and the computational performance of the different formulations are presented in Chapter 5. In Chapter 6, the obtained results are discussed and a conclusion is made based on the results and discussion.
Chapter 2

An avionics scheduling problem

This chapter contains an overview of the solution approach and the mathematical model implemented in the scheduling tool developed at Saab. It also contains an overview of the multiprocessor scheduling problem with precedence relations and a communication network, that the scheduling tool aims to find a feasible schedule for. A feasible schedule is represented by a starting time for each task and a time slot for each communication message. For a more detailed description of the problem and solution approach, see [1].

The first section of this chapter describes the system model, including the task system, precedence relations and communication network. An overview of the solution approach based on a MIP formulation and constraint generation is presented in the second and third section.

2.1 System model

In this section, the system model from [1] is described. Firstly the two types of modules, where the tasks are executed, are introduced. Secondly, two other parts of the system (precedence relations and a communication network) are briefly explained.

The system consists of a set of nodes and executes periodically with period P, where each period is called a major frame. Each node has a set of modules (processors), consisting of one communication module (CM) and a set of application modules (AMs). The tasks are already assigned to the modules and no migration of tasks between the modules is allowed. The set of all tasks is denoted $\mathcal{I}$. 
### Table 2.1: The parameters and sets introduced in Chapter 2

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{I}$</td>
<td>set of tasks</td>
</tr>
<tr>
<td>$\mathcal{H}^{CM}$</td>
<td>set of CMs</td>
</tr>
<tr>
<td>$\mathcal{H}^{AM}$</td>
<td>set of AMs</td>
</tr>
<tr>
<td>$\mathcal{F}^{fix}$</td>
<td>set of fix tasks</td>
</tr>
<tr>
<td>$p_i$</td>
<td>execution requirement for task $i$</td>
</tr>
<tr>
<td>$t_i^{r}$</td>
<td>release time for task $i$</td>
</tr>
<tr>
<td>$t_i^{d}$</td>
<td>deadline for task $i$</td>
</tr>
<tr>
<td>$t_i^{r}$</td>
<td>release time for task $i$ in section $r$</td>
</tr>
<tr>
<td>$t_i^{d}$</td>
<td>deadline for task $i$ in section $r$</td>
</tr>
<tr>
<td>$t_{ij}^{idle}$</td>
<td>idle time between task $i$ and task $j$</td>
</tr>
<tr>
<td>$q_{\text{start}}$</td>
<td>first task of each sequence</td>
</tr>
<tr>
<td>$q_{\text{end}}$</td>
<td>last task of each sequence</td>
</tr>
<tr>
<td>$\mathcal{S}_h$</td>
<td>subsets on CM $h$</td>
</tr>
<tr>
<td>$\mathcal{R}_h$</td>
<td>sections on CM $h$</td>
</tr>
<tr>
<td>$\mathcal{F}_{\text{sub}}$</td>
<td>tasks in subset $s$</td>
</tr>
<tr>
<td>$\mathcal{I}_i^{+}$</td>
<td>all possible immediate successors of task $i$</td>
</tr>
<tr>
<td>$\mathcal{I}_i^{-}$</td>
<td>all possible immediate predecessors of task $i$</td>
</tr>
<tr>
<td>$\mathcal{S}^{\text{gen}}$</td>
<td>set of tasks added in generated sequences</td>
</tr>
</tbody>
</table>
Every task $i$ is executed one time every period with an execution requirement (duration) $p_i$, a release time (earliest starting time) $t^r_i$ and a deadline (latest completion time) $t^d_i$. Task $i$ generates an infinite succession of instances that execute once every period for the duration of its execution requirement. The tasks on a specific module have to be scheduled without overlapping each other and no pre-emption of tasks is allowed (to be interrupted and then resumed at a later stage).

2.1.1 AM

The AMs contain a small amount of tasks with long execution requirements, all with a period of $P/64$. The set of all AMs is denoted $\mathcal{H}_\text{AM}$ and the set of tasks assigned to AM $h$ is denoted $\mathcal{I}_h$, $h \in \mathcal{H}_\text{AM}$. Each pair of AM tasks need to have an idle time between them, denoted $l_{ij}^{\text{idle}}$ for the time needed between the completion time of task $i$ and starting time of task $j$ if $i$ executes before task $j$ in a major frame, where $i \in \mathcal{I}_h$, $j \in \mathcal{I}_h$ and $h \in \mathcal{H}_\text{AM}$.

2.1.2 CM

The CMs contain a large amount of tasks with short execution requirements, and an important characteristic for the solution approach used in [1] is that a large portion of these tasks are fixed. The set of all CMs is denoted $\mathcal{H}_\text{CM}$ and the set of tasks assigned to CM $h$ is denoted $\mathcal{I}_h$, $h \in \mathcal{H}_\text{CM}$. The set of fixed tasks on CM $h$ is denoted $\mathcal{I}^{\text{fix}}_h$, $h \in \mathcal{H}_\text{CM}$. Every task on a CM has the period of a major frame.

2.1.3 Precedence relations

There are two sets of precedence relations, called chains and dependencies. Tasks belonging to the same chain have to be executed in a specific cycle, where any of them can be chosen as the first one. The dependencies restrict the time between the start of a task instance to the start of an instance of another task. Constraints to manage both dependencies and chains are covered in [1], in this thesis they are referred to as "Precedence relations constraints".

2.1.4 Communication network

The communication between the nodes is managed through a communication network (CN), and each node is connected to the CN through its CM. The CN-messages sent between the nodes have to be assigned to discrete CN-slots and
each CM can only send or receive one CN-message in each CN-slot. The constraint handling the CN messages are referred to as "CN-scheduling constraints" and are covered in [1].

2.2 Constraint generation

The mathematical model used to schedule the tasks on CMs is formulated in a way that results in a huge number of constraints due to the large amount of CM tasks. To avoid creating redundant constraints and variables, the data input is firstly processed by a pre-processing component described in [1]. Two different models are then used to generate constraints systematically instead of all constraints being created at once. These models are presented in following subsections as the \( \alpha \)- and \( \beta \)-model, followed by the MIP formulations used for AM and CM tasks.

In the \( \alpha \)-model, sections on the CMs are created in-between the fix CM tasks, and non-fixed tasks are assigned to these sections. The set of all sections on CM \( h \) is denoted \( R_h \), \( h \in \mathcal{H}_{CM} \). The \( \beta \)-model then sequences the non-fixed tasks within their given section. If some tasks cannot be sequenced without overlaps occurring, constraints are added to both the \( \alpha \)- and \( \beta \)-model ensuring that these are forced to be sequenced without overlaps. This is illustrated in Figure 2.1.

![Figure 2.1: Solution approach based on constraint generation.](image-url)
2.2. Constraint generation

2.2.1 $\alpha$-model

The $\alpha$-model is a relaxation of the complete mathematical model since it might not sequence all CM tasks. The aim of the $\alpha$-model is to place non-fixed CM tasks into sections while taking all other constraints in consideration. In this thesis different objective functions to guide the assignment is used, these are described in [1]. A consequence of the relaxation is that there are no generated sequenced in the first iteration, but in later iterations the ones identified by the $\beta$-model are included. The complete structure of the $\alpha$-model can be described as:

$$\begin{align*}
\text{min}/\text{max} & \quad \text{Objective function} \\
\text{s.t.} & \quad \text{Objective constraints} \\
& \quad \text{Precedence relations constraints} \\
& \quad \text{CN-scheduling constraints} \\
& \quad \text{CM-assignment constraints} \\
& \quad \text{AM-scheduling} \quad (\text{Constraint 2.1 - 2.5}) \\
& \quad \text{Generated sequences} \quad (\text{Constraint 2.13 - 2.16})
\end{align*}$$

2.2.2 $\beta$-model

The aim of the $\beta$-model is to sequence the CM tasks within the sections obtained from a solution to the $\alpha$-model. If sequencing of these tasks cannot be done without overlapping for some section, "generated sequences" are added to both the $\alpha$- and the $\beta$-model.

With an objective function that maximizes the number of successfully sequenced tasks, the $\beta$-model sequences as many tasks as possible given a solution from the $\alpha$-model. The complete structure of the $\beta$-model can be summarized as:

$$\begin{align*}
\text{max} & \quad \text{Number of successfully sequenced tasks} \\
\text{s.t.} & \quad \text{CN-scheduling constraints} \\
& \quad \text{Precedence relations constraints} \\
& \quad \text{AM-scheduling} \quad (\text{Constraint 2.1-2.5}) \\
& \quad \text{Generated sequences} \quad (\text{Constraint 2.13-2.16}) \\
& \quad \beta\text{-sequences} \quad (\text{Constraint 2.6-2.12})
\end{align*}$$
2.3 Sequencing component

In this section, the mathematical models for the sequencing of AM and CM tasks are presented. These are the parts of the problem that will be replaced by different sequencing formulations later on in this thesis. The MIP formulations presented here will be referred to as Formulation 0. In order to simplify the notation for the sequencing of both AM and CM tasks, the two artificial tasks $\tilde{q}_{\text{start}}$ and $\tilde{q}_{\text{end}}$ with execution requirement 0 are set to be the first and last tasks, respectively, of all sequences. A variable for task $i$, $i \in \mathcal{I}$, is also introduced:

$$x_i = \text{start time of task } i \text{ offset its period start.}$$

2.3.1 MIP formulation for AM tasks

In this section, the MIP formulation sequencing the AM tasks is presented. The first part of the sequencing consists of the pre-processing, that for each AM creates the sets $\mathcal{I}_i^+$ and $\mathcal{I}_i^-$, denoting all tasks that can be the immediate successor and immediate predecessor of task $i$, respectively, $i \in \mathcal{I}_h$, $h \in \mathcal{H}_{\text{AM}}$, and is described in [1]. A binary variable is then introduced for $h \in \mathcal{H}_{\text{AM}}$, $i \in \mathcal{I}_h$, $j \in \mathcal{I}_i^+$:

$$y_{ij} = \begin{cases} 
1 & \text{if task } i \text{ is the immediate predecessor of task } j, \\
0 & \text{otherwise.}
\end{cases}$$

The following constraints then sequence the AM tasks:

$$\sum_{j \in \mathcal{I}_i^+} y_{ij} = 1, \quad i \in \mathcal{I}_h \setminus \{\tilde{q}_{\text{end}}\}, \ h \in \mathcal{H}_{\text{AM}}, \ (2.1)$$

$$\sum_{j \in \mathcal{I}_i^-} y_{ji} = 1, \quad i \in \mathcal{I}_h \setminus \{\tilde{q}_{\text{start}}\}, \ h \in \mathcal{H}_{\text{AM}}, \ (2.2)$$

$$x_i + p_i + l_{ij}^{\text{idle}} \leq x_j + M(1 - y_{ij}), \quad j \in \mathcal{I}_i^+, \ i \in \mathcal{I}_h, \ h \in \mathcal{H}_{\text{AM}}, \ (2.3)$$

$$t_{i}^l \leq x_i \leq t_{i}^d - p_i, \quad i \in \mathcal{I}_h, \ h \in \mathcal{H}_{\text{AM}}, \ (2.4)$$

$$y_{ij} \in \{0,1\}, \quad j \in \mathcal{I}_i^+, \ i \in \mathcal{I}_h, \ h \in \mathcal{H}_{\text{AM}} \ (2.5)$$

The first two constraints make sure that all tasks have one immediate successor and one immediate predecessor. The third constraint ensures that a task cannot start until its immediate predecessor has finished and the needed idle time has passed, and the fourth constraint ensures that all tasks are executed within their given time window. For the sequencing of AM tasks, $M = t_i^d - t_j^l + l_{ij}^{\text{idle}}$. 
2.3. Sequencing component

2.3.2 MIP formulation for CM tasks

This MIP formulation is used for the CM tasks in the $\beta$-model, when tasks have already been assigned to sections, and aims to sequence all tasks that belongs to the same section. For the CM tasks, the set $S_h$ is introduced, such that each set $I_{i,s}^h$, $s \in S_h$, $h \in H^{CM}$, includes non-fixed tasks on CM $h$ that can, for at least one section, be assigned together in the same section. The set $I_{i,s}^+$ denotes all tasks that can be the immediate successor of task $i$ in subset $s$, 

$S_h$ is introduced, such that each set $I_{s,h}^-$ includes non-fixed tasks on CM $h$ that can, for at least one section, be assigned together in the same section. The set $I_{i,s}^-$ denotes all tasks that can be the immediate predecessor of task $i$ in subset $s$.

A binary variable is introduced, for $i \in I_{i,s}^-$:

$$ y_{ij,s} = \begin{cases} 
1 & \text{if task } i \text{ is the immediate predecessor of task } j \text{ in subset } s, \\
0 & \text{otherwise.} 
\end{cases} $$

The following binary variable is also introduced, for $i \in I_{i,s}^+$:

$$ \beta_{i,s} = \begin{cases} 
1 & \text{if task } i \text{ is successfully sequenced in subset } s, \\
0 & \text{otherwise.} 
\end{cases} $$

Let the subset of tasks assigned to section $r$ be denoted by $\bar{s}_r$, $r \in R_h$, $h \in H^{CM}$. Each subset of tasks is then sequenced by the following constraints, with the release time and the deadline of task $i$ restricted to $t_{i,r}^r$ and $t_{d,i}^r$, respectively, $i \in I_{\bar{s}_r}^s$, $r \in R_h$, $h \in H^{CM}$.

$$ \sum_{j \in I_{j,s}^+} y_{ij,s} = \beta_{i,s}, \quad i \in I_{i,s}^+ \setminus \{\bar{q}_{\text{end}}\}, \ r \in R_h, \ h \in H^{CM}, \quad (2.6) $$

$$ \sum_{j \in I_{j,s}^-} y_{ij,s} = \beta_{i,s}, \quad i \in I_{i,s}^- \setminus \{\bar{q}_{\text{start}}\}, \ r \in R_h, \ h \in H^{CM}, \quad (2.7) $$

$$ x_i + p_i \leq x_j + M(1 - y_{ij,s}), \quad j \in I_{j,s}^+ \setminus \{\bar{q}_{\text{end}}\}, \ r \in R_h, \ h \in H^{CM}, \quad (2.8) $$

$$ t_{i,r}^r \leq x_i \leq t_{d,i}^r - p_i, \quad \beta_{i,s} \in \{0,1\}, \quad i \in I_{i,s}^+, \ r \in R_h, \ h \in H^{CM}, \quad (2.9) $$

$$ y_{ij,s} \in \{0,1\}, \quad \beta_{i,s} \in \{0,1\}, \quad j \in I_{i,s}^+ \setminus \{\bar{q}_{\text{end}}\}, \ i \in I_{i,s}^+ \setminus \{\bar{q}_{\text{end}}\}, \ r \in R_h, \ h \in H^{CM}, \quad (2.11) $$

$$ x_i = t_{i,r}^r, \quad \beta_{i,s} \in \{0,1\}, \quad i \in I_{i,s}^-, \ h \in H^{CM}, \quad (2.10) $$

This formulation is referred to as $\beta$-sequences where $\beta_{i,s} = 0$ indicate that task $i$ could not be ensured not to overlap another task in subset $\bar{s}_r$, $r \in R_h$, $h \in H^{CM}$. For the sequencing of CM tasks, $M = t_{d,i}^r - t_{i}^r$. 


2.3.3 Generated sequences

Following constraints are referred to as Generated sequences, with the set $S^{\text{gen}}$ containing all tasks in a section $r$ that, for some solution of the $\alpha$-model, has generated at least one $\beta_{i_{\bar{s}}r} = 0$, $r \in R_h$, $h \in H^{\text{CM}}$:

$$\sum_{j \in I_{is}^+} y_{ij_{s}} = 1, \quad i \in I^{\text{sub}}_s \setminus \{\tilde{q}_{\text{end}}\}, \quad s \in S^{\text{gen}}, \quad (2.13)$$

$$\sum_{j \in I_{is}^-} y_{j_{is}} = 1, \quad i \in I^{\text{sub}}_s \setminus \{\tilde{q}_{\text{start}}\}, \quad s \in S^{\text{gen}}, \quad (2.14)$$

$$x_i + p_i \leq x_j + M(1 - y_{ij_{s}}), \quad j \in I_{is}^+, \quad i \in I^{\text{sub}}_s, \quad s \in S^{\text{gen}}, \quad (2.15)$$

$$y_{ij_{s}} \in \{0, 1\}, \quad j \in I_{is}^+, \quad i \in I^{\text{sub}}_s, \quad s \in S^{\text{gen}} \quad (2.16)$$

Just like for the $\beta$-sequences, $M = t^d_i - t^r_j$. Since these constraints can be obtained by setting $\beta_{i_{\bar{s}}r} = 1$ in the MIP formulation in Section 2.3.2, the Generated sequences will for the rest of this thesis be assumed to be created by doing so. This leads to the requirement that the MIP formulations for sequencing of CM tasks will have to include a $\beta$-variable with a corresponding variable definition as the one in Section 2.3.2.
Chapter 3

MIP formulations for sequencing

In this chapter, the MIP formulations for sequencing that have been studied for this thesis are presented. Section 3.1-3.3 describes the formulations that have been adapted in order to replace the sequencing component in Section 2.3.2, while the Section 3.4 briefly mentions the ones assessed as not suitable for this purpose.

The first two formulations, among others, have been compared in [2], with the statement that the first one (Completion time variables) often appears in textbooks and other literature but does not generally perform well in practice, while the second formulation (Assignment and positional date variables) is the choice they prefer. It is also stated that the performances of the formulations depend on the number of tasks, total execution time of all tasks and the objective functions.

Different objective functions can be used for all formulations presented below, some with the need to add more constraints. However, since the scheduling problem in this thesis focuses on finding a feasible schedule rather than minimizing the make span, these are not discussed any further here.

In this chapter, the same notations for parameters as in Chapter 2 is used, meaning that $p_i$, $t_i$ and $t_i^d$ denotes the execution requirement, release time and deadline for task $i$, respectively. However, since the sets and variables are specific for each MIP formulation, these are introduced separately in each section, with the notations in this chapter mainly inspired from [2] and [6].
3.1 Completion time variables

This formulation, first described for single machine scheduling problems in [3] but here written as in [2], is based on a variable representing the completion time of each task and a binary variable deciding a predecessor and a successor for each pair of tasks. The set $\mathcal{N}$ contains all tasks, numbered from 1 to $n$. For $i \in \mathcal{N}$, $j \in \mathcal{N}$ and $i < j$, the binary variable

$$r_{ij} = \begin{cases} 1 & \text{if task } i \text{ is executed before task } j, \\ 0 & \text{if task } j \text{ is executed before task } i \end{cases}$$

is introduced. For $i \in \mathcal{N}$, the following continuous variable is also presented:

$$C_i = \text{completion time of task } i.$$

Then the following constraints are presented in order to sequence the tasks:

$$C_i \geq p_i + t_i, \quad i \in \mathcal{N}, \quad (3.1)$$
$$C_i + p_j \leq C_j + M(1 - r_{ij}), \quad i, j \in \mathcal{N}, \quad i < j, \quad (3.2)$$
$$C_j + p_i \leq C_i + Mr_{ij}, \quad i, j \in \mathcal{N}, \quad i < j, \quad (3.3)$$
$$C_i \geq 0, \quad i \in \mathcal{N}, \quad (3.4)$$
$$r_{ij} \in \{0, 1\}, \quad i, j \in \mathcal{N}, \quad i < j \quad (3.5)$$

Constraint (3.1) ensures that no task can start before its release time, and (3.2) and (3.3) states that if task $i$ is executed before task $j$, the starting time of task $j$ is greater than or equal to the completion time of task $i$, and vice versa. According to [2], $M$ is generally set greater than the sum of all executing times and the maximum value of release times.

3.2 Assignment and positional date variables

In the formulation presented in [4], the idea is to decide in which order the tasks should be executed by assigning them to so-called positions, with "task $i$ is assigned to position $k$" meaning that $i$ is the $k$-th task to be executed. In [5], a similar position-based sequencing formulation is presented, but with an added continuous variable that sorts out which tasks are the immediate successors and predecessors of each other. It is this formulation that is presented in this section.

Let the set $\mathcal{N}$ contain all tasks, and the set $\mathcal{P}$ contain all positions numbered from 1 to $n$, $n$ being the total number of tasks. For $j \in \mathcal{N}$, $k \in \mathcal{P}$, following
variable is introduced:

\[ u_{jk} = \begin{cases} 
1 & \text{if task } j \text{ is assigned to position } k, \\
0 & \text{otherwise.} 
\end{cases} \]

The following variable is also introduced, for \( k \in \mathcal{P} \):

\[ \varphi_k = \text{completion time of the task assigned to position } k. \]

The third variable introduced is a continuous variable, but due to the constraint structure it will only become 1 or 0. For \( i \in \mathcal{N}, j \in \mathcal{N} \) and \( k \in \mathcal{P} \), introduce following variable:

\[ q_{ijk} = \begin{cases} 
1 & \text{if task } j \text{ is assigned to position } k \text{ and } \text{task } i \text{ is assigned to position } k - 1, \\
0 & \text{otherwise.} 
\end{cases} \]

In order to sequence the tasks, following constraints are introduced:

\[ \sum_{k \in \mathcal{P}} u_{jk} = 1, \quad j \in \mathcal{N}, \]  \hspace{1cm} (3.6)

\[ \sum_{j \in \mathcal{N}} u_{jk} = 1, \quad k \in \mathcal{P}, \]  \hspace{1cm} (3.7)

\[ \varphi_1 \geq \sum_{j \in \mathcal{N}} p_j u_{j1}, \]  \hspace{1cm} (3.8)

\[ \varphi_k \geq \varphi_{k-1} + \sum_{j \in \mathcal{N}} p_j u_{jk}, \quad k \in \mathcal{P} \setminus \{1\}, \]  \hspace{1cm} (3.9)

\[ \varphi_k \geq \sum_{j \in \mathcal{N}} (p_j + t^*_j) u_{jk}, \quad k \in \mathcal{P}, \]  \hspace{1cm} (3.10)

\[ x_j + p_j \geq \varphi_k - M (1 - u_{jk}), \quad j \in \mathcal{N}, \quad k \in \mathcal{P}, \]  \hspace{1cm} (3.11)

\[ \sum_{i \in \mathcal{N}} q_{ijk} = u_{jk}, \quad j \in \mathcal{N}, \quad k \in \mathcal{P} \setminus \{1\}, \]  \hspace{1cm} (3.12)

\[ \sum_{i \in \mathcal{N}} q_{jik} = u_{jk} - 1, \quad j \in \mathcal{N}, \quad k \in \mathcal{P} \setminus \{1\}, \]  \hspace{1cm} (3.13)

\[ \varphi_k \geq 0, \quad k \in \mathcal{P}, \]  \hspace{1cm} (3.14)

\[ q_{ijk} \geq 0, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}, \quad k \in \mathcal{P}, \quad i \neq j, \]  \hspace{1cm} (3.15)

\[ u_{jk} \in \{0, 1\}, \quad j \in \mathcal{N}, \quad k \in \mathcal{P} \]  \hspace{1cm} (3.16)
The first two constraints ensure that each task is assigned to exactly one position and that each position has exactly one task assigned to it. Constraints (3.8)–(3.9) ensure that no task can be finished before the finishing time of the previous task plus its own processing time. Constraint (3.10) ensures that no task can start before its release time, and Constraint (3.11) is added in the cases where the objective function is to minimize the completion time of the last task. Constraints (3.14) and (3.15) link the $q$-variables to the $u$-variables, and will also ensure that $q_{ijk}$ will only become 1 or 0.

### 3.3 On/off event-based formulation

In [6], a formulation for resource-constrained project scheduling problems is introduced. The idea is to have so called events on which the tasks can start, and a binary variable indicating if a specific task starts at or is still being executed directly after a specific event. A continuous variable then decides on what time every event takes place.

As presented in [6], this formulation includes constraints similar to dependencies. These are not included here since the dependencies are already handled elsewhere in the scheduling tool. This formulation allows for several tasks to be executed at once, as long as they do not exceed the limited amount of resources that the problem specifies. The set $\mathcal{E}$ contains all events numbered from 0 to $n - 1$ and the set $\mathcal{N}$ contains all tasks numbered from 0 to $n - 1$, $n$ being the number of tasks. The set $\mathcal{R}$ denotes all resources, the parameter $B_k$ denotes the available amount of resource $k$ and the parameter $b_{ik}$ represents how much of resource $k$ is used during the execution of task $i$, $k \in \mathcal{R}$, $i \in \mathcal{N}$. For $i \in \mathcal{N}$ and $e \in \mathcal{E}$, following binary variable is introduced:

$$z_{ie} = \begin{cases} 1 & \text{if task } i \text{ starts at and/or continues directly after event } e, \\ 0 & \text{otherwise.} \end{cases}$$

For $e \in \mathcal{E}$, following continuous variable represents the time for each event:

$$\sigma_e = \text{time event } e.$$

Then the sequencing requirement can be expressed with the following constraints.

$$\sum_{e \in \mathcal{E}} z_{ie} \geq 1, \quad i \in \mathcal{N}, \quad (3.17)$$

$$\sigma_{e+1} \geq \sigma_e, \quad e \in \mathcal{E} \setminus \{n\}, \quad (3.18)$$

$$\sigma_0 = 0, \quad (3.19)$$
\[ \sigma_f \geq \sigma_e + ((z_{ie} - z_{i,e-1}) - (z_{if} - z_{i,f-1}) - 1) p_i, \]
\[ i \in \mathcal{N}, \ e \in \mathcal{E}, \ f \in \mathcal{E}, \ e < f, \quad (3.20) \]

\[ \sum_{e'=0}^{e-1} z_{ie'} \leq e(1 - (z_{ie} - z_{i,e-1})), \quad e \in \mathcal{E} \setminus \{0\}, \quad (3.21) \]

\[ \sum_{e'=e}^{n-1} z_{ie'} \leq (n - e)(1 + (z_{ie} - z_{i,e-1})), \quad e \in \mathcal{E} \setminus \{0\}, \quad (3.22) \]

\[ \sum_{i=0}^{n-1} b_{ik} z_{ie} \leq B_k, \quad e \in \mathcal{E}, \ k \in R, \quad (3.23) \]

\[ t_i^e z_{ie} \leq \sigma_e \leq (t_i^d - p_i)(z_{ie} - z_{i,e-1}) + (t_n^d - p_n)(1 - (z_{ie} - z_{i,e-1})), \]
\[ e \in \mathcal{E}, \ i \in \mathcal{N}, \quad (3.24) \]

\[ \sigma_e \geq 0, \quad e \in \mathcal{E}, \quad (3.25) \]

\[ z_{ie} \in \{0, 1\}, \quad i \in \mathcal{N}, \ e \in \mathcal{E} \quad (3.26) \]

The first constraint ensures that all tasks are executed at one event or more. Constraint (3.20) ensures that if task \( i \) starts at event \( e \) and finishes before event \( f \), then \( t_f \geq t_e + p_i \). Constraints (3.21) and (3.22) ensure no pre-emption of tasks and Constraint (3.23) limits the amount of resources used at once. Lastly, Constraint (3.24) ensures that the starting time for each task occurs within its allowed time window.

### 3.4 Other formulations

This section briefly describes the MIP formulations that have been studied for this thesis but were decided not to be implemented in the avionic scheduling tool. In Chapter 4, it is further discussed why they were considered unsuitable for this purpose.

#### 3.4.1 Linear ordering variables

This formulation, first described for single machine scheduling problems in [7] and one of the MIP formulations being compared in [2], is similar to the Completion time variable formulation presented in Section 3.1 since it is also based
on a variable $r_{ij}$ that is equal to 1 if task $i$ precedes task $j$ and 0 otherwise. However, its LP-relaxation is improved compared to the Completion time variable formulation.

### 3.4.2 Time index variables

This formulation was first described for single machine scheduling problems in [8], and is one of the MIP formulations being compared in [2]. It discretizes the planning horizon into periods 1, 2, 3, ... $T$, where period $t$ starts at time $t - 1$ and ends at time $t$. The sequencing is then based on a binary variable that decides whether a task starts on a specific period or not.

### 3.4.3 Start/End event-based

This formulation, presented in [6] and later corrected in [9], has several similarities with the On/Off event-based formulation presented in Section 3.3. However, it has two separate binary variables - one representing the events on which tasks can start and the other representing the events on which they can finish. The two event-based formulations are compared in [6], with the conclusion that the On/Off formulation outperforms the Start/End formulation on all tested instance sets.
Chapter 4

Adaptation of MIP formulations for sequencing

This chapter describes how the MIP formulations for scheduling presented in Chapter 3 can be adapted in order to replace the sequencing component in Section 2.3. For each formulation, two different adaptations are presented due to the different characteristics of AM and CM scheduling. Notations from Chapter 2 are used, meaning that the set of tasks assigned to AM \( h \) is denoted \( \mathcal{I}_h, h \in \mathcal{H}^{AM} \) and the set of tasks assigned to CM \( h \) is denoted \( \mathcal{I}_h, h \in \mathcal{H}^{CM} \). The set of all sections on CM \( h \) is denoted \( \mathcal{R}_h, h \in \mathcal{H}^{CM} \), and each set \( \mathcal{I}_{s^{sub}}, s \in \mathcal{S}_h, h \in \mathcal{H}^{CM} \) includes non-fixed tasks on CM \( h \) that can, for at least one section, be assigned together in the same section. Finally, the subset of tasks assigned to section \( r \) is denoted by \( \bar{s}_r, r \in \mathcal{R}_h, h \in \mathcal{H}^{CM} \).

When adapting MIP formulations to the sequencing component, there are two things to consider. Firstly, the following variable have to be included in every formulation, for \( i \in \mathcal{I} \):

\[
x_i = \text{start time of task } i \text{ offset its period start.}
\]

Secondly, all CM formulations must include a binary variable, \( \beta \), representing whether a task have been successfully sequenced or not. This is due to the structure for creating the Generated sequences presented in Section 2.3.3.

4.1 Completion time variables

In this section, the Completion time variables formulation presented in Section 3.1 is altered so it can replace the sequencing component in Section 2.3. This
formulation is referred to as Formulation 1 in the rest of this thesis.

The Completion time variables formulation assumes that all tasks can be sequenced without overlaps, which is why only \( r \)-variables where \( i < j \) are created since either task \( i \) will be executed before task \( j \) or vice versa. In order to allow for tasks to not be sequenced, \( r \)-variables for all combinations of \( i \) and \( j, i \in I_h, j \in I_h, i \neq j \), are created in our adaptation.

### 4.1.1 AM

Introduce, for \( h \in H^{AM} \), \( i \in I_h, j \in I_h \) and \( i \neq j \), following binary variable:

\[
    r_{ij} = \begin{cases} 
    1 & \text{if task } i \text{ is processed before task } j, \\
    0 & \text{otherwise.} 
    \end{cases}
\]

In the pre-processing component, let every \( r_{ij} \) where \( t_i^d < t_j^e \) be set to 1 and corresponding \( r_{ji} \) to 0. Let the set \( I^+_i \) denote all tasks that are possible successors of task \( i, i \in I_h \) (that is, all tasks \( j \in I_h, j \neq i \), where \( r_{ij} \) has not been given a value). The following constraints sequence the tasks:

\[
    t_i^r \leq x_i \leq t_i^d - p_i, \quad i \in I_h, \ h \in H^{AM}, \ (4.1)
\]

\[
    x_j \geq x_i + p_i + l_{ij}^{idle} - M(1 - r_{ij}), \quad i \in I_h, \ j \in I_i^+, \ h \in H^{AM}, \ (4.2)
\]

\[
    r_{ij} + r_{ji} = 1, \quad i \in I_h, \ j \in I_i^+, \ h \in H^{AM}, \ (4.3)
\]

\[
    r_{ij} \in \{0, 1\}, \quad i \in I_h, \ j \in I_i^+, \ h \in H^{AM} \ (4.4)
\]

Constraint (4.1) ensures that all tasks start after their release time and that they are completed before their deadline. Constraint (4.4) ensures that if task \( i \) is not executed before task \( j \) it has to be the other way around, meaning all tasks will be sequenced despite not having the requirement that \( i < j \), which the Completion time variables formulation originally has. Since \( x_j \geq t_j^r \geq t_j^r - (t_i^d - (x_i + p_i)) = x_i + p_i - (t_i^d - t_j^r) \), the same value for \( M \) as in the avionic scheduling model can be used, i.e. \( M = t_i^d - t_j^r + l_{ij}^{idle} \).

### 4.1.2 CM

Introduce, for \( h \in H^{CM} \), \( s \in S_h \), \( i \in I_{s}^{sub}, j \in I_{s}^{sub}, i \neq j \), following binary variable:

\[
    r_{ijs} = \begin{cases} 
    1 & \text{if task } i \text{ is executed before task } j \text{ in subset } s, \\
    0 & \text{otherwise.} 
    \end{cases}
\]
To handle the cases where all tasks cannot be sequenced without overlaps occurring, introduce following binary variable for $h \in H^{CM}$, $s \in S_h$, $i \in I_{s}^{sub}$:

$$\beta_{is} = \begin{cases} 
1 & \text{if task } i \text{ has been sequenced successfully in subset } s, \\
0 & \text{otherwise.} 
\end{cases}$$

In the same way as with AM tasks, let the pre-processing consist of setting all $r_{ijis}$ where $t_{ij}^d < t_{ij}^r$ to 1 and corresponding $r_{jis}$ to 0. Then let the set $I_{isr}^{+}$ denote all tasks that is a possible successor of task $i$ in subset $s_r$, meaning all tasks $j$ where $r_{ijis}$ has not been given a value and $i \neq j$, $i \in I_{s}^{sub}$, $r \in R_h$, $h \in H^{CM}$. Let the following constraints sequence the tasks:

$$t_{ir}^i \leq x_i \leq t_{ir}^d - p_i, \quad i \in I_{s}^{sub}, \quad r \in R_h, \quad h \in H^{CM},$$

$$x_i + p_i \leq x_j + M(1 - r_{ijis}), \quad i \in I_{s}^{sub}, \quad j \in I_{i}^{+}, \quad r \in R_h, \quad h \in H^{CM},$$

$$\beta_{isr} \leq r_{ijis} + r_{jisr}, \quad i \in I_{s}^{sub}, \quad j \in I_{i}^{+}, \quad r \in R_h, \quad h \in H^{CM},$$

$$r_{ijis} \in \{0, 1\}, \quad i \in I_{s}^{sub}, \quad j \in I_{i}^{+}, \quad r \in R_h, \quad h \in H^{CM},$$

$$\beta_{isr} \in \{0, 1\}, \quad i \in I_{s}^{sub}, \quad r \in R_h, \quad h \in H^{CM}$$

Note that these constraints are similar to Constraint (4.1)–(4.4) with the exception that due to Constraint (4.7), it is no longer a requirement that task $i$ is either executed before task $j$ or vice versa. With the same reasoning as in Section 4.1.1 but without the idle time, $M = t_{ir}^d - t_{ir}^r$ applies on the CM tasks.

## 4.2 Assignment and positional date variables

In this section, the MIP formulation presented in Section 3.2 is adapted to the avionic scheduling problem. While this is the preferred sequencing method according to [2], it differs more from the MIP formulations in Section 2.3.1 and 2.3.2 than the Completion time variables formulation does. For example, this formulation does not result in the starting time of each task, which is a requirement for a feasible schedule in this context. From now on, the AM and CM formulations presented below will be referred to as Formulation 2.

### 4.2.1 AM

Based on the model in Section 3.2, we here assume that on each AM, there exists as many positions as tasks. The idea is to, instead of arranging the tasks relatively to each other, assign the tasks to these positions where the starting
times are yet to be decided. Let the set \( P_h \) denote all positions on AM \( h \), numbered from 1 to \( n \), with \( n \) being the number of tasks on AM \( h \), \( h \in H^{AM} \). Introduce, for \( i \in I_h, k \in P_h, h \in H^{AM} \):

\[
u_{ik} = \begin{cases} 
1 & \text{if task } i \text{ is assigned to position } k, \\
0 & \text{otherwise.}
\end{cases}
\]

For \( k \in P_h, h \in H^{AM} \), introduce following variable:

\[
\varphi_k = \text{completion time of the job assigned to position } k.
\]

To handle the idle time, it is necessary to identify which tasks are consecutive, which causes the need of a variable similar to the extra variables in [5]. Following variable is introduced for this purpose, for \( i \in I_h, j \in I_h, k \in P_h, h \in H^{AM} \):

\[
q_{ijk} = \begin{cases} 
1 & \text{if task } j \text{ is assigned to position } k \\
& \text{and task } i \text{ is assigned to position } k-1, \\
0 & \text{otherwise.}
\end{cases}
\]

The following constraints then assign the tasks to positions:

\[
\sum_{k \in P_h} u_{ik} = 1, \quad i \in I_h, h \in H^{AM}, (4.10)
\]

\[
\sum_{i \in I_h} u_{ik} = 1, \quad k \in P_h, h \in H^{AM}, (4.11)
\]

\[
\varphi_0 = 0, \quad (4.12)
\]

\[
\varphi_k \geq \sum_{i \in I_h} (p_i + t_i^r)u_{ik}, \quad k \in P_h, h \in H^{AM}, (4.13)
\]

\[
\varphi_k \leq \sum_{i \in I_h} t_i^d u_{ik}, \quad k \in P_h, h \in H^{AM}, (4.14)
\]

\[
\varphi_k \geq \varphi_{k-1} + \sum_{j \in I_h} p_j u_{jk} + \sum_{i \in I_h} \sum_{j \in I_h} t_{ij}^{idle} q_{ijk}, \quad k \in P_h, h \in H^{AM}, (4.15)
\]
4.2. Assignment and positional date variables

\[ x_i + p_i \geq \varphi_k - M_1 (1 - u_{ik}), \quad i \in \mathcal{I}_h, \ k \in \mathcal{P}_h, \ h \in \mathcal{H}^\text{AM}, \ (4.16) \]

\[ x_i + p_i \leq \varphi_k + M_2 (1 - u_{ik}), \quad i \in \mathcal{I}_h, \ k \in \mathcal{P}_h, \ h \in \mathcal{H}^\text{AM}, \ (4.17) \]

\[ \sum_{i \in \mathcal{I}_h} q_{ijk} = u_{jk}, \quad j \in \mathcal{I}_h, \ k \in \mathcal{P}_h, \ h \in \mathcal{H}^\text{AM}, \ (4.18) \]

\[ \sum_{i \in \mathcal{I}_h} q_{jik} = u_{jk} - 1, \quad j \in \mathcal{I}_h, \ k \in \mathcal{P}_h, \ h \in \mathcal{H}^\text{AM}, \ (4.19) \]

\[ u_{ik} \in \{0, 1\}, \quad i \in \mathcal{I}_h, \ k \in \mathcal{P}_h, \ h \in \mathcal{H}^\text{AM} \ (4.20) \]

Constraint (4.10)–(4.11) ensures that every task is assigned to a position and every position has a task assigned to it. Constraint (4.13) ensures that no task start before its release time, Constraint (4.14) that no task is finished after its deadline and Constraint (4.15) that there is enough time between consecutive places. In order to obtain the starting time of every task, Constraints (4.16) and (4.17) together ensures that \( x_i + p_i = \varphi_k \) if task \( i \) is assigned to position \( k \). Constraint (4.18)–(4.19) connect the \( q \)-variables to the \( u \)-variables. \( M_1 \) and \( M_2 \) will in this formulation be large enough if \( M_1 = \max_{i \in \mathcal{I}_h} \{ t^d_i \} - t^r_j - p_j \) and \( M_2 = \max_{i \in \mathcal{I}_h} \{ t^d_i \} \).

4.2.2 CM

Since there is no idle time to take in consideration for the CM tasks, there is no need for a \( q \)-variable here. However, a \( \beta \)-variable is needed to handle the case where not all tasks can be sequenced without overlaps. For this formulation, let \( \beta_{is} \) represent whether or not task \( i \) has been assigned to a position in subset \( s \). Let \( \mathcal{P}_s \) denote all positions in subset \( s \), numbered from 1 to \( n \), with \( n \) being the number of tasks in subset \( s \), \( s \in \mathcal{S}_h, \ h \in \mathcal{H}^\text{CM} \). Introduce a binary variable, for \( i \in \mathcal{I}_s^{\text{sub}}, \ k \in \mathcal{P}_s, \ s \in \mathcal{S}_h, \ h \in \mathcal{H}^\text{CM} \):

\[ u_{iks} = \begin{cases} 
1 & \text{if task } i \text{ is assigned to position } k \text{ in subset } s, \\
0 & \text{otherwise.} 
\end{cases} \]

For \( k \in \mathcal{P}_s, \ s \in \mathcal{S}_h \) and \( h \in \mathcal{H}^\text{CM} \), introduce following continuous variable:

\[ \varphi_{ks} = \text{completion time of the task assigned to position } k \text{ in subset } s \]

For \( i \in \mathcal{I}_s^{\text{sub}}, \ s \in \mathcal{S}_h, \ h \in \mathcal{H}^\text{CM} \), let:

\[ \beta_{is} = \begin{cases} 
1 & \text{if task } i \text{ has been assigned a position in subset } s, \\
0 & \text{otherwise.} 
\end{cases} \]
The sequencing requirements can then be formulated as

\[ \sum_{k \in P_{s_r}} u_{ik{s_r}} = \beta_{i{s_r}}, \quad i \in I_{s_r}, \quad r \in R_{h}, \quad h \in H_{CM}, \quad (4.21) \]

\[ \sum_{i \in I_{s_r}} u_{ik{s_r}} \leq 1, \quad k \in P_{s_r}, \quad r \in R_{h}, \quad h \in H_{CM}, \quad (4.22) \]

\[ \varphi_{0{s_r}} = 0, \quad r \in R_{h}, \quad h \in H_{CM}, \quad (4.23) \]

\[ \varphi_{k{s_r}} \geq \varphi_{k-1,s_r} + \sum_{i \in I_{s_r}} p_i u_{ik{s_r}}, \quad k \in P_{s_r}, \quad r \in R_{h}, \quad h \in H_{CM}, \quad (4.24) \]

\[ \varphi_{k{s_r}} \geq \sum_{i \in I_{s_r}} (p_i + t_{ir}) u_{ik{s_r}}, \quad k \in P_{s_r}, \quad r \in R_{h}, \quad h \in H_{CM}, \quad (4.25) \]

\[ \varphi_{k{s_r}} \leq \sum_{i \in I_{s_r}} t_{ir} u_{ik{s_r}} + M_2 (1 - \sum_{i \in I_{s_r}} u_{ik{s_r}}), \quad k \in P_{s_r}, \quad r \in R_{h}, \quad h \in H_{CM}, \quad (4.26) \]

\[ x_i + p_i \geq \varphi_{k{s_r}} - M_1 (1 - u_{ik{s_r}}), \quad i \in I_{s_r}, \quad k \in P_{s_r}, \quad r \in R_{h}, \quad h \in H_{CM}, \quad (4.27) \]

\[ x_i + p_i \leq \varphi_{k{s_r}} + M_2 (1 - u_{ik{s_r}}), \quad i \in I_{s_r}, \quad k \in P_{s_r}, \quad r \in R_{h}, \quad h \in H_{CM}, \quad (4.28) \]

\[ \varphi_{k{s_r}} \geq 0, \quad k \in P_{s_r}, \quad r \in R_{h}, \quad h \in H_{CM}, \quad (4.29) \]

\[ u_{ik{s_r}} \in \{0,1\}, \quad i \in I_{s_r}, \quad k \in P_{s_r}, \quad r \in R_{h}, \quad h \in H_{CM}, \quad (4.30) \]

\[ \beta_{i{s_r}} \in \{0,1\}, \quad i \in I_{s_r}, \quad k \in P_{s_r}, \quad r \in R_{h}, \quad h \in H_{CM}, \quad (4.31) \]

Since it is not a requirement that all tasks are sequenced, all tasks will not necessarily be assigned to a position and all positions will therefore not necessarily have a task assigned to them. This is expressed in Constraints (4.21)–(4.22). Also, allowing a position to not have a task assigned to it creates the need for the added term in Constraint (4.24), ensuring that \( \varphi_{k{s_r}} \) is not automatically set to 0 if there is no task assigned to position \( k \) in subset \( s \). The rest of the constraints above are similar to the ones in Section 4.2.1, with the value of \( M_1 \) and \( M_2 \) being the same as in Section 4.2.1.
4.3 On/off event-based formulation

In this section, the formulation that will be referred to as Formulation 3 in the rest of this thesis is presented. It is based on the formulation from Section 3.3, but since this formulation is created with a slightly different problem in mind, several alterations need to be made in order to be comparable to the avionic scheduling problem. For example, Constraint (3.23) that restricts the usage of resources have to be changed to only allow for one resource with the maximum capacity of 1, with the interpretation that only one task can be executed at once.

4.3.1 AM

Since all tasks will be sequenced without overlaps, there will never be more than one task being executed on the same event which means that every task will start at one event and end before the next event. This leads to the basic idea of this formulation becoming quite similar to the AM part of Formulation 2 presented in Section 4.2.1. With this in mind, a variable similar to the \( q \)-variable presented in [5] can be used in order to find out which tasks that are consecutive. For \( h \in H^{AM} \), let the set \( E_h \) denote all events on AM \( h \) numbered from 0 to \( n - 1 \), \( n \) being the number of tasks on AM \( h \). For \( e \in E_h, i \in I_h, h \in H^{AM} \), introduce a binary variable:

\[
    z_{ie} = \begin{cases} 
        1 & \text{if task } i \text{ starts at or still continues directly after event } e, \\
        0 & \text{otherwise.} 
    \end{cases}
\]

For \( e \in E_h, h \in H^{AM} \), introduce following continuous variable:

\[
    \sigma_e = \text{time for event } e
\]

Also, introduce following variable for \( e \in E_h, i \in I_h, j \in I_h \) and \( h \in H^{AM} \):

\[
    q_{ije} = \begin{cases} 
        1 & \text{if task } j \text{ starts at event } e \text{ and task } i \text{ starts at event } e - 1, \\
        0 & \text{otherwise.} 
    \end{cases}
\]

An extra variable \( z_{i,-1} = 0 \) is also added, \( i \in I_h, h \in H^{AM} \), in order to handle the cases where \( z_{i,e-1} \) occurs in a constraint where the case \( e = 0 \) is included. The following constraints then sequence the tasks:

\[
    \sum_{e \in E} z_{ie} \geq 1, \quad i \in I_h, h \in H^{AM}, \quad (4.32)
\]

\[
    \sigma_{e+1} \geq \sigma_e, \quad e \in E_h \setminus \{n_h - 1\}, h \in H^{AM}, \quad (4.33)
\]

\[
    \sigma_0 = 0, \quad (4.34)
\]
\[
\sigma_f \geq \sigma_e + ((z_{ie} - z_{i,e-1}) - (z_{if} - z_{i,f-1}) - 1)p_i + \sum_{i \in \mathcal{I}_h} \sum_{j \in \mathcal{I}_h} q_{ijf}^\text{idle}, \\
i \in \mathcal{I}_h, \ e \in \mathcal{E}_h, \ f \in \mathcal{E}_h, \ e < f, \ h \in \mathcal{H}^\text{AM}, \quad (4.35)
\]

\[
\sum_{e'=0}^{e-1} z_{ie'} \leq e(1 - (z_{ie} - z_{i,e-1})), \quad i \in \mathcal{I}_h, \ e \in \mathcal{E}_h, \ h \in \mathcal{H}^\text{AM}, \quad (4.36)
\]

\[
\sum_{e'=e}^{n-1} z_{ie'} \leq (n - e)(1 + (z_{ie} - z_{i,e-1})), \quad i \in \mathcal{I}_h, \ e \in \mathcal{E}_h, \ h \in \mathcal{H}^\text{AM}, \quad (4.37)
\]

\[
\sum_{i=0}^{n-1} z_{ie} \leq 1, \quad e \in \mathcal{E}_h, \ h \in \mathcal{H}^\text{AM}, \quad (4.38)
\]

\[
t^e_{i}z_{ie} \leq \sigma_e \leq (t_i^d - p_i)(z_{ie} - z_{i,e-1}) + M_1(1 - (z_{ie} - z_{i,e-1})), \quad i \in \mathcal{I}_h, \ e \in \mathcal{E}_h, \ h \in \mathcal{H}^\text{AM}, \quad (4.39)
\]

\[
x_i \geq \sigma_e - M_1(1 - (z_{ie} - z_{i,e-1})), \quad i \in \mathcal{I}_h, \ e \in \mathcal{E}_h, \ h \in \mathcal{H}^\text{AM}, \quad (4.40)
\]

\[
x_i \leq \sigma_e + M_2(1 - (z_{ie} - z_{i,e-1})), \quad i \in \mathcal{I}_h, \ e \in \mathcal{E}_h, \ h \in \mathcal{H}^\text{AM}, \quad (4.41)
\]

\[
\sum_{i \in \mathcal{I}_h} q_{ije} = z_{ie}, \quad i \in \mathcal{I}_h, \ e \in \mathcal{E}_h, \ h \in \mathcal{H}^\text{AM}, \quad (4.42)
\]

\[
\sum_{i \in \mathcal{I}_h} q_{jie} = z_{i,e-1}, \quad i \in \mathcal{I}_h, \ e \in \mathcal{E}_h, \ h \in \mathcal{H}^\text{AM}, \quad (4.43)
\]

\[
\sigma_e \geq 0, \quad e \in \mathcal{E}_h, \ h \in \mathcal{H}^\text{AM}, \quad (4.44)
\]

\[
q_{ije} \geq 0, \quad i \in \mathcal{I}_h, \ j \in \mathcal{I}_h \ e \in \mathcal{E}_h, \ h \in \mathcal{H}^\text{AM}, \quad (4.45)
\]

\[
z_{ie} \in \{0, 1\}, \quad i \in \mathcal{I}_h, \ e \in \mathcal{E}_h, \ h \in \mathcal{H}^\text{AM} \quad (4.46)
\]

Constraints (4.32)–(4.34) and (4.36)–(4.37) are practically the same as Constraints (3.17)–(3.19) and (3.21)–(3.22) in Section 3.3, and Constraint (4.35) corresponds to Constraint (3.20) but with the added idle time. Constraint (4.38) is a version of Constraint (3.23) but with only one resource with a maximum capacity of one, meaning that there can at most be one task processed on each event. Constraint (4.39) ensures that the starting time for each task occurs within its allowed time window. Constraints (4.40)–(4.41) gives values to the \( x \)-variables in a similar way as in Section 4.2.1 and Constraints (4.42)–(4.43) connects the \( q \)-variables to the \( z \)-variables. For this formulation, \( M_1 = \max_{i \in \mathcal{I}_h} \{t_i^d\} \) and \( M_2 = \max_{i \in \mathcal{I}_h} \{t_i^d - p_i\} \) is used.
4.3.2 CM

Let the events in subset $s$ be numbered from 0 to $n - 1$, and let the set of these be denoted $E_s$, where $n$ is the total amount of tasks in subset $s$, $s \in S_h$, $h \in H^{CM}$. For $e \in E_s$, $i \in I_{sub}^s$, $s \in S_h$, $h \in H^{CM}$, introduce following binary variable:

$$z_{ies} = \begin{cases} 
1 & \text{if task } i \text{ starts at or still continues directly after event } e \text{ in subset } s, \\
0 & \text{otherwise.}
\end{cases}$$

For $e \in E_s$, $s \in S_h$, $h \in H^{CM}$, introduce the continuous variable

$$\sigma_{es} = \text{time for event } e \text{ in subset } s$$

For $e \in E_s$, $s \in S_h$, $h \in H^{CM}$, also introduce a binary variable

$$\beta_{es} = \begin{cases} 
1 & \text{if there is at most one task that executes} \\
& \text{during event } e \text{ in subset } s, \\
0 & \text{otherwise.}
\end{cases}$$

Just like for the AM tasks, $z_{i,-1,s} = 0$ is added for $i \in I_{sub}^s$, $s \in S_h$, $h \in H^{CM}$. The following constraints can then be stated to sequence the tasks:

$$\sum_{e \in E_{\bar{s}r}} z_{ie\bar{s}r} \geq 1,$$  

$$i \in I_{sub}^s, \; r \in R_h, \; h \in H^{CM}, \tag{4.47}$$

$$\sigma_{e+1,\bar{s}r} \geq \sigma_{e\bar{s}r},$$  

$$e \in E_{\bar{s}r}, \; r \in R_h, \; h \in H^{CM}, \tag{4.48}$$

$$\sigma_{0\bar{s}r} = 0,$$  

$$r \in R_h, \; h \in H^{CM}, \tag{4.49}$$

$$\sigma_{f\bar{s}r} \geq \sigma_{e\bar{s}r} + ((z_{i\bar{s}r} - z_{i,e-1,\bar{s}r}) - (z_{if\bar{s}r} - z_{i,f-1,\bar{s}r}) - 1)p_i,$$  

$$i \in I_{sub}^s, \; e, f \in E_{\bar{s}r}, \; e \neq f, \; r \in R_h, \; h \in H^{CM}, \tag{4.50}$$

$$\sum_{e' = 0}^{e-1} z_{ie'e\bar{s}r} \leq e(1 - (z_{ie\bar{s}r} - z_{i,e-1,\bar{s}r})), $$

$$i \in I_{sub}^s, \; e \in E_{\bar{s}r}, \; r \in R_h, \; h \in H^{CM}, \tag{4.51}$$

$$\sum_{e' = e}^{n-1} z_{ie'e\bar{s}r} \leq (n - e)(1 + (z_{ie\bar{s}r} - z_{i,e-1,\bar{s}r})), $$

$$i \in I_{sub}^s, \; e \in E_{\bar{s}r}, \; r \in R_h, \; h \in H^{CM}, \tag{4.52}$$
\[ \sum_{i=0}^{n-1} z_{ie} - \bar{s}_r \leq n(1 - \beta_{e} - \bar{s}_r) + 1, \quad e \in \mathcal{E}_s, \ r \in \mathcal{R}_h, \ h \in \mathcal{H}^\text{CM}, \quad (4.53) \]

\[ t_{ir}^z z_{ie} - \bar{s}_r \leq s_e \leq (t_{ir}^d - p_i)(z_{ie} - z_{i,e-1}) + M_1(1 - (z_{ie} - z_{i,e-1})), \]

\[ i \in I_{\mathcal{S}_r}^{\text{sub}}, \ e \in \mathcal{E}_s, \ r \in \mathcal{R}_h, \ h \in \mathcal{H}^\text{CM}, \quad (4.54) \]

\[ x_i \geq s_{e} - M_1(1 - (z_{ie} - z_{i,e-1})), \]

\[ i \in I_{\mathcal{S}_r}^{\text{sub}}, \ e \in \mathcal{E}_s, \ r \in \mathcal{R}_h, \ h \in \mathcal{H}^\text{CM}, \quad (4.55) \]

\[ x_i \leq s_{e} + M_2(1 - (z_{ie} - z_{i,e-1})), \]

\[ i \in I_{\mathcal{S}_r}^{\text{sub}}, \ e \in \mathcal{E}_s, \ r \in \mathcal{R}_h, \ h \in \mathcal{H}^\text{CM}, \quad (4.56) \]

\[ \sigma_{e} \geq 0, \quad e \in \mathcal{E}_s, \ r \in \mathcal{R}_h, \ h \in \mathcal{H}^\text{CM}, \quad (4.57) \]

\[ z_{ie} \in \{0, 1\}, \quad i \in I_{\mathcal{S}_r}^{\text{sub}}, \ e \in \mathcal{E}_s, \ r \in \mathcal{R}_h, \ h \in \mathcal{H}^\text{CM} \quad (4.58) \]

The main difference between the AM formulation in Section 4.3.1 and this formulation is that here, no \( q \)-variables are needed since there is no idle time to take in consideration, but a \( \beta \)-variable is added just like for the other CM formulations. In this case, \( \beta_{e} \) indicates whether there is more than one task being executed at event \( e \) in subset \( s \) or not. \( M_1 \) and \( M_2 \) here has the same values as in Section 4.3.1.

### 4.4 Other formulations

In this section, it is briefly discussed why some of the MIP formulations from Chapter 3 were not adapted to the sequencing component of the scheduling tool.

#### 4.4.1 Linear ordering variables

Since this formulation is created with the objective to minimize the makespan, tasks will be placed as early as possible without additional restrictions, which is the reason for why this formulation has not been applied to this problem.

#### 4.4.2 Time index variables

The scheduling tool executes with a time precision of about \( 10^9 \), meaning that this formulation would lead to \( 10^9 \) periods. This would create far too many variables to handle, which is why this formulation was considered not suitable to apply to this specific problem.
4.4.3 Start/End event-based

Since the On/off event-based formulation outperforms the Start/end event-based one according to [6], this formulation was not implemented in the scheduling tool. Moreover, this formulation would have to been even more altered than the On/off even-based formulation, to the extent that the formulation was unrecognisable, which also affected the decision to not implement it.
Chapter 5

Results

In this chapter, the computational performances of the different formulations are presented and evaluated. In order to obtain well-founded results, the tests have been run using eight different objective functions and three different scheduling instances named Instance I, II and III, introduced in [1]. Instance I has 2 nodes and a total of about 6500 tasks, Instance II has 5 nodes and a total of about 14 000 tasks and Instance III has 7 nodes and about 20 000 tasks. The portions of fixed tasks on these instances are 65%, 54% and 53%, respectively. On all three instances, 8 different objective functions have been tested for each combination of one AM formulation and one CM formulation. All tests have been carried out on a computer with two Intel Xeon E5-2640-v3 Processors (8 cores, 2.6 GHz) and a 64 GB RAM. The implementation of the scheduling tool have been made in Python Version 3.6.0 and the MIP models have been solved by Gurobi Optimizer Version 7.5 beta. If a feasible schedule has not been found within 24 hours, the scheduling tool has been interrupted.

The formulations presented in Section 2.3, 4.1, 4.2 and 4.3 will here be referred to as Formulation 0, 1, 2 and 3, respectively. The notation X/Y indicate that the AM tasks are sequenced using Formulation X, and the CM tasks are sequenced using Formulation Y. For a complete account of the results, see Appendix A.

5.1 Main computational result

This section illustrates the main computational result of Formulation 0, 1, 2 and 3 in Figure 5.1, 5.2 and 5.3. This is done by, for each point in time, illustrating for how many of the objective functions a feasible schedule have been obtained.
Table 5.1: Notations used for the formulations presented in Chapter 2 and 3.

<table>
<thead>
<tr>
<th>Formulation</th>
<th>Presented in Section</th>
<th>Notation used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current formulation implemented in the tool</td>
<td>2.3</td>
<td>0</td>
</tr>
<tr>
<td>Completion time variables</td>
<td>4.1</td>
<td>1</td>
</tr>
<tr>
<td>Assignment and positional date variables</td>
<td>4.2</td>
<td>2</td>
</tr>
<tr>
<td>On/off event-based formulation</td>
<td>4.3</td>
<td>3</td>
</tr>
</tbody>
</table>

for the combinations 0/0, 1/1, 2/2 and 3/3, respectively. Thereafter, Table 5.2 shows which combinations of AM and CM formulations that resulted in the single fastest completion time for the different instances.

For Instance I, a feasible schedule was obtained within 5 minutes for all objective functions when using either combination 0/0 or 1/1. The same was achieved within 10 and 40 minutes when using combination 2/2 and 3/3, respectively, see Figure 5.1.

For Instance II, a feasible schedule was obtained within 24 hours for all objective functions when using combination 1/1. Using combination 3/3 did not result in any feasible schedules, while 0/0 and 2/2 both performed worse than 1/1 but better than 3/3, see Figure 5.2.

For Instance III, neither combination 2/2 or 3/3 could produce any feasible schedule within 24 hours. When using combination 1/1, a schedule was obtained within 40 minutes for all 8 objective functions. When using combination 0/0, 6 of the objective functions led to a feasible schedule within 120 minutes, see Figure 5.3.

Regarding which combination that, for its best choice of objective function, can produce a schedule the fastest, 3/1 is the best on Instance I, 2/1 on Instance II and 1/1 is the fastest on Instance III, see Table 5.2.
5.1. Main computational result

Figure 5.1: For how many of the objective functions a feasible schedule has been obtained on Instance I.

Figure 5.2: For how many of the objective functions a feasible schedule has been obtained on Instance II. Between 90 minutes and 24 hours, no more schedule was obtained.
5.2 Comparison of all AM/CM combinations

In this section, we want to compare the computational performance for all combinations of formulations for AMs and CMs. For this purpose, we look at the median completion times of all objective functions since only half of them need to lead to a feasible schedule in order to obtain a median time.

The three figures in this section show the median time for every combination of formulations on the three instances, respectively. On Instance I, the median times for CM formulation 0 combined with AM formulations 0-3, as well as CM
formulation 1 together with AM formulations 0-3, are all around 160 seconds. Median time for CM formulation 2 is just over 210 seconds regardless of which AM formulation it is combined with, and when using CM formulation 3 all median times become around 300 seconds.

On Instance II, the three best median times are obtained using CM formulation 1 combined with AM formulation 0-2. The next three best times are obtained using CM formulation 0 together with AM formulation 0-2. No combination using AM formulation 3 result in an obtained median time, meaning that a feasible schedule was found for less than half of the objective functions. Using CM formulation 3 only results in a median time when combining it with AM formulation 1. However, this median time is almost 150 minutes, which is considerably higher than the other obtained median times, that are all between 10 and 70 minutes.

On Instance III, no combination using AM or CM formulation 3 results in an obtained median time. The four best median times were obtained by Formulation 1/1 followed by 0/1, 1/0 and 0/0.
Figure 5.5: Median runtimes on Instance II

* indicates the cases where no median time could be obtained (i.e. a schedule was found within 24 hours for less than half of the objective functions).

Figure 5.6: Median runtimes on Instance III

* indicates the cases where no median time could be obtained (i.e. a schedule was found within 24 hours for less than half of the objective functions).
To summarize the results, combination 0/0 and 1/1 are the fastest when only comparing the combinations 0/0, 1/1, 2/2 and 3/3 on all three instances. When comparing all AM/CM formulations, 0/0, 1/0, 0/1 and 1/1 have the smallest median completion times when looking at the overall performances on the different instances.
Chapter 6
Discussion

From the results in Section 5.1, it is clear that when using the same formulation for both the AMs and the CMs, Formulation 1 and Formulation 0 are as good as equal on the smallest instance. For larger instances, however, Formulation 1 performs considerably better than Formulation 0. Formulation 2 performs better than Formulation 3, but they both perform clearly worse than the first two formulations on all three instances.

When comparing all different combinations of AM and CM formulations, there is a pattern indicating that the CM formulation makes a bigger impact on the computational runtime than the AM formulation does. For example, the single best computational time for all instances are achieved by using Formulation 1 for the CM tasks but different formulations for the AM tasks. However, the choice of AM formulation seems to matter since Formulation 3 obtains no schedule on larger instances regardless of which CM formulation is used.

This result is not in keeping with the study made in [2], that concluded that the Completion time variable formulation does not perform well in practice while the Assignment and positional date variables formulation performs better. An explanation for this could be that, even though the instances contain up to 20,000 tasks, there is generally not a particularly large amount of tasks assigned to each section on the CMs. Therefore, it is not necessarily better with a formulation that is efficient on many tasks.

One possible reason why Formulation 2 and 3 are considerably worse than Formulation 1, is that there is a greater difference between these and Formulation 0, which means that major adaptations had to be made and the efficiency of the original MIP formulations can have been lost by doing this. Also, unlike Formulation 0 and 1, there was no pre-processing done for the last two formulations which also may have impacted their computational times negatively.
In further studies, it could be investigated whether some sort of pre-processing could improve Formulation 2 and 3. Since this study only covers sequencing in a very specific context, another subject for future research would be to compare these formulations in other contexts as well.

To conclude, it can be stated that the choice of mathematical model for the sequencing component in the scheduling tool makes a considerable impact on the computational times, especially the formulation used for CM tasks. By replacing the sequencing component with Formulation 1, it is possible to improve the computational time of the scheduling tool significantly.
Chapter 7

Bibliography


Appendix A

Computational results

On the following pages, all computational times are presented, meaning a total of 16 different combinations of AM and CM formulations. The notation 3/0 indicates that the AM tasks are sequenced using Formulation 3 and the CM tasks are sequenced using Formulation 0. "x" indicates that the time limit of 24 hours was reached without a feasible schedule being found. A table that, for each instance, shows the median times for each combination of formulations is also presented, with "x" indicating that no mean time was obtained (i.e. less than half of the objective functions were completed within 24 hours).
### Table A.1: Computational times (mm:ss)

<table>
<thead>
<tr>
<th>Data instance</th>
<th>Objective function</th>
<th>Formulations used (AM/CM)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0/0</td>
</tr>
<tr>
<td>Instance I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>2:41</td>
</tr>
</tbody>
</table>

### Table A.2: Computational times (mm:ss)

<table>
<thead>
<tr>
<th>Data instance</th>
<th>Objective function</th>
<th>Formulations used (AM/CM)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2/0</td>
</tr>
<tr>
<td>Instance I</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table A.3: Computational times (hh:mm:ss / mm:ss)

<table>
<thead>
<tr>
<th>Data instance</th>
<th>Objective function</th>
<th>Formulations used (AM/CM)</th>
<th>0/0</th>
<th>0/1</th>
<th>0/2</th>
<th>0/3</th>
<th>1/0</th>
<th>1/1</th>
<th>1/2</th>
<th>1/3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table A.4: Computational times (hh:mm:ss / mm:ss)

<table>
<thead>
<tr>
<th>Data instance</th>
<th>Objective function</th>
<th>Formulations used (AM/CM)</th>
<th>2/0</th>
<th>2/1</th>
<th>2/2</th>
<th>2/3</th>
<th>3/0</th>
<th>3/1</th>
<th>3/2</th>
<th>3/3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table A.5: Computational times (hh:mm:ss / mm:ss)

<table>
<thead>
<tr>
<th>Data instance</th>
<th>Objective function</th>
<th>0/0</th>
<th>0/1</th>
<th>0/2</th>
<th>0/3</th>
<th>1/0</th>
<th>1/1</th>
<th>1/2</th>
<th>1/3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance III</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>33:49</td>
<td>x</td>
<td>x</td>
<td>29:22</td>
<td>28:50</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>02:04:34</td>
<td>01:45:06</td>
<td>x</td>
<td>x</td>
<td>13:07</td>
<td>10:17</td>
<td>07:52:14</td>
<td>x</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>17:01</td>
<td>12:36</td>
<td>01:34:37</td>
<td>x</td>
<td>23:18</td>
<td>13:02</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>x</td>
<td>01:03:54</td>
<td>01:41:25</td>
<td>05:58:05</td>
<td>x</td>
<td>31:02</td>
<td>01:20:45</td>
<td>x</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>x</td>
<td>01:03:05</td>
<td>01:35:30</td>
<td>x</td>
<td>30:30</td>
<td>02:57:16</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>35:28</td>
<td>21:40</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>21:13</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>38:58</td>
<td>18:38</td>
<td>01:21:50</td>
<td>01:05:15</td>
<td>22:56</td>
<td>19:21</td>
<td>02:03:30</td>
<td>x</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>23:57</td>
<td>19:49</td>
<td>01:08:12</td>
<td>x</td>
<td>44:17</td>
<td>20:22</td>
<td>02:03:49</td>
<td>x</td>
</tr>
</tbody>
</table>

### Table A.6: Computational times (hh:mm:ss)

<table>
<thead>
<tr>
<th>Data instance</th>
<th>Objective function</th>
<th>2/0</th>
<th>2/1</th>
<th>2/2</th>
<th>2/3</th>
<th>3/0</th>
<th>3/1</th>
<th>3/2</th>
<th>3/3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance III</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>
### Table A.7: Median completion times

<table>
<thead>
<tr>
<th>AM/CM</th>
<th>Instance I</th>
<th>Instance II</th>
<th>Instance III</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/0</td>
<td>0:02:42</td>
<td>0:19:55</td>
<td>0:37:13</td>
</tr>
<tr>
<td>0/1</td>
<td>0:02:39</td>
<td>0:12:55</td>
<td>0:24:07</td>
</tr>
<tr>
<td>0/2</td>
<td>0:03:34</td>
<td>0:36:13</td>
<td>1:38:27</td>
</tr>
<tr>
<td>0/3</td>
<td>0:04:59</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>1/0</td>
<td>0:02:41</td>
<td>0:19:13</td>
<td>0:36:50</td>
</tr>
<tr>
<td>1/1</td>
<td>0:02:39</td>
<td>0:11:48</td>
<td>0:20:48</td>
</tr>
<tr>
<td>1/2</td>
<td>0:03:33</td>
<td>0:53:15</td>
<td>5:24:45</td>
</tr>
<tr>
<td>1/3</td>
<td>0:05:00</td>
<td>2:21:22</td>
<td>x</td>
</tr>
<tr>
<td>2/0</td>
<td>0:02:40</td>
<td>0:16:01</td>
<td>x</td>
</tr>
<tr>
<td>2/1</td>
<td>0:02:40</td>
<td>0:10:02</td>
<td>x</td>
</tr>
<tr>
<td>2/2</td>
<td>0:03:34</td>
<td>1:06:42</td>
<td>x</td>
</tr>
<tr>
<td>2/3</td>
<td>0:05:01</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>3/0</td>
<td>0:02:40</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>3/1</td>
<td>0:02:40</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>3/2</td>
<td>0:03:35</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>3/3</td>
<td>0:04:58</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
Copyright

The publishers will keep this document online on the Internet – or its possible replacement – from the date of publication barring exceptional circumstances. The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/her own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: [http://www.ep.liu.se/](http://www.ep.liu.se/)

Upphovsrätt

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den umfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida [http://www.ep.liu.se/](http://www.ep.liu.se/)

© 2017, Jessika Boberg