

Leibniz Universität Hannover
Wirtschaftswissenschaftliche Fakultät
Institut für Produktionswirtschaft
Prof. Dr. Stefan Helber

<Seminararbeit/Hausarbeit>
im Rahmen des <Seminars Operations Management & Research/der Veranstaltung
xy> im <Semester>
(Veranstaltungs-Nr. <Belegnummer>)

Thema Nr. <Themanummer>
<Thema>

<Vor- und Nachname>

<Straße>

<PLZ Ort>

Matr.-Nr.: <Matrikelnummer>

Abgabedatum: <Abgabedatum>

<Titel der Arbeit, der ggf. auch etwas länger sein wird und sich darum über mehrere Zeilen erstrecken könnte>

Masterarbeit

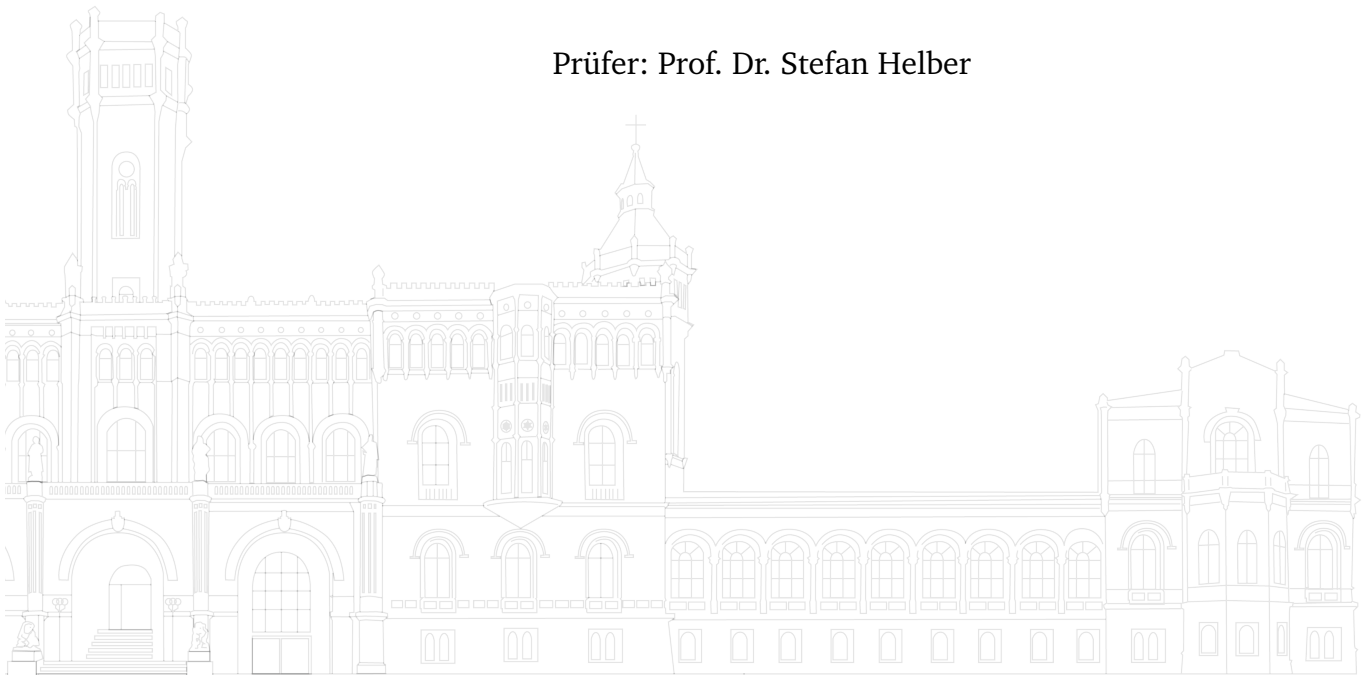
zur Erlangung des akademischen Grades „Master of Science (M.Sc.)“ im Studiengang
Wirtschaftswissenschaft der Wirtschaftswissenschaftlichen Fakultät
der Leibniz Universität Hannover

vorgelegt von

<Vorname Nachname>

geboren am <Datum> in <Ort>

Prüfer: Prof. Dr. Stefan Helber



<Ort>, <Abgabedatum>

Todo list

Figure: Hier fehlt die Subfigure links	3
Figure: Hier fehlt die Subfigure rechts	3
Figure: Hier fehlt eine Abbildung	4
Figure: Und hier fehlt noch eine Abbildung	4
Hier steht eine Aufgabe, die noch gemacht werden muss.	8
z. B.: Hier muss noch eine Quelle gesucht werden.	8

Inhaltsverzeichnis

Abbildungsverzeichnis	ii
Tabellenverzeichnis	ii
Algorithmusverzeichnis	ii
Quellcodeverzeichnis	ii
Abkürzungsverzeichnis	ii
Symbolverzeichnis	iii
1 Hinweise zur Einleitung	1
2 Beispiele	1
2.1 Beispiele zum Fließtext	1
2.2 Beispiele zu Abbildungen	2
2.3 Beispiele zu Tabellen	4
2.4 Beispiele zu mathematischen Formeln, Notation und Modellannahmen .	5
2.5 Beispiel zu Pseudocode	8
2.6 Beispiel zu To-dos	8
2.7 Beispiele zu Farben	9
2.8 Beispiele zu TikZ	9
3 Ein neues Kapitel	10
A Überschrift für den Inhalt des Anhangs	11
B Python-Code Beispiele	12
C Code-Beispiele für weitere Programmiersprachen	14

Abbildungsverzeichnis

1	Logo des Instituts für Produktionswirtschaft	2
2	Komponenten einer dynamisch-induktive Ladeinfrastruktur	3
3	Zwei Bilder mit Subfigure nebeneinander	3
4	Hier steht die Unterschrift für die linke Abbildung	4
5	Hier steht die Unterschrift für die rechte Abbildung	4
6	Farben in Abbildungen	9
7	Dies ist ein Beispiel für eine Abbildung mit TikZ.	10
8	Dies ist ein weiteres Beispiel für eine Abbildung mit TikZ.	10

Tabellenverzeichnis

1	Eine kleine Beispieltabelle	4
2	Eine größere Beispieltabelle	5
3	Notation zum Modell der Ablaufplanung	7

Algorithmusverzeichnis

1	Nearest Neighbor-Heuristik für das TSP - Hauptroutine	8
---	---	---

Quellcodeverzeichnis

1	Hier kann eine Überschrift eingefügt werden, die den Code beschreibt .	12
2	Auch hier ist eine Überschrift möglich	13
3	Dies ist ein Beispiel mit GAMS	14
4	Dies ein Beispiel für C++	15

Abkürzungsverzeichnis

LP	Lineare Programmierung (<i>engl. Linear Programming</i>)
MIP	gemischt-ganzzahlige Programmierung (<i>engl. Mixed Integer Programming</i>)
BD	Benders Dekomposition
TSP	Problem des Handlungsreisenden (<i>engl. Traveling Salesman Problem</i>)

Symbolverzeichnis

d_{ij}	Parameter, welcher die Entfernung von Ort i zu Ort j beschreibt
$i \in \mathcal{I}$	Kundenstandort i in der Menge \mathcal{I}
Y_{im}	Binäre Variable mit Wert 1, wenn Ort i in der Tour m enthalten ist

1 Hinweise zur Einleitung

In der Einleitung sollte das Problem motiviert und begründet werden. Der Umfang des Problems sollte verständlich gemacht werden. Hierfür sollte ein Fließtext verwendet werden. An geeigneten Stellen ist mit Quellen zu arbeiten. Im Text wird die Kurzzitierweise genutzt, wie am Ende dieses Satzes.¹

Zusätzlich soll bei Abschlussarbeiten eine oder mehrere Forschungsfrage(n) definiert werden. Diese können im Text hervorgehoben werden, z. B. durch

Hier könnte eine Forschungsfrage stehen. Diese Art der Darstellung eignet sich insbesondere, wenn eine Forschungsfrage präsentiert wird.

oder durch eine Aufzählung

- Hier könnte eine Forschungsfrage stehen.
- Hier könnte noch eine Forschungsfrage stehen. Diese Darstellung eignet sich insbesondere bei zwei Forschungsfragen.

2 Beispiele

Hier steht in der Arbeit kein Text, da Unterpunkte folgen!

2.1 Beispiele zum Fließtext

Der Großteil der schriftlichen Arbeit besteht aus Fließtext. Um Aussagen hervorzuheben, kann es in **Einzelfällen** sinnvoll sein, Teile **fett** oder *kursiv* zu schreiben. Auch das Unterstreichen von Wörtern ist möglich.

An geeigneten Stellen sollten Absätze verwendet werden. Diese lassen sich erzeugen, indem im Quelltext eine leere Zeile eingefügt wird (wie vor diesem Satz). Durch das Paket `usepackage[skip=12pt, indent=0pt]parskip` wurde in `preamble.tex` für diese Vorlage festgelegt, dass nach jedem Absatz ein Abstand von 12pt eingefügt wird und der Beginn einer jeden Zeile nicht eingerückt ist. Wenn keine leere Zeile nach einem Absatz gewünscht ist, kann dies dort geändert werden. In jedem Fall ist es sinnvoll, eine gleichbleibende Struktur für die gesamte Arbeit zu wählen.

Wie bei der Einleitung kann mit Quellen in Fußnoten arbeiten, z. B. so.² Auch Inter-

¹Vgl. `beispielname2024beispieltitel`, S. 57.

²Vgl. `beispielname2024beispieltitel`, S. 1.



Abbildung 1: Logo des Instituts für Produktionswirtschaft

netseiten lassen sich auf diese Art zitieren.³ Manchmal gibt es mehrere Quellen zu einer Aussage. Dies können in eine gemeinsame Fußnote aufgenommen werden.⁴ Ggf. kann es vorkommen, dass eine Quelle direkt im Text genannt wird, wie z. B. bei dem Satz: Das Modell nach **sudbeck2024using** zeigt . . .⁵ Eine zusätzliche Fußnote kann am Ende des Satzes hinzugefügt werden, um die Seiten anzugeben.

In Fußnoten können auch Zusatzinformationen oder Anmerkungen untergebracht werden, die beim Lesen des Textes nicht notwendigerweise erforderlich sind.⁶ Für alle Fußnoten gilt, dass diese durch Latex automatisch auf der passenden Seite angezeigt werden.

Anführungszeichen müssen speziell gesetzt werden, wie dieses Beispiel mit "normalen" Anführungszeichen zeigt. Dieses geschieht mit `glqq` und `grqq`, wie in diesem Beispiel: „und“. Wenn hinter dem zweiten Anführungszeichen ein weiteres Wort folgt, sodass ein Leerzeichen benötigt wird, muss eine geschweifte Klammer eingefügt werden: Dies ist „ein“ Beispiel. Ansonsten fehlt das Leerzeichen und das folgende Wort steht direkt am zweiten Anführungszeichen.

2.2 Beispiele zu Abbildungen

Eine Abbildung wird wie folgt eingefügt:

In diesem Fall soll die Grafik bei Breite von 20% der Textbreite haben. Es können PDFs, PNGs oder JPEGs eingefügt werden. Mit `caption` wird eine Abbildungsunterschrift eingefügt, welche automatisch ins Abbildungsverzeichnis aufgenommen wird. Um den Arbeitsbereich übersichtlich zu halten, ist es sinnvoll, alle Abbildungen in einem Ordner (in diesem Beispiel `Abbildungen`) zu sortieren. Über `label` wird ein Label definiert, sodass auf die Abbildung Bezug genommen werden kann. Dies geschieht dann mit `autoref`: `Abbildung 1`.

Latex ordnet Abbildungen (und auch Tabellen) automatisch auf der Seite an. Deshalb erscheint eine Grafik nicht zwingend an der Position, in der sie im Quelltext aufgeführt

³**insitutprod**, <https://www.prod.uni-hannover.de/de/lehre/lehrveranstaltungen>

⁴Vgl. **beispielname2024beispieltitel**, S. 56-58 und **beispielname2024beispieltitel2**, S. 7-17.

⁵Vgl. **sudbeck2024using**, S. 12-13.

⁶Beispielsweise bietet es sich an, bei einer numerischen Untersuchung die genutzte Hardware in einer Fußnote anzugeben.

wurde. Abbildung 1 ist ein gutes Beispiel dafür. Soll eine Grafik an einer bestimmten Position platziert werden, so kann dies mit einem H in eckigen Klammern erzwungen werden, wie bei der Abbildung 2.

Wird eine Abbildung übernommen, muss die Quelle kenntlich gemacht werden. Damit die Quelle im Abbildungsverzeichnis nicht mit aufgenommen wird und im Dokument einheitlich aufgeführt werden, wurde in `preamble.tex` der Befehl `captionsource` definiert. Hiermit kann, wie in Abbildung 2 gezeigt, die Quelle angegeben werden.

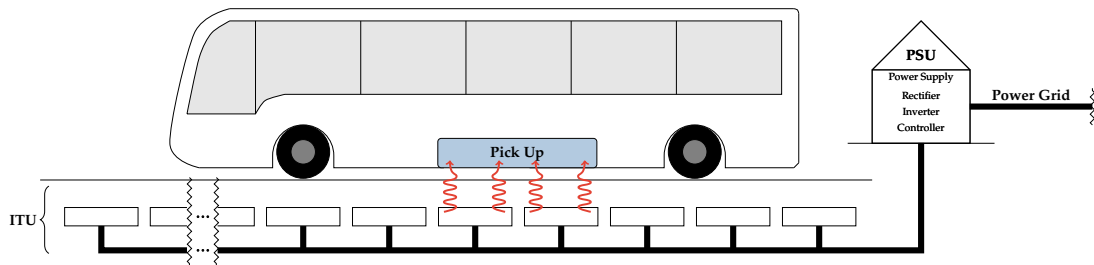
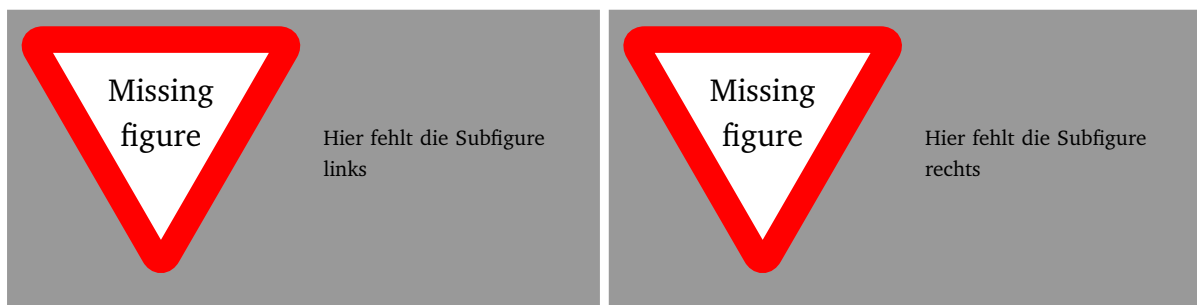


Abbildung 2: Komponenten einer dynamisch-induktiven Ladeinfrastruktur
Quelle: broihan2022designing, S. 2

Ggf. kann es sinnvoll sein, zwei Abbildungen nebeneinander zu platzieren. Es muss aber sichergestellt sein, dass die Abbildungen lesbar bleiben und insbesondere die Schrift hierdurch nicht zu klein wird. Wenn die beiden Abbildungen thematisch eng zusammengehören, eignet sich hierfür eine `subfigure`, wie in Abbildung 3 dargestellt. Diese setzt sich aus Abbildung 3a und Abbildung 3b. Sollen zwei unabhängige Abbildungen nebeneinander platziert werden, eignet sich die Erstellung einer `minipage`. Hierbei können dann zwei (oder auch mehrere) Elemente nebeneinander platziert werden, wie bei Abbildung 4 und Abbildung 5.



(a) Unterschrift Bild Nr. 1

(b) Unterschrift Bild Nr. 2

Abbildung 3: Zwei Bilder mit Subfigure nebeneinander

Kategorie	Wert 1	Wert 2	Wert 3
A	10	20	30
B	15	25	35
C	20	30	40
D	25	35	45

Tabelle 1: Eine kleine Beispieltabelle

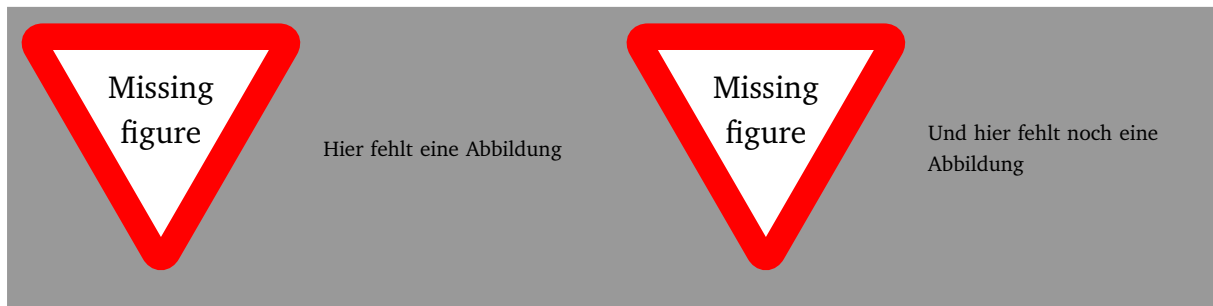


Abbildung 4: Hier steht die Unterschrift für die linke Abbildung

Abbildung 5: Hier steht die Unterschrift für die rechte Abbildung

2.3 Beispiele zu Tabellen

Tabellen zu erstellen ist mit Latex leider etwas unübersichtlich, insbesondere wenn diese größer sind. Es gibt allerdings Seiten, mit denen die Tabellen leicht erzeugt und anschließend ins Dokument eingefügt werden können. Ein Beispiel für eine kleine Tabelle ist in Tabelle 1 gegeben. Mit `lrrr` wird die Ausrichtung der einzelnen Spalten angegeben. Wie bei Abbildungen setzt Latex die Position der Tabellen automatisch, wenn diese nicht festgelegt wurde

Tabellen können natürlich auch deutlich komplexer gestaltet werden. Beispielsweise ist es möglich, einzelne Zellen miteinander zu verbinden, wie in Tabelle 2 gezeigt. Auch bei Tabellen kann der zuvor definierte Befehl `captionsource` verwendet werden.

Um Zahlen mit Einheiten sinnvoll darzustellen, kann das Paket `numprint` verwendet werden. Mit `1,8 h` wird beispielsweise die Einheit `h` für Stunde in einem festgelegten Abstand mit der Zahl verbunden. Zusätzlich kann einmalig in `preamble.tex` eingestellt werden, welche Trennzeichen für Tausender und Dezimalzahlen verwendet werden soll.

		# Instanzen	Modell 1			Modell 2		
			\bar{t}	\tilde{t}	$\overline{\text{gap}}$	\bar{t}	\tilde{t}	$\overline{\text{gap}}$
klein	A	20	3,2 s	4,2 s	0 %	10,8 s	11,9 s	0 %
	B	20	7,1 s	6,9 s	0 %	35,2 s	40,5 s	0,6 %
groß	A	20	1,2 h	1,4 h	0,7 %	2 h	2 h	5,1 %
	B	20	1,8 h	2 h	7,3 %	2 h	2 h	10,8 %

Tabelle 2: Eine größere Beispieltabelle
Quelle: Hier könnte nun eine Quelle stehen.

2.4 Beispiele zu mathematischen Formeln, Notation und Modellannahmen

Wie Zahlen mit Einheiten geschrieben werden können, wurde in Unterabschnitt 2.3 gezeigt.⁷

Um mathematische Formeln im Text darzustellen, wird das `$`-Zeichen verwendet. So kann zum Beispiel die Variable x_i einfach im Text genannt werden. Auch ein Variable mit zwei Indizes ist möglich, hier müssen allerdings geschweifte Klammern verwendet werden y_{ij} . Analog dazu kann man Variable mit hochgestelltem Index haben: c_i^g .

Längere Formeln sollten nicht im Fließtext stehen, sondern abgesetzt werden, wie bei

$$\rho = \frac{\lambda}{\mu}. \quad (1)$$

Vergibt man bei einer Formel ein `label`, so kann man im späteren Text darauf Bezug nehmen, z. B.

$$E[W_q] \approx \frac{c_a^2 + c_s^2}{2} \cdot \frac{\rho}{1 - \rho} \cdot \frac{1}{\mu}. \quad (2)$$

Nun lässt sich im Text Bezug darauf nehmen. Auch hier kann `autoref` verwendet werden, dies sieht dann so aus: Gleichung 2. Hierbei werden die runden Klammern jedoch nicht mit ausgegeben. Daher wurde ein eigener Befehl in `preamble.tex` erstellt. Mit `myeqref` wird die Gleichung nun mit runden Klammern dargestellt: Gleichung (2).

Einige mathematische Zeichen können direkt von der Tastatur verwendet werden, wie z. B. das Plus: `+`. Andere, wie beispielsweise beim Multiplikationszeichen (`\cdot`), haben

⁷Hier zeigt sich, dass man auch ein Kapitel mit einem `label` versehen kann und darauf verwiesen werden kann.

spezielle Befehle. Auch griechische Buchstaben lassen sich einfach darstellen, wie z. B. α .

Sind Gleichungen zu lang für eine Zeile, lassen sich diese mit über mehrere Zeilen darstellen, wie in Gleichung (3).

$$\begin{aligned} E[F(q)] &= \sum_{d=0}^{\infty} \max(d - q, 0) \cdot \text{Prob}[D = d] \\ &= \sum_{d=q}^{\infty} (d - q) \cdot \text{Prob}[D = d] \end{aligned} \tag{3}$$

Auch mathematische Optimierungsmodelle lassen sich mithilfe dieser Art darstellen. Bevor ein Modell eingeführt wird, muss jedoch die Notation ausführlich eingeführt werden (Dies bedeutet, dass die Notation **nicht** nur in einer Tabelle eingeführt wird!).

Notation lässt sich im Fließtext beschreiben, z. B.: Im Modell der Ablaufplanung bezeichnet $j \in \mathcal{J} = \{1, \dots, J\}$ die Menge der Jobs und $r \in \mathcal{R} = \{1, \dots, R\}$ die Menge der Ressourcen. Weiterhin stellt $t, \tau \in \mathcal{T} = \{1, \dots, T\}$ die Perioden da mit T als letzte Periode. Für jeden Job j gibt es bestimmte durchzuführende Prozessschritte, welche durch die Menge $s \in \mathcal{S}_j = \{1, \dots, l_{s_j}\}$ beschrieben werden. Der letzten Prozessschritt von Job j ist definiert als l_{s_j} . Die Durchführung des Prozessschritts s von einem Job j erfordert zu jedem Zeitpunkt $a_{j sr}$ Kapazitätseinheiten der Ressource r

Alternativ ist es auch möglich, die Notation als Aufzählung (in jeweils vollständigen Sätzen) einzuführen, z. B.:

- Im Modell der Ablaufplanung bezeichnet $j \in \mathcal{J} = \{1, \dots, J\}$ die Menge der Jobs und $r \in \mathcal{R} = \{1, \dots, R\}$ die Menge der Ressourcen.
- Die Perioden werden mit der Menge $t, \tau \in \mathcal{T} = \{1, \dots, T\}$ beschrieben mit T als letzte Periode.
- Für jeden Job j gibt es bestimmte durchzuführende Prozessschritte, welche durch die Menge $s \in \mathcal{S}_j = \{1, \dots, l_{s_j}\}$ beschrieben werden. Hierbei bezeichnet l_{s_j} den letzten Schritt von Job j .
- Die Durchführung des Prozessschritts s von einem Job j erfordert zu jedem Zeitpunkt $a_{j sr}$ Kapazitätseinheiten der Ressource r .
- . . .

Tabelle 3: Notation zum Modell der Ablaufplanung

Indizes und Mengen	
$j \in \mathcal{J} = \{1, \dots, J\}$	Jobs
$r \in \mathcal{R} = \{1, \dots, R\}$	Ressourcen
$s \in \mathcal{S}_j = \{1, \dots, l_{s_j}\}$	Schritte des Jobs j
$t, \tau \in \mathcal{T} = \{1, \dots, T\}$	Perioden
Parameter	
a_{jst}	benötigte Kapazitätseinheiten der Ressource r für Schritt s des Jobs j
b_{rt}	Kapazität der Ressource r in Periode t
...	
Entscheidungsvariablen	
$X_{jst}^{tP} \in \{0, 1\}$	binäre Variable mit Wert 1, wenn in Periode t der Schritt s des Jobs j beendet wird und 0 sonst
...	

Bitte denken Sie daran, dass auch die **Entscheidungsvariablen** zur Notation gehören und **erläutert** werden müssen!

Zusätzlich ist es bei Abschlussarbeiten sinnvoll, die Notation in einer Tabelle übersichtlich zusammenzufassen (vgl. Tabelle 3).

Nach der Einführung der Notation kann das Modell eingeführt werden.

$$\min Z = c \cdot \sum_{j \in \mathcal{J}} TF_{j, l_{s_j}} \quad (4)$$

unter Beachtung der Restriktionen

$$\sum_{t=1}^T X_{jst} = 1 \quad \forall j \in \mathcal{J}, s \in \mathcal{S}_j \quad (5)$$

$$TF_{js} \geq TF_{j, s-1} + d_{js} \quad \forall j \in \mathcal{J}, s \in \{2, \dots, l_{s_j}\} \quad (6)$$

$$\sum_{t=1}^T t \cdot X_{jst} = TF_{js} \quad \forall j \in \mathcal{J}, s \in \mathcal{S}_j \quad (7)$$

$$\sum_{j \in \mathcal{J}} \sum_{s \in \mathcal{S}_j} \sum_{\tau=t}^{t+d_{js}-1} a_{jst} \cdot X_{jst} \leq b_{rt} \quad \forall r \in \mathcal{R}, t \in \mathcal{T} \quad (8)$$

Auf die einzelnen Restriktionen kann nun Bezug genommen werden. Da es sich hierbei um Restriktionen handelt, sollte das Wort „Restriktion“ genutzt werden. Damit die Referenzierung richtig funktioniert, gibt es auch hierfür in `preamble.tex` den selbst

definierten Befehl `myrestref`. Somit können wir beschreiben, dass durch Restriktion (5) jeder Schritt in einem Job in genau einer Periode beendet wird und Gleichung (8) die Zielfunktion darstellt, in welcher

2.5 Beispiel zu Pseudocode

Bei Algorithmen ist es sinnvoll, diese als Pseudocode in der Arbeit zu veranschaulichen. Der eigentliche Code wird im Anhang der Arbeit aufgeführt (vgl. Anhang B). Für Pseudocode gibt es eigene Pakete in Latex (`algorithm` und `algorithmic`). Ein Beispiel für einen Pseudocode ist in Algorithmus 1 gegeben.

Algorithmus 1: Nearest Neighbor-Heuristik für das TSP - Hauptroutine

```

1  while Unvisited  $\neq \emptyset$  do
2     $dist_{min} = \infty$ 
3     $i = NextTown$ 
4    for  $j \in Unvisited$  do
5      if  $d(i, j) < dist_{min}$  then
6         $dist_{min} = d(i, j)$ 
7         $NextTown = j$ 
8      end if
9    end for
10    $dist_{tot} = dist_{tot} + dist_{min}$ 
11    $Unvisited = Unvisited - NextTown$ 
12    $Tour = Tour + NextTown$ 
13 end while
14  $dist_{tot} = dist_{tot} + d(j, s)$ 

```

2.6 Beispiel zu To-dos

Mit dem Paket `todonotes` lassen sich in der Arbeit Kommentare/To-dos einfügen. Diese können entweder im Text stehen:

Hier steht eine Aufgabe, die noch gemacht werden muss.

oder auch an den Seitenrand geschrieben werden.

Zu Beginn der Arbeit kann auch eine Übersicht mit allen To-dos ausgegeben werden (siehe `listoftodos` in `main.tex`). Diese Übersicht sollte natürlich **vor Abgabe der Arbeit auskommentiert** werden. Auch Platzhalter für Abbildungen, die mit `missing-figure` erstellt wurden (vgl. Abbildung 4), werden in die Übersicht mit aufgenommen.

z. B.: Hier muss noch eine Quelle gesucht werden.

2.7 Beispiele zu Farben

Farben können eingesetzt werden, um Aspekte gezielt zu betonen oder um Abbildungen gut erkennbar zu machen. Gleichzeitig sollten Farben in einer wissenschaftlichen Arbeit **sparsam** eingesetzt werden. Mit dem Paket `xcolor` und der Option `dvipsnames` (siehe `preamble.tex`) stehen 68 vordefinierte Farben zur Verfügung. Alternativ können in `preamble.tex` auch eigene Farben definieren werden, wie dies in dieser Vorlage mit den Farben der Leibniz Universität gemacht wurde. Es empfiehlt sich, im Rahmen der Arbeit einige wenige, zueinander passende Farben auszuwählen und diese in der Arbeit durchgängig einzusetzen.

Mit einer Farbe können beispielsweise **Textabschnitte eingefärbt** werden. Auch lässt sich ein Text **farbig unterlegen**. Auch Aufzählungen lassen sich einfärben:

- Erster Punkt
- Zweiter Punkt

Insgesamt stellt eine farbliche Markierung von Text in einer wissenschaftlichen Arbeit aber die Ausnahme da. Auch in Abbildungen lassen sich die Farben dann verwenden (vgl. Abbildung 6).

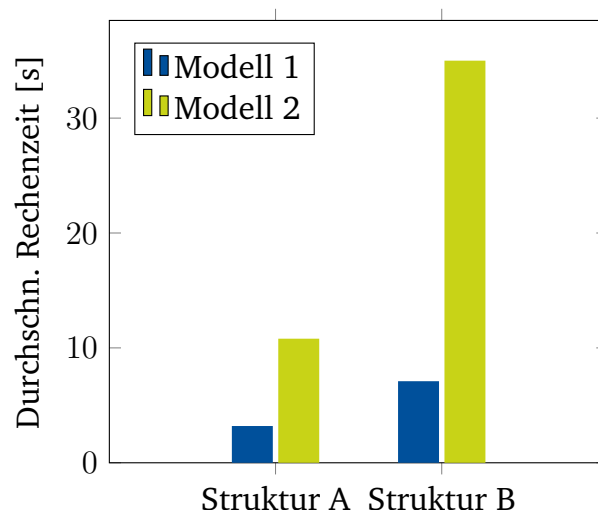


Abbildung 6: Farben in Abbildungen

2.8 Beispiele zu TikZ

Mit dem Paket `tikz` lassen sich auch andere Abbildungen direkt in Latex erstellen. Dies ist jedoch fortgeschritten und wird, insbesondere bei Bachelor- und Seminararbeiten, nicht von Ihnen erwartet. Wer interessiert ist, kann sich natürlich aber trotzdem die folgenden Beispiele anschauen, welche in Abbildung 7 und Abbildung 8 dargestellt sind.

Zu Tikz Grafiken findet sich vieles im Internet und auch KI-Tools wie ChatGPT lassen sich gut nutzen, um diese Grafiken zu erstellen.

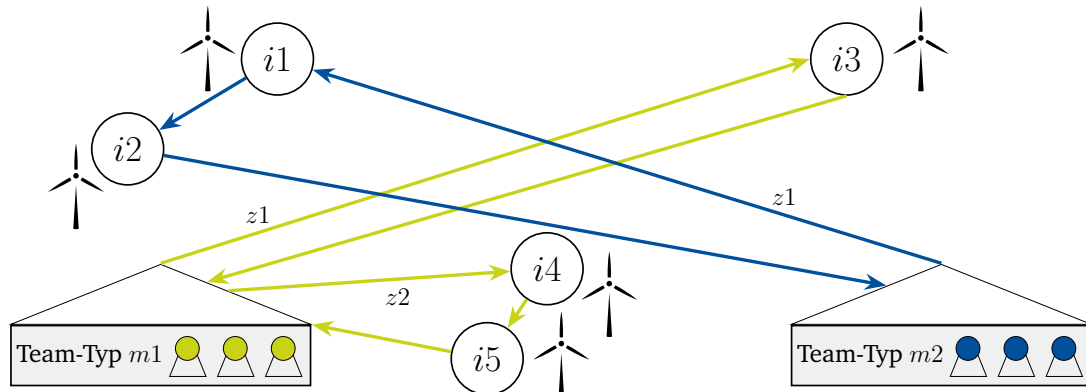


Abbildung 7: Dies ist ein Beispiel für ein Diagramm mit TikZ.
Tipp: Auch hierbei die zuvor definierten Farben nutzen.

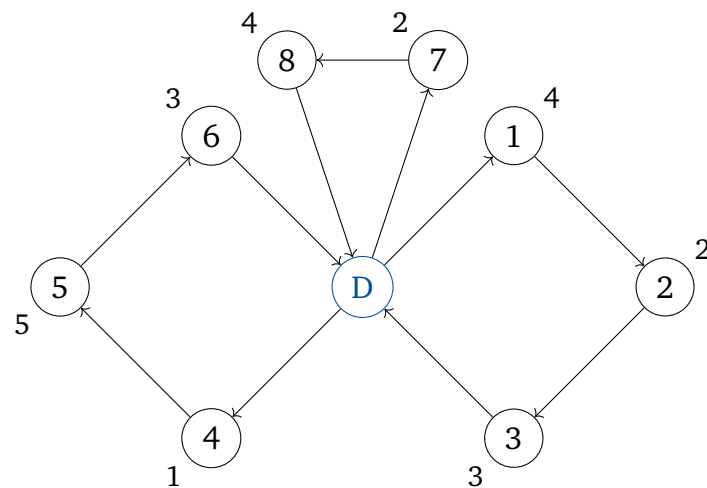


Abbildung 8: Dies ist ein weiteres Beispiel für eine Abbildung mit TikZ.

3 Ein neues Kapitel

Aus Gründen der Übersichtlichkeit kann es sinnvoll sein, die einzelnen Kapitel in unterschiedlichen Dateien zu organisieren und zu bearbeiten. Hier könnte nun ein neues Kapitel beginnen.

A Überschrift für den Inhalt des Anhangs

Wählen Sie eine aussagekräftige Überschrift für den Anhang, die den entsprechenden Inhalt beschreibt. Der Anhang kann z.B. umfangreiche Abbildungen oder Tabellen beinhalten. Der Anhang ist jedoch nicht dafür gedacht, Inhalte der Arbeit auszulagern, um bei Seminararbeiten so den Rahmen von 15 Seiten einzuhalten. Die Arbeit muss auch ohne den Anhang verständlich sein, d. h. alle wesentlichen Aussagen sind in den 15 Seiten der Seminararbeit darzustellen. Auch bei Abschlussarbeiten sollten nur weiterführende Informationen in den Anhang ausgelagert werden, um unnötiges Blättern beim Lesen der Arbeit zu vermeiden.

B Python-Code Beispiele

Es kann sinnvoll sein, den Anhang in einzelne Kapitel zu unterteilen. Wurde im Rahmen einer Arbeit eine Implementierung erstellt, so ist der Quellcode in den Anhang der Arbeit mit aufzunehmen. Dies geht am besten mit dem Listings-Paket. Im Textteil der Arbeit sollte dann auf den Code verwiesen werden, wie z.B. „Der zugehörige Quellcode ist in Anhang B in Quellcode 1 dargestellt“.

Quellcode 1: Hier kann eine Überschrift eingefügt werden, die den Code beschreibt

```
1 #import gurobipy as gp
2 from gurobipy import *

4 def Ablaufplanung_model(J, R, S, T, SJ, a, b, c, d, ls):
5     model = Model()

7     TF = model.addVars(J, S, vtype=GRB.CONTINUOUS, lb=0.0, name='TF')
9     X = model.addVars(J, S, T, vtype=GRB.BINARY, name='X')

11    #quicksum ist eine Funktion des package "gurobipy"
12    obj = c * quicksum(TF[j,ls[j]] for j in J)

14    model.setObjective(obj, GRB.MINIMIZE)

17    model.addConstrs((quicksum(X[j,s,t] for t in T) == 1 for j in J
        for s in SJ[j]), 'Einmal')

19    for j in J:
20        for indes, s in enumerate(SJ[j]):
21            if indes >= 1:
22                model.addConstr((TF[j,s] >= TF[j,S[indes-1]] + d[j,s
                    ]), 'Schritte')

24    model.addConstrs((quicksum((t index+1) * X[j,s,t] for tindex, t
        in enumerate(T)) == TF[j,s] for j in J for s in SJ[j]), '
        Zeitpunkte')

26    model.addConstrs((quicksum(a[j,s,r] * X[j,s,tau] for j in J for s
        in SJ[j] for indetau, tau in enumerate(T) if indet <= indetau
        <= indet + d[j,s] - 1) <= b[r,t] for r in R for indet, t in
        enumerate(T)), 'Ressourcen')

28    return model
```

Es kann schnell unübersichtlich werden, wenn der Code direkt in der .tex-Datei des Anhangs aufgeschrieben wird. Es können daher auch direkt Python-Dateien eingebunden werden, wie in Quellcode 2 dargestellt.

Quellcode 2: Auch hier ist eine Überschrift möglich

```

1  #import gurobipy as gp
2  from gurobipy import *

4  def Ablaufplanung_model(J, R, S, T, SJ, a, b, c, d, ls):
5      model = Model()

7      TF = model.addVars(J, S, vtype=GRB.CONTINUOUS, lb=0.0, name='TF')

9      X = model.addVars(J, S, T, vtype=GRB.BINARY, name='X')

11     #quicksum ist eine Funktion des package "gurobipy"
12     obj = c * quicksum(TF[j,ls[j]] for j in J)

14     model.setObjective(obj, GRB.MINIMIZE)

17     model.addConstrs((quicksum (X[j,s,t] for t in T) == 1 for j in J
        for s in SJ[j]), 'Einmal')

19     for j in J:
20         for indes, s in enumerate(SJ[j]):
21             if indes >= 1:
22                 model.addConstr((TF[j,s] >= TF[j,S[indes-1]] + d[j,s
                    ]), 'Schritte')

24     model.addConstrs((quicksum((t index+1) * X[j,s,t] for tindex, t
        in enumerate(T)) == TF[j,s] for j in J for s in SJ[j]), '
        Zeitpunkte')

26     model.addConstrs((quicksum(a[j,s,r] * X[j,s,tau] for j in J for s
        in SJ[j] for indetau, tau in enumerate(T) if indet <= indetau
        <= indet + d[j,s] - 1) <= b[r,t] for r in R for indet, t in
        enumerate(T)), 'Ressourcen')

28     return model

```

C Code-Beispiele für weitere Programmiersprachen

In einem weiteren Kapitel könnte nun z.B. auch noch der Code einer anderen Programmiersprache aufgeführt werden, wie z.B. ein GAMS-Quellexcode:

Quellexcode 3: Dies ist ein Beispiel mit GAMS

```
1  Sets
2  j          Jobs
3  r          Ressourcen
4  t          Perioden
5  s          Schritte
6  SJ(s, j) Schritte des Jobs j
7  ;

9  alias(t, tau);

11 Parameter
12 a(j, s, r) benötigte Kapazitätseinheiten der Ressource r für Schritt
    s des Jobs j
13 b(r, t)    Kapazität der Ressource r in Periode t
14 c          Umrechnungsfaktor von Perioden in Stunden
15 d(j, s)    Dauer von Schritt s des Jobs j
16 ls(j)      letzter Schritt von Job j
17 ;

19 Positive Variables
20 TF(j, s) Fertigstellungszeitpunkt von Schritt s des Jobs j
21 ;

23 Binary Variables
24 X(j, s, t) binäre Variable mit Wert 1 wenn in Periode t der Schritt s
    des Jobs j beendet wird 0 sonst
25 ;

27 Variables
28 Z          Zielfunktionswert
29 ;

31 Equations
32 1_ObjFunc          Minimierung der Gesamtdurchlaufzeit

34 2_Einmal(j, s)     Jeden Arbeitsgang einmal abschliessen
35 3_Schritte(j, s)   Einhaltung der Arbeitspläne aller Jobs
36 4_Zeitpunkte(j, s) Kopplung stetiger und binärer Variablen
37 5_Ressourcen(r, t) Kapazitätsgrenzen der Ressourcen r zum Zeitpunkt
    t
38 ;

40 * Modell
41 1_ObjFunc..
42     Z =e= c * sum((j, s)\$(ord(s)=ls(j)), TF(j, s));

45 2_Einmal(j, s)\$SJ(s, j)..
46     \textbf{sum}(t, x(j, s, t)) =e= 1;

48 3_Schritte(j, s)\$(ord(s)>=2 and ord(s)<=ls(j))..
49     TF(j, s) =g= TF(j, s-1) + d(j, s);

51 4_Zeitpunkte(j, s)$SJ(s, j)..
52     sum(t, ord(t) * x(j, s, t)) =e= TF(j, s);
```

```
54 5_Ressourcen(r, t)..
55     sum(j,
56         sum(s$SJ(s, j),
57             sum(tau$((ord(tau)>=ord(t)) and (ord(tau)<=ord(t)+d(j,s)-1)),
58                 a(j, s, r) * x(j, s, tau)))) =1= b(r, t);
```

Und natürlich geht das auch mit C++:

Quellcode 4: Dies ein Beispiel für C++

```
1  #include<stdio.h>
2  #include<iostream>
3  int main(void)
4  {
5      // Ausgabe
6      printf("Hello_World\n");
7      return 0;
8  }
```

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht sind und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

<Ort, Datum (tatsächliches Abgabedatum, vgl. Titelblatt)>

<Unterschrift>