



# BenchQC - Anwendungsfälle der Optimierung

OptWare GmbH

6. September 2023

## **Zusammenfassung**

Das Forschungsprojekt BenchQC, Anwendungsgetriebenes Benchmarking von Quantencomputern, befasst sich mit echten Anwendungsfällen der Industrie aus Machine Learning, Simulation und Optimierung. In diesem Dokument sollen die folgenden industriellen Anwendungsfelder der Optimierung betrachtet werden:

- Fließbandabstimmung
- Testfahrzeugkonfiguration
- ...

Die einzelnen Anwendungsfälle werden zunächst definiert, ihre klassischen Algorithmen erarbeitet und anschließend für die Anwendung auf Quantencomputern modifiziert. Abschließend werden die klassischen und QC Lösungen mithilfe eines Benchmarking Frameworks (QUARK) verglichen.

## Inhaltsverzeichnis

<b>1</b>	<b>Anwendungsfälle</b>	<b>3</b>
1.1	Anwendungsfall Fließbandabstimmung . . . . .	3
1.1.1	Grundproblem . . . . .	3
1.1.2	Transformation zu Bin Packing . . . . .	4
1.2	Anwendungsfall Testfahrzeugkonfiguration . . . . .	6
1.2.1	Transformation Testinkompatibilitäten . . . . .	8
1.2.2	Transformation zu Bin Packing mit Konflikten . . . . .	9
1.3	Anwendungsfall Reihenfolgeplanung (Sequencing) . . . . .	10
<b>2</b>	<b>Mathematische Grundprobleme</b>	<b>11</b>
2.1	Bin Packing . . . . .	11
2.1.1	Assembly Line Balancing Problem . . . . .	12
2.1.2	Bin Packing mit Konflikten . . . . .	14
<b>3</b>	<b>Versionshistorie</b>	<b>16</b>
	<b>Liste der noch zu erledigenden Punkte</b>	<b>18</b>

# 1 Anwendungsfälle

Im Folgenden sollen zunächst die industriellen Anwendungsfälle vorgestellt werden.

## 1.1 Anwendungsfall Fließbandabstimmung

Moderne Produktion, wie zum Beispiel die Automobilproduktion, erfolgt oft mithilfe der Fließbandfertigung. Bei der Fließbandfertigung durchläuft das zu fertigende Produkt auf dem Weg zur Fertigstellung zahlreiche Arbeitsstationen, in denen die einzelnen Arbeitsschritte zur Fertigstellung des Produktes ausgeführt werden.

Die Beplanung einer Fertigungslinie erfolgt dabei in zwei Schritten: Die Fließbandabstimmung und die Reihenfolgeplanung (siehe Abschnitt 1.3). Die Fließbandabstimmung beschäftigt sich damit, Arbeitsschritte nach ihrem mittleren Zeitbedarf auf Arbeitsstationen zu verteilen. In der Reihenfolgeplanung (dem Sequencing) werden konkrete Folgen von Aufträgen gebildet, die in Form einer operativen Feinplanung zu große Abweichung von den in der Fließbandabstimmung verwendeten Mittelwerten auf den einzelnen Stationen verhindern sollen. Die Fließbandabstimmung liefert somit eine taktische Grobplanung, auf der die operative Reihenfolge(fein)planung basiert.

Der mittlere Zeitbedarf der Arbeitsschritte, die als Ausgangsbasis der Fließbandabstimmung dient, ergibt sich aus dem tatsächlichen Zeitbedarf für einen Arbeitsschritt multipliziert mit der erwarteten Häufigkeit des Auftretens dieses Schrittes. Dauert z.B. die Montage einer Anhängerkupplung 100 Sekunden, und ist eine Anhängerkupplung nur bei jedem zehnten Fahrzeug zu erwarten, beträgt der mittlere Zeitbedarf 10 Sekunden.

### 1.1.1 Grundproblem

Gegeben ist eine Reihe von Arbeitsschritten, die auf Arbeitsstationen zugeordnet werden sollen. Dabei soll eine möglichst geringe Anzahl und eine im Mittel gleichmäßige Auslastung der Arbeitsstationen angestrebt werden. Pro Arbeitsstation steht maximal ein Takt zur Verfügung. Zum Taktschlag sollen die Arbeitsaufträge auf der nächsten Station bearbeitet werden. Somit darf die summierte durchschnittliche Arbeitszeit der Arbeitsschritte die Taktzeit nicht überschreiten.

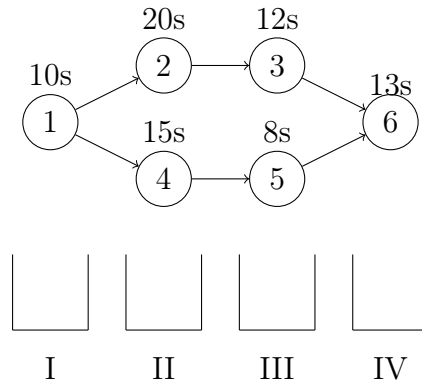


Abbildung 1: Beispiel für Fließbandabstimmung: Die einem Reihenfolgegraph unterliegenden Arbeitsschritte (1-6) sollen nach Zeitbedarf gleichmäßig auf Arbeitsstationen (I-IV) verteilt werden. Dabei darf die Arbeitslast pro Station 25 Sekunden nicht überschreiten.

Die Arbeitsschritte müssen teilweise in einer vorgegebenen Reihenfolge abgearbeitet werden. Abbildung 1 zeigt im oberen Teil einen beispielhaften Reihenfolgegraphen mit den Tätigkeiten (1-6) zu denen auch jeweils eine mittlere Dauer (z.B. 10 Sekunden bei Tätigkeit 1) angegeben ist. Tätigkeit 2 hängt von Tätigkeit 1 ab und kann deshalb erst durchgeführt werden, wenn Tätigkeit 1 bereits abgeschlossen ist. Gleiches gilt für Tätigkeit 3, sie hängt wiederum von Tätigkeit 2 ab. Tätigkeit 4 im Beispiel kann parallel zu Tätigkeit 2 erfolgen, da sie nur von Tätigkeit 1 abhängig ist und nicht etwa von Tätigkeit 2 oder 3.

In Abb. 1 sind im unteren Teil Arbeitsstationen (I-IV) als offene Behälter angedeutet. Den Arbeitsstationen sollen nun, unter Achtung des Reihenfolgegraphen, Tätigkeiten zugeteilt werden. Die Taktzeit beträgt 25 Sekunden. Die Zuordnung in Abb. 2 wäre denkbar, während die Zuordnung in Abb. 3 nicht möglich ist, da sie dem Reihenfolgegraphen widerspricht und Station III die Taktzeit überschreitet.

### 1.1.2 Transformation zu Bin Packing

Das Fließbandabstimmungsproblem lässt sich klassisch als Assembly Line Balancing Problem aus Abschnitt 2.1.1 aufstellen und lösen. Die Arbeitsschritte sind dabei die Aufgaben und die Arbeitsstationen die Stationen des klassischen Problems. Die Taktzeit wird durch die maximale Zeit pro Station

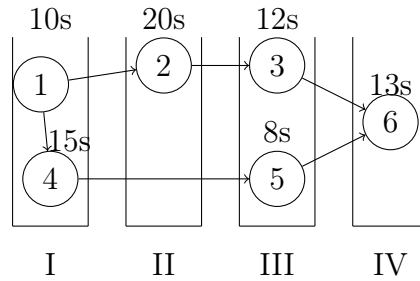


Abbildung 2: Mögliche Lösung für das Ausgangsproblem aus Abb. 1

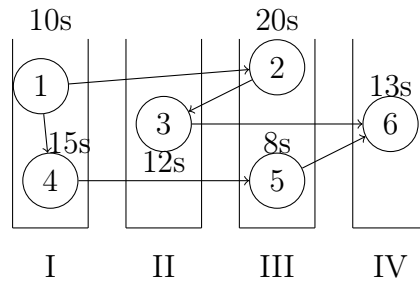


Abbildung 3: Unzulässige Lösung für das Ausgangsproblem aus Abb. 1

dargestellt.

Zusammengefasst, lässt sich die Fließbandabstimmung wie folgt auf die Notation aus Abschnitt 2.1.1 mappen:

Symbol	Bedeutung	Definitionsbereich
Mengen		
$T$	Menge der Arbeitsschritte	$\mathbb{N}$
$S$	Menge der Arbeitsstationen	$\mathbb{N}$
$R$	Menge an Abhängigkeitsbeziehungen zwischen den Arbeitsschritten	$\mathbb{N} \times \mathbb{N}$
Konstanten		
$c$	Taktzeit	$\mathbb{N}$
$v_t$	Benötigte Zeit für Arbeitsschritt $t \in T$	$\mathbb{N}$
Variablen		
$x_{ts}$	Genau dann, wenn $x_{ts} = 1$ wird Arbeitsschritt $t \in T$ der Station $s \in S$ zugeteilt	$\{0, 1\}$

Tabelle 1: Notation Fließbandabstimmung

## 1.2 Anwendungsfall Testfahrzeugkonfiguration

Bevor ein neues Fahrzeug in Serie produziert werden kann, müssen mit Vorserienfahrzeugen eine bestimmte Anzahl von Tests ausgeführt werden, um die Funktion der einzelnen Komponenten und die allgemeine Produzierbarkeit sicherzustellen. Dabei hat sich die Anzahl der zu testenden Funktionen in den letzten Jahren signifikant gesteigert. Da die Herstellung dieser Vorserienfahrzeuge sehr kostenintensiv und die Ausführung der dazugehörigen Tests sehr zeitaufwendig ist, soll für eine gegebene Anzahl an Tests die Zahl der zu produzierenden Testfahrzeuge minimiert werden.

Jeder Test gehört dabei einer bestimmten Kategorie an, die über die Reihenfolge der Ausführung entscheidet. Crashtests müssen zum Beispiel immer als letzter Test eines Vorserienautos durchgeführt werden, da diese Teile der Vorserienfahrzeuge zerstören können und somit die Ausführung der anderen Tests unmöglich machen.

Die genauen Zusammenhänge der Testkategorien können den Abb. 4 und 5 entnommen werden.

after <column>  
I can perform <line>

CAT	A	B	C	D	Z
A	✓	✗	✗	✗	✗
B	✓	✓	✗	✗	✗
C	✓	✓	✓	✗	✗
D	✓	✓	✓	✓	✗
Z	✓	✓	✓	✓	✗

before <line>  
I can perform <column>

Abbildung 4: Zusammenhang der Testkategorien als Matrix

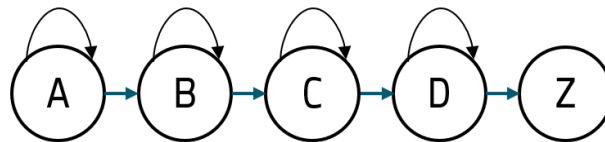


Abbildung 5: Zusammenhang der Testkategorien vereinfacht als gerichteter Graph

Die Testfahrzeuge werden in drei verschiedenen, teilweise überlappenden, Bauphasen produziert, die in Abb. 6 dargestellt sind. Jeder Test ist an Fahrzeuge einer bestimmten Bauphase gekoppelt. Eine Bauphase erstreckt sich über mehr als ein Jahr. Die Tests müssen spätestens ein Jahr nach Bau eines Fahrzeugs absolviert sein.

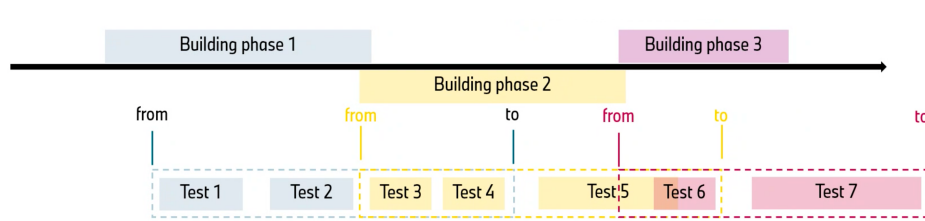


Abbildung 6: Bauphasen der Vorserienfahrzeuge

Pro konkrete Bauphase ist somit eine Menge an Tests vorgegeben, die auf  $n$  verschiedenen Testfahrzeugen durchgeführt werden müssen. Pro Test sind dabei gewisse Sonderausstattungen (SAs) vorgegeben oder ausgeschlossen. In Testfahrzeuge können je nach Modelltyp unterschiedliche Sonderausstattungen verbaut werden. Sonderausstattungen können andere Sonderausstattungen fordern oder ausschließen. Zum Beispiel kann ein Auto nicht gleichzeitig einen 4- und 8 Zylindermotor verbaut haben. Diese direkten Abhängigkeiten können auf die Tests übertragen werden, so dass eine minimale Anzahl an Testfahrzeugen gefunden werden kann.

### 1.2.1 Transformation Testinkompatibilitäten

Ziel ist es herauszufinden, welche Tests auf demselben Testfahrzeug getestet werden dürfen. Dies ergibt sich aus den folgenden Informationen:

- Abhängigkeiten und Ausschlüsse der SAs
- Verfügbare Testfahrzeugkategorien
- Forderung von unterschiedlichen Testfahrzeugen
- Zugehörigkeit zu Testkategorien

**Abhängigkeiten und Ausschlüsse der SAs** Für jeden Tests sollte bekannt sein, welche Sonderausstattung des Testfahrzeugs gefordert werden sollte. Neben den direkten Abhängigkeiten gibt es auch die indirekten, die sich aus den geforderten SAs ergeben. Wie die indirekten Abhängigkeiten auf die Tests übertragen werden können, sollen exemplarisch die nachfolgenden Regeln zeigen. Angenommen eine Sonderausstattung  $SA1$  wird gefordert.

Bedingung an SAs	Abhängigkeit im Test
$SA1 \Rightarrow SA2$	$SA2$ für Test benötigt
$SA1 \Rightarrow \sim SA2$	$\sim SA2$ für Test benötigt
$SA1 \Rightarrow SA2 \mid SA3$	$SA2$ oder $SA3$ für Test benötigt
Nur eins $SA1, SA2, SA3$	$\sim SA2$ und $\sim SA3$ für Test benötigt
$SA1 \& SA2$	$SA2$ für Test benötigt
$SA2 \Rightarrow \sim SA1$	dh. $SA1 \Rightarrow \sim SA2$ $\sim SA2$ für Test benötigt



Wenn eine benötigte *SA* oder ausgeschlossene *SA* gefunden wurde, müssen auch für diese die Regeln überprüft werden.

Am Ende ergibt sich eine Menge an Tests, für die bekannt ist, welche *SAs* auf dem Testfahrzeug sein müssen und welche *SAs* nicht erlaubt sind.

Tests, die sich ausschließende *SAs* haben, können somit nicht auf demselben Testfahrzeug getestet werden.

**Verfügbare Testfahrzeugkategorien** Des Weiteren sind auch die *SAs* auf den Testfahrzeugen eingeschränkt. Testfahrzeuge gehören unterschiedlichen Testfahrzeugkategorien an. Jede Testfahrzeugkategorie hat erlaubte *SAs*. Ein Test kann nur dann durchgeführt werden, wenn mindestens ein Testfahrzeugkategorie alle benötigten Sonderausstattungen erlaubt. Andernfalls kann dieser Test nicht ausgeführt werden und die Verteilung aller Tests ist nicht möglich.

Tests, die nicht auf demselben Testfahrzeug ausgeführt werden können, können ebenfalls gegenseitig ausgeschlossen werden.

**Forderung von unterschiedlichen Testfahrzeugen** Normalerweise müssen Tests mehr als einmal auf unterschiedlichen Testfahrzeugen durchgeführt werden. Diese Tests schließen sich somit auch gegenseitig aus.

**Zugehörigkeit zu Testkategorien** Aus den Testfahrzeugkategorien ergibt sich die Ausführungsreihenfolge der Tests. Tests, die zur Kategorie ‘Z’ gehören, dürfen allerdings nur einmal pro Fahrzeug existieren. Diese Tests sind somit gegenseitig auszuschließen, damit gewährleistet werden kann, dass nur ein Test pro Fahrzeug zugeteilt werden kann. Tests anderer Kategorien müssen nicht besonders behandelt werden.

### 1.2.2 Transformation zu Bin Packing mit Konflikten

Gegeben ist eine Liste an Tests einer konkreten Bauphase, die genau einmal durchgeführt werden müssen. Für jeden Test ist bekannt:

- welche Tests nicht am selben Fahrzeug durchgeführt werden dürfen,
- wie lange für die Durchführung benötigt wird,
- wann er frühestmöglich ausgeführt werden darf und

- bis wann er spätestens ausgeführt werden muss.

Um die Kosten möglichst gering zu halten, soll die Anzahl der Testfahrzeuge minimal gehalten werden. Alle Tests müssen ausgeführt werden.

Das Testfahrzeugkonfigurationsproblem lässt sich auf das spezielle Bin Packing Problem mit Konflikten (siehe Abschnitt 2.1.2) abbilden.

Die Testfahrzeuge sind dabei die Behälter, deren Anzahl im Optimum möglichst gering sein muss. Die Objekte sind die auszuführenden Tests, die in dieser Bauphase abgeschlossen sein müssen. Die Test-Durchführungszeit ist im klassischen Problem das Gewicht der Objekte. Die maximale Kapazität pro Testfahrzeug ist ein Jahr, da alle Tests nacheinander ausgeführt werden müssen.

Die letzten beiden Eigenschaften der Tests bezüglich der Ausführungszeit ignorieren wir zunächst.

Zusammengefasst, lässt sich die Testfahrzeugkonfiguration wie folgt auf die Notation aus Abschnitt 2.1.2 mappen:

Symbol	Bedeutung	Definitionsbereich
Mengen		
$O$	Menge der Tests	$\mathbb{N}$
$B$	Menge der Testfahrzeuge	$\mathbb{N}$
$I$	Menge der inkompatiblen Tests	$\mathbb{N} \times \mathbb{N}$
Konstanten		
$c$	Ein Jahr	$\mathbb{N}$
$w_o$	Benötigte Zeit für Test $o \in O$	$\mathbb{N}$
Variablen		
$x_{ob}$	Genau dann, wenn $x_{ob} = 1$ wird Test $o \in O$ dem Testfahrzeug $b \in B$ zugeteilt	$\{0, 1\}$

Tabelle 2: Notation zu Testfahrzeugkonfiguration

### 1.3 Anwendungsfall Reihenfolgeplanung (Sequencing)

Wie bereits in der Fließbandabstimmung beschrieben, erfolgt die moderne Produktion, wie zum Beispiel die Automobilproduktion, oft mithilfe der

Fließbandfertigung. Bei der Fließbandfertigung durchläuft das zu fertigende Produkt auf dem Weg zur Fertigstellung zahlreiche Arbeitsstationen, in denen die einzelnen Arbeitsschritte zur Fertigstellung des Produktes ausgeführt werden.

Nachdem in der Fließbandabstimmung (siehe Abschnitt 1.1) die einzelnen Arbeitsschritte auf die Arbeitsstationen verteilt wurden, geht es anschließend bei der Reihenfolgeplanung (dem Sequencing) darum, konkrete Arbeitsaufträge, die die einzelnen Stationen durchlaufen müssen, in die optimale Reihenfolge zu bringen. Die operative Feinplanung muss dafür Sorgen, dass zu keinem Zeitpunkt zu große Abweichungen, von den in der Fließbandabstimmung verwendeten Mittelwerten pro Station, entstehen. ...

## 2 Mathematische Grundprobleme

Im Folgenden sollen die mathematischen Grundprobleme analysiert werden.

### 2.1 Bin Packing

Beim Bin Packing Problem (oder auch Behälterproblem) geht es darum, eine gegebene Anzahl an Objekten auf eine kleinstmögliche Anzahl an Behältern zu verteilen, sodass kein Behälter überläuft.

Als ganzzahliges Optimierungsproblem lässt sich das Bin Packing Problem wie folgt darstellen: Jedes Objekt muss genau einem Behälter zugeteilt werden (2). Pro Behälter darf die maximale Kapazität nicht überschritten werden (3). Das Ziel ist, die Anzahl der Behälter zu minimieren, was mithilfe der Zielfunktion (1) erreicht wird.

Die Zielfunktion minimiert die Anzahl der Behälter nicht immer. In der Literatur werden andere Formulierungen genutzt, die jedoch mehr binäre Variablen benötigen. Daher sollte die Wahl der Zielfunktion noch weiter diskutiert werden.

Symbol	Bedeutung	Definitionsbereich
Mengen		
$O$	Menge der Objekte	$\mathbb{N}$
$B$	Menge der Behälter	$\mathbb{N}$
Konstanten		
$c$	Maximale Kapazität der Behälter	$\mathbb{N}$
$w_o$	Gewicht von Objekt $o \in O$	$\mathbb{N}$
Variablen		
$x_{ob}$	$x_{ob} = 1$ gdw. Objekt $o \in O$ dem Behälter $b \in B$ zugeteilt wird	$\{0, 1\}$

Tabelle 3: Notation Bin Packing

$$\min \sum_{b \in B, o \in O} b \cdot x_{ob} \quad (1)$$

$$s.t. \quad \sum_{b \in B} x_{ob} = 1, \quad \forall o \in O \quad (2)$$

$$\sum_{o \in O} w_o \cdot x_{ob} \leq c, \quad \forall b \in B \quad (3)$$

$$x_{ob} \in \{0, 1\}, \quad \forall o \in O, b \in B \quad (4)$$

### 2.1.1 Assembly Line Balancing Problem

Eine interessante Erweiterung des Bin Packing Problems ergibt sich durch die Einführung von Reihenfolgebeziehungen zwischen den Objekten.

In der Literatur ist das Problem auch als Simple Assembly Line Balancing Problem (SALBP) bekannt. Bei diesem geht es darum, eine Reihe von Aufgaben mit Vorrangbedingungen auf mehrere hintereinanderliegende Stationen zu verteilen [BSS22]. Bei der im Folgenden betrachteten Problemvariante SALBP-1 ist die maximale Zeit für die Ausführung aller Aufgaben, die einer Station zugeteilt werden, vorgegeben, während die Anzahl der Stationen minimiert werden soll.

Die Objekte des Bin Packing Problems sind somit nun die Aufgaben und die Behälter die Stationen des Problems.

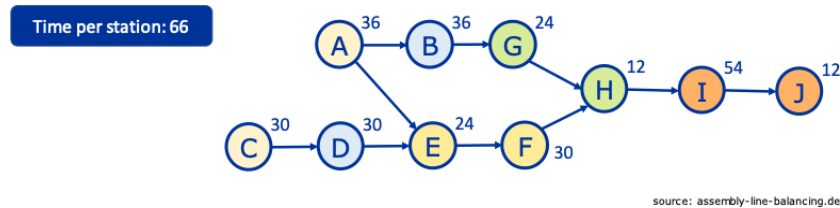


Abbildung 7: Beispiel Assembly Line Balancing Problem und Lösung

Als ganzzahliges Optimierungsproblem lässt sich SALBP-1 wie folgt darstellen: Jede Aufgabe muss genau einer Station zugeteilt werden (6). Pro Station muss die maximale Bearbeitungszeit eingehalten werden (7). Die Reihenfolgebedingungen müssen eingehalten werden. Das heißt, wenn zwischen zwei Aufgaben eine vorgegebene Reihenfolge gilt, dann darf die zweite Aufgabe nicht auf einer Station durchgeführt werden, die vor der ersten Aufgabe liegt (8). Das Ziel ist, die Anzahl der benötigten Stationen zu minimieren, was mithilfe der Zielfunktion (5) erreicht wird.

Siehe Kommentar in Abschnitt 2.1

Symbol	Bedeutung	Definitionsbereich
Mengen		
$T$	Menge der Aufgaben	$\mathbb{N}$
$S$	Menge der Stationen	$\mathbb{N}$
$R$	Menge an Vorrangbedingungen zwischen zwei Aufgaben	$\mathbb{N} \times \mathbb{N}$
Konstanten		
$c$	Taktzeit	$\mathbb{N}$
$v_t$	Benötigte Zeit für Aufgabe $t \in T$	$\mathbb{N}$
Variablen		
$x_{ts}$	$x_{ts} = 1$ gdw. Aufgabe $t \in T$ Station $s \in S$ zugeteilt wird	$\{0, 1\}$

Tabelle 4: Notation SALBP-1

$$\min \sum_{s \in S, t \in T} s \cdot x_{ts} \quad (5)$$

$$s.t. \sum_{s \in S} x_{ts} = 1, \quad \forall t \in T \quad (6)$$

$$\sum_{t \in T} v_t \cdot x_{ts} \leq c, \quad \forall s \in S \quad (7)$$

$$\sum_{s \in S} s \cdot x_{ts} \leq \sum_{s \in S} s \cdot x_{t's}, \quad \forall (t, t') \in R \quad (8)$$

$$x_{ts} \in \{0, 1\}, \quad \forall t \in T, s \in S \quad (9)$$

### 2.1.2 Bin Packing mit Konflikten

Eine weitere interessante Erweiterung des Bin Packing Problems ergibt sich durch die Einführung von inkompatiblen Objekten. Inkompatible Objekte müssen zwingend auf unterschiedliche Behälter aufgeteilt werden (13).

Dadurch ergibt sich das folgende ganzzahlige Optimierungsproblem:

Symbol	Bedeutung	Definitionsbereich
<b>Mengen</b>		
$O$	Menge der Objekte	$\mathbb{N}$
$B$	Menge der Behälter	$\mathbb{N}$
$I$	Menge der inkompatiblen Objekte	$\mathbb{N} \times \mathbb{N}$
<b>Konstanten</b>		
$c$	Maximale Kapazität der Behälter	$\mathbb{N}$
$w_o$	Benötigte Größe für Objekt $o \in O$	$\mathbb{N}$
<b>Variablen</b>		
$x_{ob}$	Genau dann, wenn $x_{ob} = 1$ wird Objekt $o \in O$ dem Behälter $b \in B$ zugeteilt	$\{0, 1\}$

Tabelle 5: Notation zu Bin Packing mit Konflikten

$$\min \sum_{b \in B, o \in O} b \cdot x_{ob} \quad (10)$$

$$s.t. \sum_{b \in B} x_{ob} = 1, \quad \forall o \in O \quad (11)$$

$$\sum_{o \in O} w_o \cdot x_{ob} \leq c, \quad \forall b \in B \quad (12)$$

$$x_{ob} + x_{o'b} \leq 1, \quad \forall (o, o') \in I, b \in B \quad (13)$$

$$x_{ob} \in \{0, 1\}, \quad \forall o \in O, b \in B \quad (14)$$

### 3 Versionshistorie

Version	Datum	Autor(en)	Änderungen
0.0	2023-07-28	OptWare	Initiale Version: Beschreibung Use-Case, Bin Packing
0.1	2023-09-05	OptWare	Anpassung Beschreibung Testfahrzeugkonfiguration



## Literatur

- [BSS22] Nils Boysen, Philipp Schulze und Armin Scholl. “Assembly line balancing: What happened in the last fifteen years?” In: *European Journal of Operational Research* 301.3 (2022), S. 797–814. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2021.11.043>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221721009942>.

## Liste der noch zu erledigenden Punkte

Liste ist nicht vollständig! . . . . .	8
Die Zielfunktion minimiert die Anzahl der Behälter nicht immer. In der Literatur werden andere Formulierungen genutzt, die jedoch mehr binäre Variablen benötigen. Daher sollte die Wahl der Ziel- funktion noch weiter diskutiert werden. . . . .	11
Siehe Kommentar in Abschnitt 2.1 . . . . .	13