

Generierung realitätsnaher Testdaten für die Simulation von Produktionen

Generation of realistic test data for the simulation of productions

Martin Krockert, Marvin Matthes, Torsten Munkelt, Hochschule für Technik und
Wirtschaft Dresden, Dresden (Germany), martin.krockert@htw-dresden.de,
marvin.matthes@htw-dresden.de, torsten.munkelt@htw-dresden.de
Sven Völker, Technische Hochschule Ulm, Ulm (Germany), sven.voelker@thu.de

Abstract: Test data are required for the simulative validation of production planning and control (PPC) procedures. To comprehensively determine the performance of PPC methods, it is necessary to examine different productions based on corresponding test data. A generator for test data is presented, which generates realistic master data based on structure-describing characteristics of a production. In addition, a method for determining the intermediate arrival time of customer orders is presented, which ensures a desired utilization of the production, which the generated test data describe. The validation of the test data generation shows that already for relatively small productions test data are generated, whose structure-describing characteristics correspond to the given parameters. With the same parameterization, the test data generator can generate reproducible test data and provide it for the simulative test of any PPC procedure.

1 Einleitung

Für die simulative Validierung von Verfahren der Produktionsplanung und -steuerung (PPS) werden Testdaten benötigt. Testdaten können auf drei unterschiedliche Arten gewonnen werden (siehe Tabelle 1).

Jede dieser Vorgehensweisen zur Erstellung von Testdaten hat Vor- und Nachteile. Die Gewinnung von Testdaten aus Praxisdaten ist meist mit sehr aufwendiger Extraktion und Transformation verbunden. Zudem sind die Daten meist unvollständig, schlecht gepflegt oder aus Datenschutzgründen schlicht nicht für eine Veröffentlichung geeignet (D. Libes et al., 2017).

Tabelle 1: Arten der Testdatengewinnung in Anlehnung an (Thomas Döring et al., 1999)

	Praxisdaten	Daten aus Problembibliothek	Generierte Daten
Korrektheit	O	+	+
Vollständigkeit	O	+	+
Umfang	+	O	+
Festlegung von Verteilung	-	-	+
Realitätsnähe	+	-	O
Bereitstellungsaufwand	-	+	O
Datenschutzkonformität	-	+	+

- schlecht erfüllt; O wenig erfüllt; + gut erfüllt

Die Nutzung von Problembibliotheken ist dann hilfreich, wenn ein neues Verfahren auf eine konkrete Problemstellung angewendet und mit bestehenden Verfahren verglichen werden soll. Eine bekannte Bibliothek für Problemstellungen unterschiedlichster Art stellt die OR-Library von Beasley (1990) dar. Beasley stellt in der Bibliothek u.a. statische Testdatensätze für verschiedene Fertigungstypen einer Produktion zur Verfügung. Der Nachteil von Problembibliotheken ergibt sich aus der hohen Spezialisierung und dem eher geringen Umfang der Daten innerhalb eines Testdatensatzes. So bilden die Probleme nur einen kleinen Teil möglicher Probleme ab, sind zu speziell oder bilden, aufgrund des Auslassens von Nebenbedingungen, nicht genau das Problem ab, welches das zu testende Verfahren der PPS lösen soll.

Testdatengeneratoren erzeugen Testdaten hingegen einfach, vollständig und in beliebigem Umfang (Sven Völker et al., 2001). Sie können darüber hinaus nicht nur zur Validierung von Verfahren der Produktionsplanung und -steuerung, sondern auch zur Erzeugung von umfangreichen Testdatensätze genutzt werden u.a. für die Bereitstellung von Trainingsdaten für den Einsatz von künstlicher Intelligenz in der Produktion (D. Libes et al., 2017). Im Gegensatz zu Praxisdaten entsprechen die generierten Daten nur in ihren strukturbeschreibenden Kenngrößen den realen Praxisdaten. Es existieren bereits viele kommerzielle und nicht-kommerzielle Generatoren für Testdaten. Bekannte Generatoren sind u.a. Bogus in C# oder Faker in Python. Darüber hinaus gibt es viele dedizierte Lösungen zur Generierung von Testdaten, wie z.B. Mockaroo oder FINRA DataGenerator. Die Generatoren haben keinen spezifischen Anwendungsbezug und ermöglichen die Generierung von einfachen semantisch randomisierten und teilweise einfachen hierarchisch strukturierten Testdaten. Die Generierung komplexer Testdaten, wie sie in Form von aufeinander aufbauenden Produktstrukturen und Arbeitsplänen in der Produktion vorliegen, ist mit diesen Generatoren nicht möglich. Neben den generischen Testdatengeneratoren, existieren bereits Testdatengeneratoren im Umfeld der Produktion. Fernandes et al. (2020) stellt einen Generator von Problemszenarien vor, bei denen verschiedene Fertigungstypen mit variierbaren Eingabeparametern simuliert werden können. Bei diesen Simulationen werden jedoch keine Produktstrukturen generiert, womit sich der Generator auch nicht für die Validierung von verschiedenen Verfahren zur Produktionsplanung und -steuerung eignet. Darüber

hinaus gibt es Generatoren für Produktstrukturen aus dem Anwendungsgebiet des Product-Lifecycle-Management, welche auf Basis von CAD-Modellen, Produktstrukturen generieren (Adolphy et al., 2015). Die synthetische Generierung von Produktstrukturen auf Basis strukturbeschreibenden Kenngrößen einer Produktion ist mit diesen Generatoren nicht möglich.

Die bisherigen Testdatengeneratoren genügen unseren Anforderungen nicht, da sie keine Testdaten in Form von Stammdaten für die Simulation von Produktionen bereitstellen oder nur einzelne detaillierte Produktstrukturen erzeugen. Deshalb stellen wir eine Methode zur automatischen Generierung von Stammdaten vor. Die generierten Testdaten sollen keine semantische, sondern eine strukturelle Realitätsnähe aufweisen. Dazu werden durch den Testdatengenerator Erzeugnisstrukturen, Arbeitspläne, Maschinenstrukturen und Werkzeuge generiert, welche in ihren Bezeichnungen generisch sind, in ihren Strukturen jedoch den vorgegebenen strukturbeschreibenden Kenngrößen entsprechen, die auch aus Realdaten gewonnen werden können.

Des Weiteren stellen wir ein Verfahren zur Generierung von Bewegungsdaten (Kundenaufträgen) bereit, mit denen für die zuvor erstellten Stammdaten eine ausgewogene Auslastung der Produktion gewährleistet wird. Ziel ist die Bereitstellung eines Generators zum Erzeugen realitätsnaher und reproduzierbarer Testdaten für die Simulation von Produktionen und den Test von Verfahren der PPS für andere Forschende und uns.

Im folgenden Kapitel werden die zugrundeliegenden strukturbeschreibenden Kenngrößen eingeführt. Darauf aufbauend, beschreibt Kapitel drei die Vorgehensweise des Generators. Das anschließende Kapitel stellt die Validierung des Testdatengenerators vor. Dabei wird u.a. ausgewertet, wie gut die aus den Testdaten gewonnenen strukturbeschreibenden Kenngrößen den vorgegebenen entsprechen.

2 Strukturbeschreibende Kenngrößen einer Produktion

Die Daten einer Produktion werden in Stammdaten und Bewegungsdaten unterteilt. Stammdaten beschreiben u.a. Materialien, Stücklisten, Arbeitspläne und Maschinen. Bewegungsdaten werden aus Stammdaten erzeugt und umfassen beispielsweise Kundenaufträge und Produktionsaufträge. Die Generierung von Testdaten zielt auf Stammdaten ab. Zur Generierung müssen zunächst die folgenden strukturbeschreibenden Kenngrößen aus den realen Daten extrahiert oder bei unvollständiger Datenlage geschätzt werden.

2.1 Erzeugnisstruktur

Die Erzeugnisstruktur entspricht einem gerichteten, azyklischen Graphen, welcher in der Produktion als Gozinto-Graph bezeichnet wird (Kurbel, 2016). Die gerichteten Kanten (e) zwischen den als Knoten (v) repräsentierten Materialien stellen eine „geht ein in“-Beziehung zwischen Materialien dar. Das Kantengewicht beschreibt die Menge des Quellmaterials, welche in das Zielmaterial eingehen. Für den Testdatengenerator werden zur Beschreibung des Erzeugnisstruktur-Graphen zum einen die Anzahl der Knoten ohne ausgehende Kante ($V_{Endprodukt}$), der Eingangsgrad (Komplexitätsgrad) und der Ausgangsgrad (Mehrfachverwendungsgrad) angeben. Der Längste Pfad von $V_{Endprodukt}$ zu $V_{Kaufteilen}$ ist die maximale

Erzeugnisstrukturtiefe. Die Anzahl der Knoten ohne eingehende Kante ($V_{Kaufteil}$) ergibt sich aus der Anzahl der Endprodukte sowie dem Verhältnis aus Mehrfachverwendungsgrad und Komplexitätsgrad.

$$V_{sales\ product} \leftarrow \{v \in V | e_G^-(v) = 0\} \quad (1)$$

$$V_{purchase\ product} \leftarrow \{v \in V | e_G^+(v) = 0\} \quad (2)$$

Der Mehrfachverwendungsgrad (MV) bzw. Ausgangsgrad des Graphen beschreibt, wie oft ein Produkt in anderen Produkten verwendet wird, wobei V die Menge an Produkten repräsentiert und $succ(v)$ die Menge der in das Produkt v eingehenden Produkte (Heinrich, 1987).

$$DM = \frac{\sum_{v \in V} |succ(v)|}{|V \setminus V_{sales\ product}|} \quad (3)$$

Der Komplexitätsgrad (KG) bzw. Eingangsgrad des Graphen gibt an, aus wie vielen anderen Produkten ein Produkt durchschnittlich besteht, wobei $pred(v)$ die Menge der Produkte repräsentiert, die in das Produkt p eingehen.

$$DC = \frac{\sum_{v \in V} |pred(v)|}{|V \setminus V_{purchase\ product}|} \quad (4)$$

Die maximale Erzeugnisstrukturtiefe (EST) beschreibt die Anzahl der Kanten des längsten gerichteten Pfades im Erzeugnisstrukturgraphen. Darüber hinaus werden der Erwartungswert und die Standardabweichung für das Kantengewicht $w_{a,b}$ im Erzeugnisstrukturgraphen angegeben, wobei das Kantengewicht die Menge eines in Produkt a eingehenden Materials b angibt.

$$EST = \max(est(V_{sales\ product})) \quad (5)$$

2.2 Maschinenstrukturen

Für jede zu generierende Maschinengruppe (m) werden die Anzahl der Maschinen (r), der Erwartungswert (μ) und die Standardabweichung (σ) für die Bearbeitungszeit der auf Maschinen dieser Maschinengruppe durchzuführenden Arbeitsgänge angegeben. Zusätzlich umfassen strukturbeschreibende Kenngrößen für Maschinenstrukturen auch die Anzahl der Werkzeuge (w) pro Maschinengruppe und eine Angabe zur Rüstzeit (ϑ).

$$m = (r, \mu, \sigma, w, \vartheta)^T \quad (6)$$

2.3 Arbeitsplanstrukturen

Arbeitsplanstrukturen beschreibt der Organisationgrad (OG) als strukturbeschreibende Kenngröße. Aufgrund des Organisationsgrades werden Arbeitspläne so erzeugt, dass sie einem bestimmten Fertigungstyp entsprechen.

Der Organisationsgrad (OG) ist ein Maß für den Fertigungstyp. Mithilfe des OGs kann stufenlos zwischen einzelnen Fertigungstypen einer Produktion, von Werkstatt- über Reihen- bis hin zu Fließfertigung unterschieden werden (Corsten und Gössinger, 2012). In der nachfolgenden Formel repräsentiert m die Anzahl der

Maschinengruppen und π_{ij} die Wahrscheinlichkeit des Übergangs von der Maschinengruppe i auf die Maschinengruppe j (Thomas Döring et al., 1999).

$$OG = \frac{1}{M-1} \sum_{i=1}^M \sum_{j=1}^M \left(\pi_{ij} - \frac{1}{M} \right)^2 \tag{8}$$

3 Konzept für den Testdatengenerator

Abbildung 1 zeigt den grundlegenden Ablauf zur Generierung der Testdaten. Die Generierung der Testdaten ist in zwei Abschnitte untergliedert.

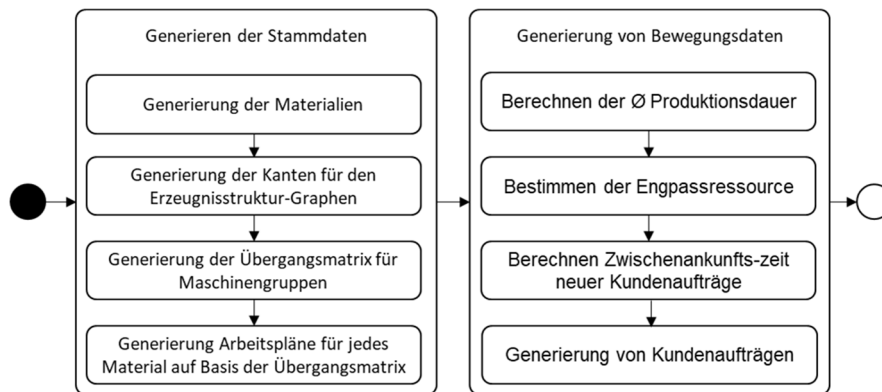


Abbildung 1: Ablauf der Testdatengenerierung

3.1 Generierung der Stammdaten

Auf Basis der strukturbeschreibenden Kenngrößen, KG , MV und EST , generiert Algorithmus 1 im ersten Schritt der Generierung die Materialien bzw. Knoten des Erzeugnisstrukturgraphen.

Algorithmus 1: Generierung der Produkte

```

begin
  V ← ∅
  for est = 1 to EST
    for i = 0 to ⌊(KG / MV)⌋est-1
      V ← V ∪ {vest,i}
    end
  end
end
  
```

Bei einer konvergierenden Struktur ($MV < KG$) beginnt die Generierung mit dem Hinzufügen der Kanten bei $V_{Endprodukt}$ (siehe Algorithmus 2) und bei einer divergierenden Struktur bei $V_{Kaufteil}$ (siehe Algorithmus 3). Es wird sichergestellt, dass jeder Knoten v im Erzeugnisstrukturgraphen G wenigstens über eine Kante e mit einem anderen Knoten verknüpft ist.

Algorithmus 2: Generierung der initialen Kanten für konvergierende Strukturen

```

begin
  for i = 1 increment to EST - 1
    for each k in  $V_i$ 
       $E \leftarrow E \cup \{(k, \text{beliebiges } \{v \in V_{i+1} \mid e_G^-(v) = 0\}, k)\}$ 
    end for
  end for
  for i = EST decrement to 2
    for each k in  $\{v \in V_i \mid e_G^-(v) = 0\}$ 
      beliebiges j mit  $p(j) = \frac{2^j}{i(i-1)}$  mit  $1 \leq j < i$ 
       $E \leftarrow E \cup \{(k, \text{beliebiges } \{v \in V_j\})\}$ 
    end for each
  end for
end

```

Algorithmus 3: Generierung der initialen Kanten für divergierende Strukturen

```

begin
  for i = EST decrement to 2
    for k in  $V_i$ 
       $E \leftarrow E \cup \{(k, \text{beliebiges } \{v \in V_{i-1} \mid e_G^+(v) = 0\})\}$ 
    end for each
  end for
  for i = 1 increment to EST - 1
    for each k in  $\{v \in V_i \mid e_G^+(v) = 0\}$ 
      beliebiges j mit  $p(j) = \frac{2^{(j-1)}}{(EST-i)(EST-i+1)}$  mit  $1 \leq j < i$ 
       $E \leftarrow E \cup \{(\text{beliebiges } \{v \in V_j\}, k)\}$ 
    end for each
  end for
end

```

Anschließend erzeugt Algorithmus 4 in Abhängigkeit vom KG und MV weitere Kanten. Die dabei entstehenden Multikanten zwischen zwei Knoten werden durch den Algorithmus bewusst nicht ausgeschlossen, da Multikanten auch in realen Produktstrukturen vorhanden sind.

Algorithmus 4: Generierung der zusätzlichen Kanten

```

begin
  for j =  $|E| + 1$  increment to  $\max(|V \setminus V_{\text{Endprodukt}}|, |V \setminus V_{\text{Kaufteil}}|)$ 
     $v_{ij} \leftarrow \text{beliebiges } \{v \in V \setminus V_{\text{Endprodukt}}\}$ 
    beliebiges k mit  $p(k) = \frac{2^k}{i(i-1)}$  mit  $1 \leq k < i$ 
     $E \leftarrow E \cup \{(v_{ij}, \text{beliebiges } \{v \in V_k\})\}$ 
  end for
end

```

Algorithmus 5 erzeugt eine Maschinenübergangsmatrix Π für die zu generierenden Arbeitspläne. Im ersten Schritt erzeugt er eine erweiterte $(|M| + 1) \times (|M| + 1)$ -Übergangsmatrix Π^A , bei welcher die Wahrscheinlichkeit π_{ij} den Übergang von Maschinengruppe m_i zu Maschinengruppe m_j beschreibt. Die Matrix ist insofern erweitert, als dass sie eine erste Zeile für die Quelle ein und eine letzte Spalte für die

Senke enthält. Eine $(|M| + 1) \times (|M| + 1)$ -Hilfsmatrix Π^B , wird mit $OG^B = 0$ generiert, wenn OG^A größer als der gewählte OG ist. Umgekehrt wird bei einem OG^A kleiner als der gewählte OG eine Hilfsmatrix Π^B mit $OG^B = 1$ erzeugt. Mittels eines Halbierungsverfahrens und der Hilfsmatrix Π^B , wird die Matrix Π^A angenähert, bis OG^A dem gewählten OG mit einer zuvor gewählten Mindestgenauigkeit übereinstimmt.

Algorithmus 5: Algorithmus zur Generierung der Übergangsmatrix für Maschinengruppen

```

begin
   $\Pi^A := \pi_{ij} \leftarrow \begin{cases} \frac{\lambda^{(j-i)}}{(j-i)!} & i \leq j \\ \frac{\lambda^{(i-j)}}{2 \cdot (i-j)!} & i > j \end{cases} \text{ mit } 0 < \lambda \leq 1$ 
   $\pi_i \leftarrow \sum_{j=1}^M \pi_{ij}$ 
   $\Pi^A := \pi_{ij} \leftarrow \frac{\pi_{ij}}{\pi_i}$ 
  if  $OG^A > OG$  then
     $\Pi^B := \pi_{ij} \leftarrow \left(\frac{1}{M}\right)_{(|M|+1) \times (|M|+1)}$ 
  else
     $\Pi^B := \pi_{ij} \leftarrow \begin{cases} 1 & j = i \\ 0 & \text{sonst} \end{cases}$ 
  end if
  while  $|OG^A - OG| > 0,001$ 
     $\Pi^C := \frac{1}{2} (\Pi^A + \Pi^B)$ 
    if  $OG^A - OG = OG^C - OG$  then
       $\Pi^A \leftarrow \Pi^C$ 
    else
       $\Pi^B \leftarrow \Pi^C$ 
    end if
  end while
   $\Pi \leftarrow \Pi^A$ 
end
```

Unter Anwendung der Maschinenübergangsmatrix Π erzeugt der Generator für alle Materialien eine Sequenz von Arbeitsplanpositionen. Für jede Arbeitsplanposition wird basierend auf dem Erwartungswert $\mu(m)$ und der Standardabweichung $\sigma(m)$ der gewählten Maschinengruppe eine Bearbeitungszeit ermittelt. Zusätzlich zur Maschinengruppe wird der Arbeitsplanpositionen ein Werkzeug aus dem Werkzeugpool der jeweiligen Maschinengruppe zugewiesen. Die Zuweisung der Werkzeuge erfolgt gleichverteilt. Mit der Erstellung der Arbeitspläne ist die Generierung der Stammdaten abgeschlossen.

3.2 Generierung von Bewegungsdaten

Nun werden für die generierten Stammdaten Bewegungsdaten erzeugt. Der Generator errechnet eine geeignete Zwischenankunftszeit für Kundenaufträge, um eine vorgegebene Auslastung u (z.B. 80%) auf den Engpassressourcen zu erzeugen. Die Berechnung erfolgt auf Basis der eingegebenen strukturbeschreibenden Kennzahlen. Mithilfe der Zwischenankunftszeit können ein kontinuierlicher Kundenauftragsstrom

oder für einen vorgegebenen Zeitraum alle Kundenaufträge erstellt werden. Formel 9 ermittelt die durchschnittliche Anzahl der Arbeitsplanpositionen AP je $V_{Endprodukt}$ über die gesamte Erzeugnisstruktur. Auf deren Basis lassen sich sowohl die zu erwartende Rüstzeit je Maschinengruppe $t_{Rüsten}(m)$ (Formel 10) und die erwartende Bearbeitungszeit je Maschinengruppe $t_{Bearbeiten}(m)$ (Formel 11) bestimmen. Mittels Formel 12 wird anhand des vorhandenen Kapazitätsangebots und der Verwendungshäufigkeit der minimale θ -normierte Durchsatz d über alle Maschinengruppen ermittelt. Das θ entspricht der verwendeten Zeiteinheit. Nach der Ermittlung des minimalen Durchsatzes d , wird in Formel 13, unter Berücksichtigung der angestrebten Auslastung u , ein λ für eine exponentialverteilte Zwischenankunftszeit für Kundenaufträge bestimmt.

$$AP = \frac{MV \cdot V \setminus V_{Endprodukt} \cdot w_{a,b} \cdot |M|}{V_{Endprodukt}} \quad (9)$$

$$t_{Rüsten}(m) = \min\left(\frac{AP}{w(m)}; \frac{AP}{|M|}\right) \cdot \vartheta(m) \quad (10)$$

$$t_{Bearbeiten}(m) = \frac{AP}{|M|} \cdot \mu(m) \quad (11)$$

$$d = \min_{m \in M} \frac{\theta \cdot r(m)}{t_{Rüsten}(m) + t_{Bearbeiten}(m)} \quad (12)$$

$$\lambda = \frac{1}{\theta \cdot \frac{d}{u}} \quad (13)$$

4 Validierung der generierten Testdaten

Für die Validierung der Testdaten wurden 450 Testdatensätze generiert. Dabei wurde für jede Strukturtiefe zwischen 2 und 10 eine Menge von Endprodukten erstellt. Die Menge der erzeugten Endprodukte wurde dabei ebenfalls stufenweise von 1 auf 50 erhöht. Die erzeugten Strukturen wurden nachfolgend analysiert und mit den Eingaben (= Erwartungswerten) hinsichtlich Mehrfachverwendungsgrad, Komplexitätsgrad und Organisationsgrad verglichen. Abbildung 2 zeigt die absolute Abweichung der erstellten Testdaten von den Vorgaben. Es ist gut zu erkennen, dass ab einer Strukturtiefe von 4 und einer Anzahl von 10 generierten Endprodukten die Abweichungen vom vorgegebenen Komplexitätsgrad und Mehrfachverwendungsgrad unter 5 % liegen. Damit die Abweichungen vom vorgegebenen Organisationsgrad ebenfalls gering ausfallen, benötigt man allerdings tiefere Produktstrukturen und/oder mehr Produkte.

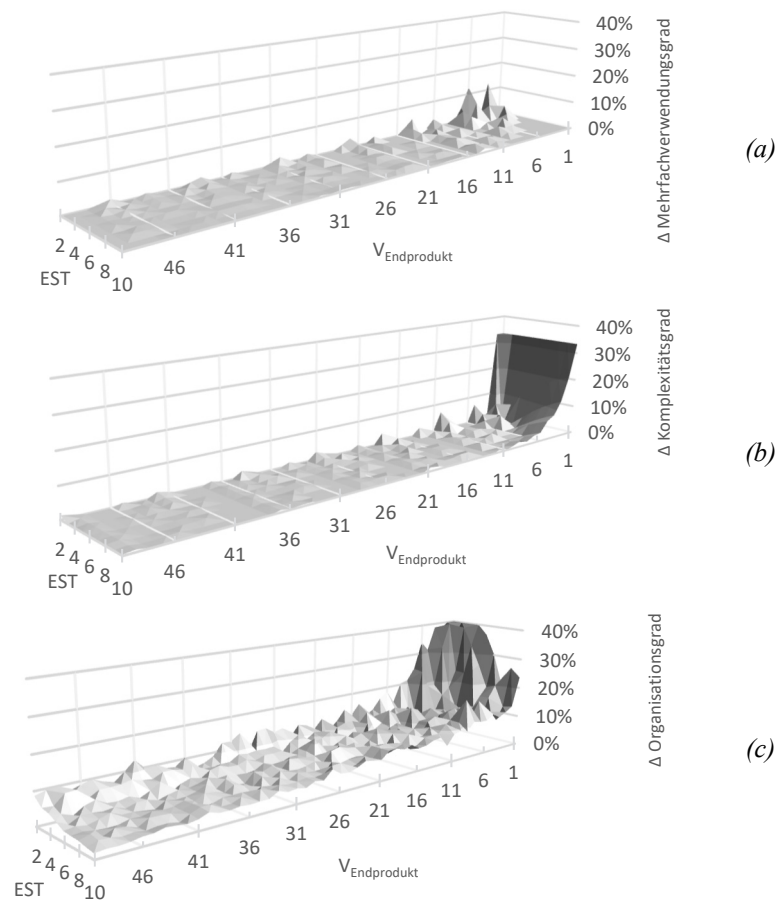


Abbildung 2: Abweichung der strukturbeschreibenden Kenngrößen

5 Zusammenfassung und Ausblick

Ziel des hier vorgestellten Generators ist das Generieren von Testdaten für die simulative Validierung von Produktionsplanung und -steuerung. Die generierten Testdaten entsprechen in ihren strukturbeschreibenden Kenngrößen bestimmten Vorgaben. Dabei können die vorzugebenden strukturbeschreibenden Kenngrößen durch Analyse realer Produktionsdaten ermittelt oder zur experimentellen Erprobung verschiedener Produkt- und Produktionsstrukturen manuell gesetzt werden. Durch die Verwendung gleicher Startwerte für die Zufallszahlengenerierung können identische Testdaten erzeugt und auf beliebige Zielsysteme übertragen werden. Für eine große Anzahl von Kombinationen aus strukturbeschreibenden Kenngrößen sind Testdaten generiert worden. Anhand dieser Testdaten wurden die Vorgehensweise bei der Testdatengenerierung evaluiert und die Grenzen des Testdatengenerators aufgezeigt, ab denen er Testdaten generiert, die in ihren strukturbeschreibenden Kenngrößen den Vorgaben entsprechen. Das vorgestellte Vorgehen bietet Raum für Erweiterungen. Beispielsweise könnten Verwendungshäufigkeit von Werkzeugen mit anderen

Verteilungen berücksichtigt werden. Auch ist es denkbar, die Einflüsse eines Produktmix bei der Generierung der Kundenaufträge zu berücksichtigen.

Danksagung

Das Forschungsvorhaben wird im Rahmen des Förderprogramms „Forschung an Fachhochschulen“ durch das Bundesministerium für Bildung und Forschung gefördert (Zeichen: 13FH133PX8).

Literatur

- Adolphy, S.; Grosser, H.; Kirsch, L.; Stark, R.: Method for Automated Structuring of Product Data and its Applications. *Procedia CIRP* 38 (2015), S. 153–158.
- Beasley, J.E., 1990: OR-Library. Online verfügbar unter <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>, zuletzt aktualisiert am February 2018, zuletzt geprüft am 29.04.2021.
- Corsten, H.; Gössinger, R.: Production management (Produktionswirtschaft): Introduction to industrial production management (Einführung in das industrielle Produktionsmanagement). München: Oldenbourg 2012.
- D. Libes; D. Lechevalier; S. Jain: Issues in synthetic data generation for advanced manufacturing. In: 2017 IEEE International Conference on Big Data (Big Data), 2017, S. 1746–1754.
- Fernandes, E.C.; Santos, L.I.d.; Camatti, J.A.; Brown, L.; Borsato, M.: Flexible Production Data Generator for Manufacturing Companies. *Procedia Manufacturing* 51 (2020), S. 1478–1484.
- FINRA DataGenerator, 2016: The DataGenerator. Online verfügbar unter <https://finraos.github.io/DataGenerator/>, zuletzt aktualisiert am 12.05.2016, zuletzt geprüft am 04.05.2021.
- Heinrich, C.E.: Mehrstufige Losgrößenplanung in hierarchisch strukturierten Produktionsplanungssystemen. Berlin, Heidelberg: Springer Berlin Heidelberg 1987.
- Mockaroo, 2021: Mockaroo: Random Data Generator and API Mocking Tool | JSON / CSV / SQL / Excel. Online verfügbar unter <https://finraos.github.io/DataGenerator/>, zuletzt aktualisiert am 03.05.2021, zuletzt geprüft am 03.05.2021.
- Sven Völker; Thomas Döring; Torsten Munkelt: The Generation of Large Test Data for the Empirical Analysis of Heuristic Procedures for Production Planning and Control. In: Bernhard Fleischmann; Rainer Lasch; Ulrich Derigs; Wolfgang Domschke; Ulrich Rieder (Hrsg.): *Operations Research Proceedings 2000. Selected Papers of the Symposium on Operations Research (OR 2000)*, Dresden, September 9-12, 2000, 2001, S. 266–270.
- Thomas Döring; Torsten Munkelt; Sven Völker: Generierung komplexer Testdaten zur statistischen Analyse von Verfahren der Produktionsplanung und -steuerung. In: Martin Bösel (Hrsg.): *Amtliche und Nichtamtliche Statistiken - 12. Ilmenauer Wirtschaftsforum, Tagungsband*, 1999, S. 34–46.