technische universität dortmund

IT in Produktion und Logistik

Literaturrecherche zur Komplexitätsklassentheorie in der Produktion und Logistik

Fachwissenschaftliche Projektarbeit

Vorgelegt von:	Dennis Daniel Kania
Matrikelnummer:	205518
Studiengang:	Maschinenbau

Ausgabedatum:	02.05.2024
Abgabedatum:	17.09.2024

Erstprüfer:	Dr.-Ing. Anne Antonia Scheidler
Zweitprüfer:	Katharina Langenbach M. Sc.

Technische Universität Dortmund

Fakultät Maschinenbau

Fachgebiet IT in Produktion und Logistik

Inhaltsverzeichnis

Abbildungsverzeichnis I
Tabellenverzeichnis II
Formelverzeichnis III
Abkürzungsverzeichnis IV
1 Einleitung 1
2 Grundlagen der Komplexitätsklassentheorie 2
2.1 Turingmaschinen 2
2.2 Algorithmen 5
2.3 Komplexitätsklassen 7
2.4 Produktion und Logistik 11
3 Vorgehensweise bei der und Ergebnisse der Literaturrecherche 13
3.1 Die Literaturrecherche 13
3.2 Analyse der Schlüsselaspekte 24
3.3 Vorstellung und Ergebnisanalyse 30
3.4 Diskussion und Fazit 33
4 Zusammenfassung und Ausblick 35
Literaturverzeichnis 36
Eidesstattliche Versicherung 48

Abbildungsverzeichnis

Abbildung 1: Turingmaschine mit der Sequenz 1011q701111 (in Anlehnung an Sipser 2005, S. 141) 3

Abbildung 2: Nichtdeterministische Turingmaschine (in Anlehnung an Sipser 2005, S.141) .. 3

Abbildung 3: Selection Sort, Programmcode in C++ (in Anlehnung an Rabe 2023, S. 13)..... 7

Abbildung 4: Hierarchien einiger Komplexitätsklassen (in Anlehnung an Gurari 1989, S. 318) 9

Abbildung 5: Klassifikation der Entscheidungsprobleme (in Anlehnung an Gurari 1989, S. 257) 10

Abbildung 6: Vorgehen bei der Literaturrecherche (in Anlehnung an vom Brocke et al. 2009, S. 7) 13

Tabellenverzeichnis

Tabelle 1: Zeitkomplexitätsklassen in der O-Notation (in Anlehnung an Hou 2021, S. 196) ... 8
Tabelle 2: Vergleich von Polynomial- und Exponentialzeit (in Anlehnung an Bovet et al. 1994, S. 51) ... 9
Tabelle 3: Taxonomie für den Scope (in Anlehnung an vom Brocke et al. 2015, S. 214) ... 13
Tabelle 4: Schlüsselbegriffe für die Literatursuche ... 15
Tabelle 5: Verwendete Suchstrings ... 16
Tabelle 6: Verwendete Datenbanken ... 16
Tabelle 7: Ergebnisse der Datenbankrecherche (in Anlehnung an Blanke 2023, S. 32-33) ... 22
Tabelle 8: Schlüsselaspekte zur Ermittlung der relevantesten Publikationen ... 23
Tabelle 9: Bewertungsmatrix (in Anlehnung an Gerdt 2023, S. 26) ... 24
Tabelle 10: Matrix zur Bewertung der Publikationen (in Anlehnung an Gerdt 2023, S. 48) ... 28
Tabelle 11: Statistiktabelle zur Auswertung der Matrix für alle 54 Publikationen ... 29
Tabelle 12: Gegenüberstellung der ausgewählten Publikationen ... 32

Formelverzeichnis

Formel 1: O-Notation 4
Formel 2: Polynomfunktion 4
Formel 3: o-Notation 4
Formel 4: Relativer Fehler 6
Formel 5: Stabilität eines Algorithmus 6
Formel 6: Rückwärtsstabilität eines Algorithmus 6
Formel 7: Boolean Formel 11

Abkürzungsverzeichnis

BPP	Bounded-error Probabilistic Polynomial Time
EXPTIME	Exponentialzeit
LAP	Location Allocation Problem
MAS	Multi-Agenten-System
MCMP2S	multi-commodity, multi-period, two-stage model
MRP	Manufacturing Ressource Planning
NP	nichtdeterministische Polynomialzeit
P	deterministische Polynomialzeit
PSPACE	polynomialer Speicherplatz Traveling Salesman Problem
SCMP2S	single-commodity, multi-period, two-stage model
SCMP2SOC	single-commodity, multi-period, two-stage open/closed model
SCMP3S	single-commodity, multi-period, three-stage model
TSP	Traveling Salesman Problem

1 Einleitung

Die zunehmende Digitalisierung in Unternehmen ermöglicht eine effizientere Gestaltung von Produktion und Logistik. Sie bietet eine Möglichkeit zur Optimierung der Organisation, gestaltet Abläufe effizient und vernetzt Betriebe untereinander (Helmold 2024). Die meisten Unternehmen fordern Präzision und Genauigkeit in den Berechnungen zu ihrer Entwicklung ein und versuchen, diese Parameter abzubilden und die Leistung zu steigern (Prabu et al. 2021). Diese Leistungssteigerung wird durch eine Optimierung der Arbeitsschritte ermöglicht. Um eine Verbesserung im digitalen Sinne umzusetzen, bedarf es eines Computers bzw. Rechners. Da ein Rechner lediglich mit Berechnungen arbeiten kann, braucht es mathematische Konzepte, die ihm Vorgaben liefern. Mit Algorithmen werden diese Vorgaben geliefert. Es gibt eine Vielzahl verschiedener Algorithmen. Sie werden, aufbauend auf ihren Ressourcen, durch Berechnungen in Komplexitätsklassen eingeteilt. Die Konzepte dieser Algorithmen sowie die berechenbare Komplexität sind elementare Komponenten in der theoretischen Informatik (Mich und Berry 2019). Algorithmen haben nicht nur einen theoretischen Hintergrund, sondern werden auch in der Praxis angewendet. So soll eine auftretende Problemstellung gelöst werden. Das bedeutet, es muss erst einmal ein Algorithmus existieren, der die Problemstellung lösen kann. Die Existenz reicht aber dahingehend nicht immer aus, um die praktische Lösung eines Problems zu garantieren. Hromkovič (2014) nennt den Grund wie folgt: „Man stellte fest, dass es beliebig schwere Aufgaben gibt, sogar solche, für deren Lösung man mehr Energie braucht, als im ganzen bekannten Universum zur Verfügung steht.“ Es entsteht folgende Frage: Wie schwer sind konkrete algorithmische Aufgaben? (Hromkovič 2014). Die theoretische Informatik und die Mathematik beschäftigen sich dazu mit der Komplexitätsklassentheorie. Die Kernaufgabe der Komplexitätsklassentheorie ist es nicht, einen einsatzfähigen Algorithmus zu entwerfen, vielmehr ermittelt sie den minimalen Ressourcenbedarf (Wegener 2003). Durch eingesparte Zeit könnte somit auch Geld innerhalb eines Produktionsablaufes eingespart werden. Für die Unterteilung von Komplexitätsgraden werden sogenannte Komplexitätsklassen entwickelt: Zur Einteilung eines Problems in eine Klasse lassen sich beispielsweise Ressourcen wie die Laufzeit oder der Speicherbedarf eines Algorithmus untersuchen, solange der Eingang der Daten identisch bleibt (Mich und Berry 2019).

Die Informatik verbindet Einflüsse aus verschiedenen Disziplinen (Hromkovič 2014). Dadurch entsteht ein großer Anwendungsraum. Im Alltag finden mit Hilfe der Informatik programmierte Geräte Anwendung. Es begann mit dem Einsatz von Computern in der Buchführung und im Ingenieurswesen. Heute reicht die Technologie so weit, dass Finanztransaktionen weltweit digital vorgenommen werden (Riley und Hunt 2014). Um präzisere Aussagen tätigen zu können und einen besseren Einblick zu gewährleisten, wird die Domäne eingegrenzt. Daher wird für die nachfolgenden Betrachtungen das Themengebiet auf die Produktion und Logistik beschränkt.

Ziel dieser Arbeit ist, Anwendungen zu finden, die in Komplexitätsklassen kategorisiert sind. So soll herausgefunden werden, ob die Komplexitätsklassentheorie ein rein theoretisches Konstrukt darstellt oder auch in der Praxis Verwendung findet und ggf. in welchem Umfang diese erfolgt. Für die Vorgehensweise ist eine systematische Literaturrecherche gewählt. Die Literaturrecherche soll Fallbeispiele in der Produktion und Logistik herausstellen und einen Bezug zu den Komplexitätsklassen aufzeigen. Dazu werden die in der Literatur verwendeten Fallbeispiele vorgestellt und diskutiert, um erkennbar werden zu lassen, welche Komplexitätsklassen besonderen Wert in der Produktion und Logistik erhalten. Das hat zum Ziel, die Relevanz der Komplexitätsklassentheorie für die Praxis zu ermitteln. Um die Komplexitätsklassen und ihre Theorie zu verstehen, werden zunächst Grundlagen geschaffen. Algorithmen werden erläutert und ihre Bedeutungen werden illustriert. Es wird auf den Aspekt der Stabilität eingegangen und ein Beispielalgorithmus vorgestellt. Algorithmen sind neben der Komplexitätsklassentheorie essenziel für die Informatik, da sie zusammen das realistische Ausführbare und das nicht effizient Ausführbare trennen (Wegener 2003). Weiterhin wird erklärt, was die Kom-

plexitätsklassentheorie beinhaltet und welchen Beitrag sie leistet. Für eine gute Übersicht werden die wichtigsten Komplexitätsklassen erläutert und in einem sinnvollen Zusammenhang aufgelistet. So entsteht ein Überblick über das komplexe Themengebiet. Zum Verständnis werden Turingmaschinen als theoretisches Konstrukt vorgestellt. Sie zeigen auf, wie Optimierungsprobleme in Komplexitätsklassen kategorisiert werden. Dafür werden sowohl die O-Notation und Begriffe wie Polynomialzeit, Determinismus, Nichtdeterminismus sowie die Ressourcen Zeit und Speicher erläutert. Dies ist notwendig, um die Komplexitätsklassen untereinander zu kategorisieren. Für ein besseres Verständnis werden die Begriffe Produktion und Logistik – da es um diese Anwendungsbereiche geht – näher analysiert und mit einer Definition von anderen Domänen abgegrenzt.

2 Grundlagen der Komplexitätsklassentheorie

Dieses Kapitel liefert eine Basis zum Verständnis der Komplexitätsklassentheorie. Bedeutende Konzepte zur Entwicklung der Komplexitätsklassentheorie und Grundbegriffe werden erläutert. Um eine Verknüpfung der Theorie zur Praxis herzustellen, werden die in dieser Arbeit diskutierten Anwendungsbereiche Produktion und Logistik näher betrachtet und erläutert.

2.1 Turingmaschinen

Entwickelt vom theoretischen Mathematiker Alan Turing (Strawn 2014), beschreibt die Turingmaschine ein simples Modell (Hopcroft 2012). Ein Modell ist in diesem Kontext ein Gedankenkonstrukt, welches physisch vorstellbar ist und zur Identifizierung von berechenbaren Funktionen eingesetzt wird (Hopcroft 2012). Es definiert somit die Bedeutung von Berechenbarkeit (Strawn 2014; Hopcroft 2012). Turings Modell hat sich für die Allgemeinheit von Mathematikern und Forschern durchgesetzt, da es eine mechanische Maschine symbolisiert (Hopcroft 2012). Das macht die Vorstellung des Konzeptes einfacher und wird in Abbildung 1 dargestellt. Die Turingmaschine benötigt einen Speicher, der in diesem Fall ein unbegrenztes langes Band mit Zellen darstellt (Strawn 2014). Auf ihm wird die Anfangsvorschrift aufgetragen und gelesen (ebd.). Das bedeutet, es wird ein lesender sowie schreibender Kopf benötigt (ebd.). Der Kopf liest die erste Anweisung, und abhängig vom Initialstatus wird die gelesene Anweisung überschrieben (Hopcroft 2012). Nun wandert der Kopf eine Zelle nach links oder rechts und liest die nächste Zelle aus (ebd.). Der Wert der gelesenen Zelle übernimmt den neuen Initialstatus (ebd.), und die Sequenz beginnt von neuem. Dies geschieht fortwährend, bis die Vorschrift vollständig ausgeführt ist und folglich anhält. Die Sequenz ist endlich, und die Anzahl der Schritte beträgt einen weniger als die Anzahl der Konfigurationen (Cook 2000). Sind die Ausführungsschritte nicht endlich, so gilt der Vorgang als unendlich (Cook 2000). Unendlichkeit tritt ein, wenn die Turingmaschine weder einen akzeptablen noch einen inakzeptablen Zustand erreicht (Sipser 2005). Das heißt, die Sequenz kann ausgeführt werden oder nicht. Folglich muss eine Schleife oder eine unendlich lange Abfolge an Befehlen der Grund sein. Insgesamt sind die Zustände der Turingmaschine der Startzustand (wenn die Sequenz beginnt) sowie der Haltezustand, der sich in den zuvor besagten akzeptierenden und nicht akzeptierenden Zustand aufteilt (Sipser 2005).

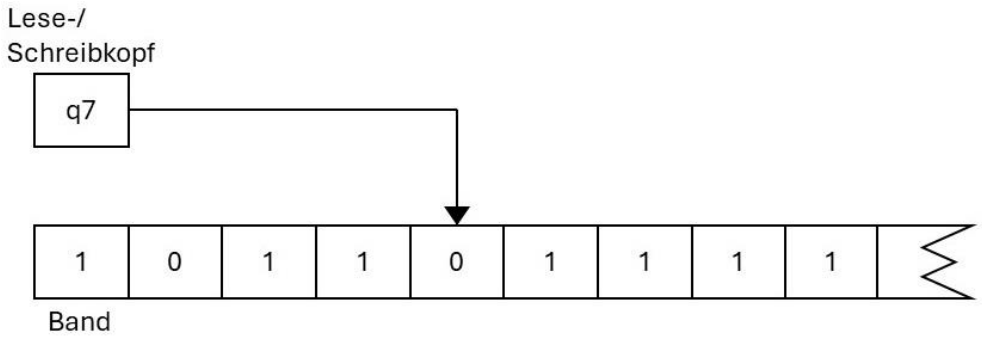


Abbildung 1: Turingmaschine mit der Sequenz 1011q701111 (in Anlehnung an Sipser 2005, S. 141)

Die sogenannte universelle Turingmaschine ist eine Version, welche das Verhalten einer anderen Turingmaschine wiedergibt (Pereira 2012). Somit stellt sie das erste Prinzip eines heutigen Computers dar (ebd.), denn auch der Computer übernimmt die getätigten Eingaben und verwertet diese innerhalb seiner Programme. Er überprüft also, ob ein geschriebenes Programm funktionsfähig ist. Der Unterschied zwischen diesem theoretischen Konzept und einem realen Computer ist der unendliche Speicher, den das Band aufnimmt. Ein Computer hat nur einen hinreichend großen Speicher und es wird auch kein Band verwendet.

Die simple Funktionsweise der Turingmaschine gewährleistet die Möglichkeit zu ihrer Modifikation. Für die Betrachtung von Computern wird z. B. die Multitape Turingmaschine verwendet, da sie in idealisierter Weise digitalen Rechnern am nächsten kommt (Hartmanis und Stearns 1965). Die bereits beschriebenen Varianten und Vorgehensweisen der Turingmaschinen beziehen sich auf ihre deterministischen Varianten. Das bedeutet, auf eine Eingabe erfolgt genau eine Ausgabe (Yamada 1962). Ein weiteres theoretisch nützliches Konzept beinhaltet die nichtdeterministische Turingmaschine. Sie ist nichtdeterministisch, wenn es eine endliche Anzahl von Möglichkeiten gibt, um auf den Initialzustand zu reagieren (Savitch 1970). Das Konzept ist wie ein binärer Baum vorzustellen, welcher in Abbildung 2 gezeigt wird. Um einen akzeptierenden Zustand zu erhalten, wird lediglich eine mögliche endliche Folge in der nichtdeterministischen Turingmaschine verlangt (Savitch 1970). Das heißt vereinfacht, es können mehrere Folgen zum Ziel führen oder auch nur eine. Diese Turingmaschine geht entlang des binären Baumes immer den richtigen Weg und ist aus diesem Grund sehr gut im ‚Raten', um den akzeptierenden Zustand zu erhalten (Papadimitriou 2003). Da die nichtdeterministische Turingmaschine ansonsten keine andere Funktionsweise aufweist, existiert als Pendant zu ihr immer auch eine äquivalente deterministische Turingmaschine (Sipser 2005).

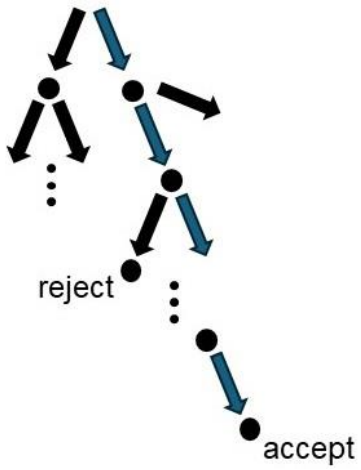


Abbildung 2: Nichtdeterministische Turingmaschine (in Anlehnung an Sipser 2005, S.141)

Zum Vergleichen der verwendeten Sequenzen in der Turingmaschine sind spezielle Notationen eingeführt worden, um z. B. bestimmen zu können, wie lange eine Sequenz benötigt, um

bearbeitet zu werden. Hier liegt der Kern der Komplexitätsklassentheorie. Nach Hartmanis und Stearns (1965) lautet die Definition der Komplexitätsklassentheorie: „The Computational complexity of a sequence is to be measured by how fast a multitape Turing machine can print out the terms of the sequence." Aus dieser Aussage geht bereits die Wichtigkeit der Zeit hervor. Innerhalb der Literatur lassen sich vor allem drei gängige Notationen erkennen. Die Omikron- (O), die Omega- (Ω) und die Theta-Notation (Θ). Alle drei Varianten haben ihre Daseinsberechtigung und werden für verschiedene Zwecke verwendet. Nach Knuth (1975) lassen sich die drei Notationsvarianten wie folgt verstehen: Die O-Notation ist das Worst-Case-Szenario. Die Ω-Notation hingegen beschreibt das Best-Case-Szenario. Und die Θ-Notation soll die exakte Komplexität ausdrücken (Knuth 1976). Nach Cormen et al. (2022) gilt für die O-Notation folgende mathematische Formel:

(1) $O\text{-Notation: } O(g(n)) = \{f(n): \text{Es existieren positive Konstanten } c \text{ und } n_0, \text{ so dass } 0 \leq f(n) \leq cg(n) \text{ für alle } n \geq n_0\}.$

Die am weitesten verbreitete und anerkannte Notation ist die O-Notation (Knuth 1976). Cormen et al. (2022) gibt für die Verwendung der O-Notation drei ausschlaggebende Gründe an. Erstens: Die maximale Laufzeit ist für praktische Anwendungen interessant, da sie Deadlines überschreiten kann. Zweitens tritt der Worst-Case oft ein. Und drittens ist der exakte Fall (die Θ-Notation) meistens ähnlich anspruchsvoll wie der Worst-Case. (Cormen et al. 2022, 31–32) Aus diesem Grund wird in dieser Arbeit weitestgehend auf die O-Notation eingegangen. Obgleich die Darstellung der Ω-Notation auch einen interessanten Faktor darstellt, ist die Berechnung des Best-Case-Szenarios oftmals die Mühe nicht Wert (Cormen et al. 2022). Um die O-Notation besser verstehen zu können, wird eine Funktion

(2) $$f(n) = dn^3 + cn^2 - bn + a$$

beschrieben. Die Koeffizienten a, b, c und d stehen für Konstanten. Die Konstanten haben einen Einfluss auf die Laufzeit in praktischen Anwendungen. Jedoch wird eine solche Funktion meist ohne die Konstanten dargestellt. Dies dient der Vereinfachung. Der Term mit der höchsten Potenz dieser Funktion lautet dn^3. Der Einfluss der Konstanten wird vernachlässigt, da die Funktion maximal asymptotisch n^3 ist; man schreibt $f(n) = O(n^3)$ und erhält so die O-Notation (Sipser 2005). Da die O-Notation die obere Schranke festlegt, ist das $O(n^3)$ auch in $O(n^4)$ enthalten (Cormen et al. 2022). Generalisiert gesagt, wird die gegebene Funktion wie folgt definiert $f(n) = O(n^c)$, wobei für die Konstante $c \geq 3$ gilt (ebd.). Sipser (2005) führt aus, dass eine Funktion, die eine niedrigere Potenz besitzt als angegeben, mit einem kleinen o gekennzeichnet wird. Cormen et al. (2022) erklärt die o-Notation wie folgt:

(3) $o\text{-Notation: } o(g(n)) = \{f(n): \text{Für jede positive Konstante } c > 0 \text{ existiert eine Konstante } n_0 > 0, \text{ so dass } 0 \leq f(n) < cg(n) \text{ für alle } n \geq n_0\},$

und führt das Beispiel $2n = o(n^2)$, aber $2n^2 \neq o(n^2)$ zur Verdeutlichung an.

In der Literatur lassen sich vorwiegend die Zeit und der Speicher als Ressource finden, welche eine Turingmaschine für ihre vollständige Sequenzierung benötigt. Sipser (2005) erklärt: „Time and space are two of the most important considerations when we seek practical solutions to many computational problems." Aus diesem Grund liegt der Fokus in dieser Arbeit auf diesen beiden Ressourcen. Laut Pereira (2012) erlangen die Begriffe Zeit und Speicher bereits durch Turing an Bedeutung: „discreteness of time and discreteness of state of mind. A Turing machine thus embodies the relationship between an unbound array of symbols in space, and a sequence of events in time, regulated by a finite number of mental states." Zeit und Speicher werden somit bereits initial als ausschlaggebend erachtet. Dabei bestimmt die Zeitressource die Anzahl der benötigten Schritte einer Turingmaschine (Papadimitriou 2003). Die Speicherressource bezieht sich auf die benötigte Anzahl der Zellen auf dem Band einer Turingmaschine (Papadimitriou 2003). Diese Definition der Speicherressource ist in der Literatur weit verbreitet,

sollte jedoch kritisch betrachtet werden, denn eine Turingmaschine, welche in einer Zelle wiederholt das Symbol verändert, würde somit keinen neuen Speicher beanspruchen (Bovet et al. 1994). Um eine modifizierte Definition der Speicherressource aufzustellen, gehen Bovet et al. (1994) wie folgt vor: Die Speicherressource bleibt undefiniert, sobald der Lesekopf der Turingmaschine nicht hält. Das heißt, erst wenn der Lesekopf hält, kann auch eine Aussage zum Speicher getätigt werden. Um nun entscheiden zu können, wann der Lesekopf hält, benötigt es eine Grenze der zu verwendenden Zellen (Bovet et al. 1994). Sonst kann nicht entschieden werden, wann der Lesekopf hält (ebd.). Die beiden Ressourcen Zeit und Speicher sind sinnvoll für eine praktikable Anwendung, um das Warten auf ein Ergebnis zu vermeiden, welches nahezu unendlich lange für seine Lösung benötigt. Yamada (1962) schreibt: „In using digital computers, it is important to know the time required to compute a given function." Auf der anderen Seite ist der Speicher in unseren Computern begrenzt, so dass dieser sparsam eingesetzt werden sollte. Reischuk (1999) betont: „[…] daß eine Speicherplatzressource mindestens so wertvoll ist wie eine Einheit Rechenzeit." Reischuk (1999) beschreibt dies konkreter: „[…] für allgemeine Maschinenmodelle (ist) die Platzressource um mindestens einen logarithmischen Faktor leistungsfähiger […]." Die beiden Ressourcen werden in unterschiedliche Komplexitätsklassen eingestuft, um zu bestimmen, ob und wie gut ein Problem lösbar ist (Sipser 2005). Dabei wird auf die Zeit und den Speicher einzeln eingegangen.

Für die Einteilung in die Komplexitätsklassen sind die polynomielle Zeit und der polynomielle Raum ein wichtiges Maß. Polynome sind Funktionen, welche Werte mit Exponenten beinhalten. Diese Werte können mit plus und minus miteinander verbunden sein. Ein Beispiel für eine polynomielle Funktion stellt die Gleichung (2) dar. Ein besonderer Fall tritt auf, sobald der Exponent den Wert 0 einnimmt. Es entsteht eine lineare Funktion, die eine eigene Gruppe bildet, aber in Analogie zum Konzept der O-Notation in der Gruppe der Polynome inbegriffen ist. Den Schwerpunkt in Forschungen auf die Polynomialzeit zu legen entstand in den 1960er Jahren durch die Erfahrung von Wissenschaftlern, die versuchten schwer zu optimierende Probleme zu lösen (Papadimitriou 2003). Die Polynomialzeit wird dabei von der Exponentialzeit unterschieden (Sipser 2005). Bei einer exponentiellen Funktion handelt es sich um eine Zahl, welche einen variablen Exponenten besitzt. Ein Beispiel für eine exponentielle Funktion lautet z. B.: $h(x) = 2^n$. Dies ist für die meisten Anwendungen jedoch nicht sinnvoll, da exponentielle Funktionen zu schnell ansteigen und so leicht eine Zahl erreichen können, die größer ist als die Anzahl der Atome im Universum (Sipser 2005).

2.2 Algorithmen

Für weitere Betrachtungen wird der Begriff Algorithmus definiert. Die bislang bei den Turingmaschinen betrachteten Sequenzen werden Algorithmen genannt. Algorithmen dienen dazu, einer Maschine – hier einer Turingmaschine oder im praktischen Fall einem Computer – Befehle zu übermitteln. Die Eingaben werden anhand des Algorithmus in Ausgaben umgewandelt (Cormen et al. 2022). Unabhängig vom Computer wird der Algorithmus als Verfahren angesehen, um ein Problem zu lösen (Sedgewick 2002). Algorithmen werden in Problemklassen zusammengefasst (Hromkovič 2014). Für die Komplexitätsklassentheorie ist hier die Problemklasse der Entscheidungsprobleme (ebd.) von Bedeutung. Algorithmen bestimmen innerhalb des Entscheidungsproblems, ob ein Problem lösbar (positiv) oder unlösbar (negativ) ist (Davis 1958). Somit beantwortet der Algorithmus im Kontext des Entscheidungsproblems eine „Ja- oder Nein-Frage" (Zelewski 1988).

Sipser (2005) leitet damit ein, dass David Hilbert im Jahr 1900 bereits 23 verschiedene mathematische Probleme veröffentlicht hat, die als Herausforderung für das (zu dieser Zeit) kommende Jahrhundert galten. Das zehnte Problem Hilberts beschreibt ein algorithmisches Problem, welches eine Definition für Algorithmen liefern soll, die es bis zu dieser Zeit nicht offiziell gegeben hat. Es wird daher ein Polynom gefordert, bei dem die Koeffizienten ganzzahlig so gewählt sind, dass die Quadratwurzel eine Null ergibt. Im Rahmen der Publikation wird bewie-

sen, dass ein solcher Algorithmus nicht existiert. Beweisverfahren haben Alan Turing und Alonzo Church vorgestellt. Die von Alan Turing konzipierte Turingmaschine aus Kapitel 2.1 und die Lambda-Infinitesimalrechnung von Alonzo Church haben sich als äquivalent herausgestellt. Aus diesen Thesen ist die sogenannte Church-Turing-These hervorgegangen. (Sipser 2005, 154–155)

Die genannte These verbindet Churchs Überlegungen mit der Turingmaschine. Zelewski (1988) definiert die Church-Turing-These wie folgt: „Wenn sich eine Funktion überhaupt […] berechnen läßt, dann muß es sich um eine rekursive Funktion handeln." Die rekursiven Funktionen hat bereits Turing betrachtet, und sie stellen die von ihm in der Turingmaschine behandelten Funktionen dar. Die Church-Turing-These bildet gleichzeitig als einziges Axiom in der theoretischen Informatik den Grundpfeiler dieser Wissenschaft (Hromkovič 2014).

Damit ein Algorithmus anpassungsfähig bleibt, werden ihm Eingaben übermittelt, und es werden Ausgaben aus ihm gewonnen. Die Stabilität eines Algorithmus beschreibt dabei die Fähigkeit, wie gut er mit Rundungsfehlern und Abweichungen umgehen kann. Je stabiler der Algorithmus, desto robuster reagiert er auf Fehler. Trefethen (1997) unterscheidet zusätzlich den Begriff der Stabilität von dem Begriff der Genauigkeit. Die Genauigkeit ist mithilfe von Formeln mit der Stabilität verknüpft. Trefethen (1997) gibt Formeln für die relativen Fehler einer Berechnung an:

(4) $$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O(\epsilon_{machine}).$$

$f(x)$ steht für die Formeln des gestellten Problems, wobei $\tilde{f}(x)$ den Algorithmus darstellt, welcher $f(x)$ abbilden soll. Je genauer nun der Algorithmus ist, desto kleiner wird die Abweichung sein, die von der gegebenen Maschine ϵ ausgeführt wird. Trefethen (1997) erläutert, dass ein schlecht gestelltes Problem $f(x)$ maximale Auswirkungen auf einen gut dargestellten Algorithmus $\tilde{f}(x)$ von $f(x)$ mit sich bringt. Um negative Auswirkungen zu vermeiden, wird daher die Stabilität herangezogen, die eine abgeänderte Form der Formel (4) aufweist:

(5) $$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\epsilon_{machine}).$$

Somit wird durch einen stabilen Algorithmus näherungsweise eine richtige Lösung für ein näherungsweise richtig gestelltes Problem erreicht. (Trefethen 1997, 103–104)

Trefethen (1997) gibt eine besondere Art der Stabilität an: die Rückwärtsstabilität. Sie ist simpler und stärker als die Stabilität und findet vermehrt Gebrauch in der numerischen linearen Algebra. Ein Algorithmus $\tilde{f}(x)$ nennt sich rückwärtsstabil für $\tilde{f}(x) = f(\tilde{x})$ für ein $\tilde{x}$ mit

(6) $$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\epsilon_{machine}).$$

Somit gibt ein rückwärtsstabiler Algorithmus nun die genaue Lösung für ein näherungsweise gestelltes Problem an. (Trefethen 1997, 104)

In der Literatur wird hier vermehrt auf die Komplexität der Sortieralgorithmen eingegangen, die in praktischen Anwendungen eingesetzt werden können. Denn die Bereitstellung geordneter Datensätze kann dazu beitragen einen Ablauf zu optimieren. Aus diesem Grund wird im Folgenden näher auf die Sortieralgorithmen eingegangen. Laut Sedgewick (2002) ist Selection Sort einer der simpelsten Sortieralgorithmen. Es wird das kleinste zu findende Element der zu sortierenden Gruppierung gesucht und mit dem Element, welches an erster Position steht, getauscht. Nun wird das zweitkleinste Element gesucht und mit dem Element, welches auf zweiter Position steht, getauscht. Das Prozedere fährt fort, bis die gesamte Gruppe sortiert ist. (Sedgewick 2002, 125)

Algorithmus 1: Selection Sort

```
#include <iostream>
using namespace std;

void selectionSort (int arr[], int n) {
   int i, j, min_index;
   for (i = 0; i < n-1; i++) {
      min_index = i;
      for (j = i+1; j < n; j++) {

          if (arr[j] < arr[min_index]) {
              min_index = j;
          }
      }
      swap(arr[min_index], arr[i]);

   }
}
```

Abbildung 3: Selection Sort, Programmcode in C++ (in Anlehnung an Rabe 2023, S. 13)

Offensichtlich muss das Verfahren Selection Sort bei n gegebenen Positionen im ungünstigsten Fall n-1 Austauschoperationen ausführen. Der damit vorzufindende ungünstigste Fall der Zeitkomplexität lautet somit $O(n^2)$ (Sedgewick 2002). Da kein zusätzlicher Speicherplatz benötigt wird (ebd.) lautet die Speicherplatzkomplexität $O(1)$. Die Verbindung der Algorithmen hinsichtlich ihrer Komplexität im zeit- und speicherplatzabhängigen Ressourcenverbrauch ist hieran erkennbar. Mich und Berry (2019) liefern mit ihrer Aussage eine eindeutige Definition für die Verbindung zwischen Algorithmen und der Komplexitätsklassentheorie:

„Comparing the efficiency of algorithms entails comparing algorithms by their consumption of space and time as they run on the same inputs. A characterization of the run-time space and time consumption of an algorithm as a mathematical function of the size of an input to the algorithm is called computational complexity of the algorithm."

Bovet et al. (1994) macht darauf aufmerksam, dass sich die in der Definition angesprochenen Ressourcen auf die Dynamik eines Algorithmus beziehen und sich in diese Untergruppe einordnen lassen. Die Statik hingegen bezieht sich beispielsweise auf die Programmgröße eines Algorithmus (ebd.) und wird in dieser Arbeit nicht weiter behandelt, da dieser Bereich der Optimierung von Algorithmen zuzuordnen ist.

Wie bereits am Beispiel von Selection Sort ersichtlich ist, wird die O-Notation zur Einteilung von Algorithmen verwendet und in Klassen eingeteilt. Im nächsten Abschnitt werden diese Klassen von Algorithmen aufgelistet und hinsichtlich ihrer Bedeutung für die bislang erarbeiteten Themen erläutert.

2.3 Komplexitätsklassen

Die Komplexitätsklassen werden allgemein gehalten, um ein Problem bestmöglich in seiner Komplexität abbilden zu können (Bovet et al. 1994). Aus diesem Grund werden Konstanten wie in den Gleichungen (1) bis (3) auch nicht berücksichtigt. Eine Konstante sorgt für eine große Abweichung in der Praxis, da sie in theoretischer Sichtweise nicht beschränkt ist. Einen großen Einfluss hat die Konstante dennoch in praktischen Anwendungsfällen und sollte deshalb dort beachtet werden. Für die Einteilung der Komplexitätsklassen ist sie unerheblich und wird aus diesem Grund nicht weiter betrachtet.

Eine Definition für Komplexitätsklassen geben Hartmanis und Stearns (1965) an, indem sie eine berechenbare Funktion T(n) definieren und ihr folgende Einschränkungen auferlegen. T(n) beschreibt eine Funktion aus den ganzen Zahlen, die in den Raum der ganzen Zahlen abbildet, sodass $T(n) \leq T(n+1)$ für eine Ganzzahl k mit $T(n) \geq \frac{n}{k}$ für alle n gilt. Eine Sequenz ist nun T-berechenbar, solange die Multitape Turingmaschine alle Sequenzen in Rahmen der T(n) Schritte auf ihr Band schreibt, während die Möglichkeit besteht, eine begrenzte Anzahl an Symbolen in eine Zelle zu drucken. So nennt sich die Klasse aller T-berechenbaren binären Sequenzen S_T. Bildet T(n) eine Zeitfunktion, dann ist der Name von S_T eine Komplexitätsklasse. (Hartmanis und Stearns 1965, 288)

Die Beziehungen zwischen den verwendeten Maschinenmodellen beeinflussen die Komplexitätsklassen bis auf Ausnahmen (Hartmanis und Stearns 1965), die hier nicht weiter behandelt werden, da sie unwesentlich für die Betrachtung der Komplexitätsklassen sind, nicht.

Tabelle 1 zeigt eine Übersicht der wichtigsten Zeitkomplexitätsklassen:

Notation	**Name**	**Beispiele**
$\boldsymbol{O(1)}$	Konstant	Popping on stack*
$\boldsymbol{O(logn)}$	Logarithmisch	Binäre Suche
$\boldsymbol{O(n)}$	Linear	Lineare Suche
$\boldsymbol{O(nlogn)}$	Linearithmisch	Merge Sort
$\boldsymbol{O(n^2)}$	Quadratisch	Bubble Sort
$\boldsymbol{O(n^3)}$	Kubisch	Einfache Matrixmultiplikation
$\boldsymbol{O(n^c)}$	Polynomiell	Lineare Programmierung
$\boldsymbol{O(c^n)}$	Exponentiell	Lösen des TSP** via dynamischer Programmierung
$\boldsymbol{O(n!)}$	Faktoriell	Lösen des TSP** via Brute Force
$\boldsymbol{O(2^{2^n})}$	Doppelt exponentiell	Wahrheit der Presburger-Arithmetik

Anmerkung. * Das Löschen eines Elements eines Stapels. ** TSP steht für Traveling Salesman Problem.

Tabelle 1: Zeitkomplexitätsklassen in der O-Notation (in Anlehnung an Hou 2021, S. 196)

Die in Tabelle 1 gezeigten oberen Schranken und ihre Namen haben zentrale Vertreter in den Komplexitätsklassen. Im Folgenden werden die wichtigsten Klassen besprochen.

Die Klasse BPP (Bounded-error Probabilistic Polynomial Time) beschreibt randomisierte Algorithmen mit maximal polynomialer Rechenzeit und einer Fehlerbeschränkung bezüglich jeder Eingabe von höchstens 1/3 (Wegener 2003). Mit Hilfe von Tabelle 1 ist eine Schranke von $O(n^c)$ erkennbar. Randomisierte bzw. probabilistische Algorithmen verwenden einen Nichtdeterminismus, der nicht immer richtig liegt. BPP-Algorithmen haben einen praktischen Nutzen, wenn wenige mögliche Ergebnisse ausgewählt werden können (Wegener 2003). Deterministische Algorithmen können einen großen Teil an Rechenzeit aufwenden oder durch falsche Probenauswahl zu Fehlanalysen führen (Sipser 2005). Somit ist ein besonderer Aspekt durch die Fehlerwahrscheinlichkeit in der BPP-Klasse gegeben. Da nicht zu einhundert Prozent ent-

schieden werden kann, ob die Lösung korrekt ist, entspricht BPP nicht den Entscheidungsproblemen (Wegener 2003). Aber BPP kann für Entscheidungsprobleme verwendet und eine Vielzahl anderer Probleme verwendet werden (Motwani und Raghavan 2007).

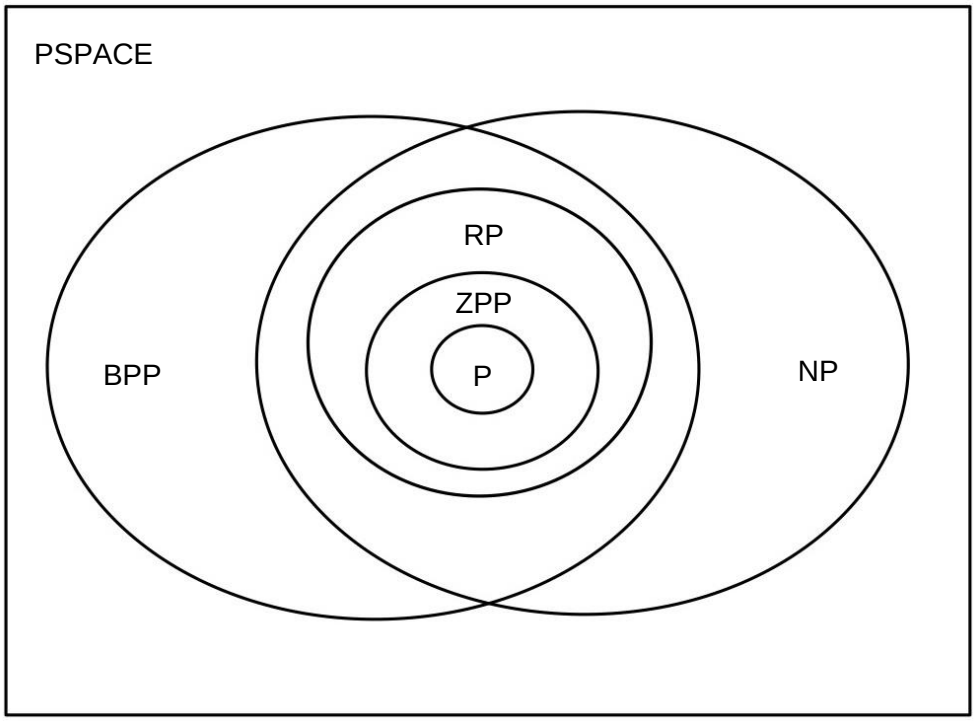


Abbildung 4: Hierarchien einiger Komplexitätsklassen (in Anlehnung an Gurari 1989, S. 318)

Die Klasse EXPTIME (Exponentialzeit) beinhaltet die am meisten in der Natur auftretenden Probleme in Form von Subexponentialzeit (Bovet et al. 1994). Tabelle 1 lässt eine obere Schranke von $O(c^n)$ erkennen. Zwangsweise beinhaltet EXPTIME alle anderen hier beschriebenen Klassen. Aus Abbildung 4 lässt sich der Zusammenhang der Klassen untereinander entnehmen. Es wird ein genereller Unterschied zwischen polynomiellen und exponentiellen Funktionen in der Komplexitätsklassentheorie gezogen (Sipser 2005). Das liegt am Zeitunterschied, welcher während der Berechnung entsteht (ebd.); dieser wird in Tabelle 2 dargestellt.

Instanz Größe n	**Algorithmus/Komplexität**				
	A_1/n^2	A_2/n^3	A_3/n^5	$A_4/2^n$	$A_5/3^n$
10	0,1 µs	1 µs	0,01 ms	1 µs	59 µs
30	0,9 µs	27 µs	24,3 ms	1 s	2,4 d
50	2,5 µs	0,125 ms	0,31 s	13 d	$2{,}3 \times 10^5$ Jahrzehnte

Tabelle 2: Vergleich von Polynomial- und Exponentialzeit (in Anlehnung an Bovet et al. 1994, S. 51)

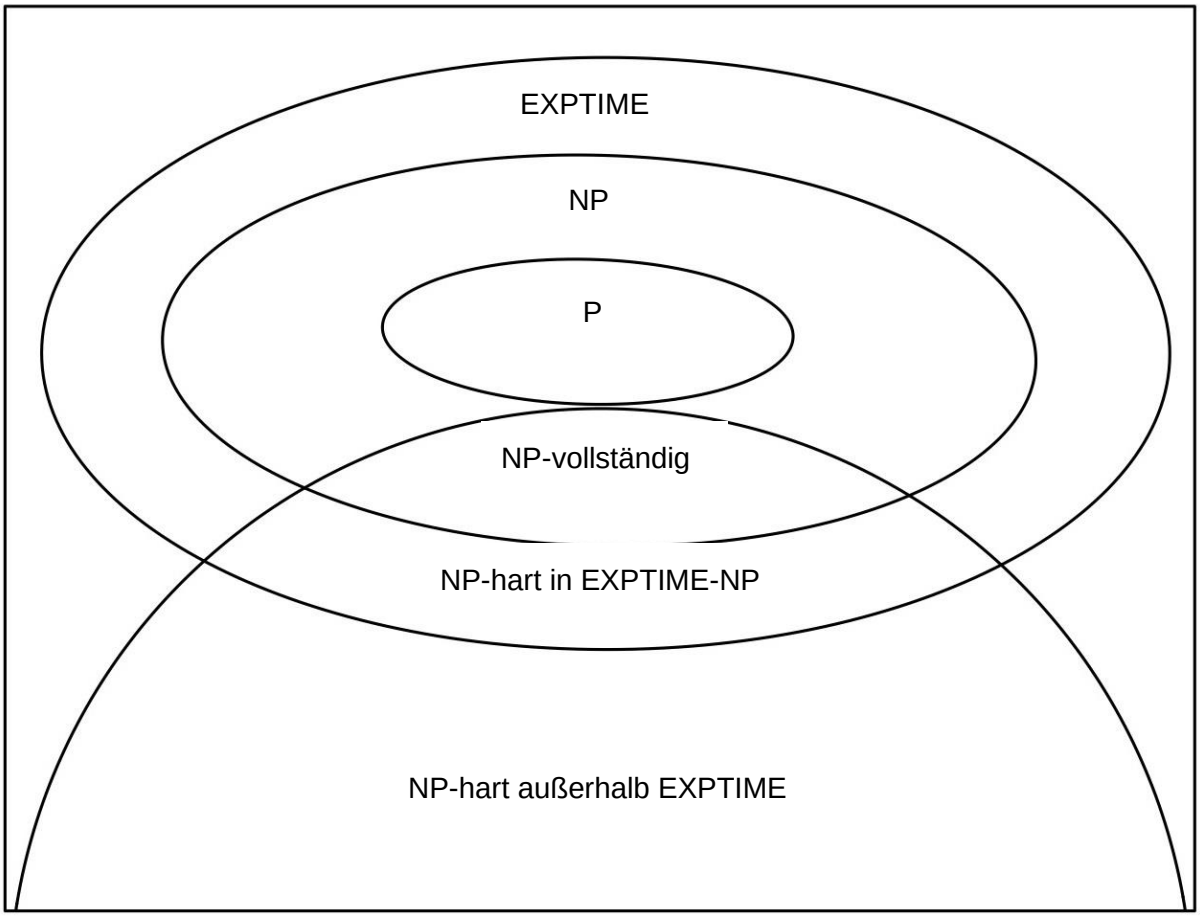


Abbildung 5: Klassifikation der Entscheidungsprobleme (in Anlehnung an Gurari 1989, S. 257)

Die Klasse PSPACE (polynomialer Speicherplatz) beinhaltet die zuvor dargelegte Klasse BPP (Gurari 1989). Sie enthält keine Zeitbegrenzung, sondern ist durch die Länge des Bandes limitiert, das von einem Polynom abhängig ist (Stockmeyer 1992).

Die Klasse P (Polynomialzeit) bezeichnet die wichtigste Komplexitätsklasse, die von Algorithmen innerhalb von Polynomialzeit berechnet werden kann (Papadimitriou 2003). Der Name spiegelt sich innerhalb der oberen Schranke mit $O(n^c)$ (siehe Tabelle 1) wider. In praktischen Anwendungen finden sich vermehrt n^3 oder n^5 Zeitalgorithmen, sodass die Klasse P zu den berechenbaren Entscheidungsproblemen zählt (Arora und Barak 2010). P ist zudem robust und ist deshalb unanfällig für Veränderungen bei der Wahl der Rechnermodelle sowie der gewählten Programmiersprachen (Stockmeyer 1992).

Die Klasse NP (nichtdeterministische Polynomialzeit) ist eine Erweiterung von P, in welcher deterministische durch nichtdeterministische Algorithmen ersetzt werden (Bovet et al. 1994). NP ist besonders interessant, da viele Anwendungen innerhalb dieser Klasse liegen (Sipser 2005). Wegener (2003) spricht von mehr als 3000 bekannten Problemen mit praktischem Interesse. Da die Klassen P und NP eine starke Ähnlichkeit aufweisen, stellt sich die Frage, ob P=NP ist. Tatsächlich stellt diese Fragestellung eines der bedeutendsten Probleme in der theoretischen Informatik wie auch der heutigen Mathematik dar (Sipser 2005). Es ist bis heute weder verifiziert noch widerlegt worden. Bovet et al. (1994) erläutert die Auswirkung, die es hätte, wenn P=NP bewiesen würde, denn dann gäbe es einen deterministischen Polynomialzeit-Algorithmus, welcher NP löst und eine geringe obere Grenze voraussetzt. Für P≠NP hingegen liefert Gurari (1989) folgende Prognose: „In particular, if it is discovered that *NP* is not equal to *P*, as is widely being conjectured, then *NP* is likely to provide some of the easiest problems (namely, the *NP*-complete problems) for proving the intractability of new problems by means of reducibility." Somit würde die Annahme neue Reduzierungen der Komplexitätsklassen ermöglichen. Alle Probleme in NP lassen sich jedoch auf das Erfüllbarkeitsproblem

(SAT) transformieren (Karp 1975). Diese Eigenschaft ist bekannt als Cooks Theorem (ebd.). Sipser (2005) erklärt das SAT-Problem anhand einer Boolean Formel. Gegeben sei eine Formel

(7) $$\phi = (\neg x \wedge y) v (x \wedge \neg z).$$

Ergibt Gleichung (7) den Wert True, ist das SAT-Problem gelöst. In diesem Fall lösen die Werte x = 0, y = 1 und z = 1 das Problem. (Sipser 2005, 271)

Die Klasse NP-hart beinhaltet Probleme, die in NP liegen und für die eine polynomielle Zeitreduktion existiert (Hou 2021). NP-Härte wird als Beweis dafür betrachtet, dass ein Problem nicht in Polynomialzeit dargestellt werden kann (Arora und Barak 2010). Ist ein Problem hart, so benötigt man superpolynomielle Zeit (Cormen et al. 2022). Diese Art von Problemen ist mindestens so schwer wie NP-vollständige Probleme (Garey und Johnson 1979). Garey und Johnson (1979) zeigen, dass NP-harte Probleme nicht in Polynomialzeit lösbar sind, es sei denn, es gilt die Annahme P=NP.

Die Klasse NP-vollständig ist praktisch lösbar unter der Annahme, dass P in NP liegt und die Ungleichheit beider Klassen anerkennt wird (Hromkovič 2014). Zudem muss das Problem NP-hart sein (Hou 2021), wie Abbildung 5 zu entnehmen ist. Cormen et al. (2022) merkt an, dass NP-vollständige Probleme zu den Entscheidungsproblemen gehören, aber auch einige Anwendungen in Optimierungsproblemen wiederzufinden sind. Optimierungsprobleme sind dabei Entscheidungsprobleme mit einer Einschränkung (ebd.).

2.4 Produktion und Logistik

Für die in dieser Arbeit zu untersuchenden Einsatzgebiete der Komplexitätsklassentheorie ist die Domäne der Produktion und Logistik gewählt. Zum besseren Verständnis und für die Verwendung im Rahmen dieser Arbeit werden sowohl Produktion als auch Logistik definiert.

Kummer (2019) erklärt Produktion als einen Wertsteigerungsprozess von Input-Einheiten hin zu Output-Einheiten. Input-Einheiten betreffen Material, Kunden, Geld oder Daten. Output-Einheiten hingegen umfassen Güter oder Dienstleistungen. An den Produktionsprozess werden die Anforderungen Zeit, Flexibilität, Wirtschaftlichkeit und Qualität gestellt. Um diesen Anforderungen gerecht zu werden und sie kontrollieren zu können, werden Prozesskennzahlen benötigt. Zu diesen Prozesskennzahlen gehören die Durchlaufzeit, die Output-Rate und der Bestand. Die Durchlaufzeit beinhaltet alle auftretenden Zeiten, um eine Input-Einheit in eine Output-Einheit umzuwandeln (Bearbeitungs- und Wartezeiten). Die Output-Rate entspricht der Menge an Produkten/Dienstleistungen pro Zeiteinheit. Der Bestand beschreibt die Menge der Produkte im Verhältnis zu seinem jeweiligen Arbeitsstand im Produktionsprozess. (Kummer 2019, 230–235)

In Werkstätten sind Bearbeitungszeiten von zehn Prozent der gesamten Durchlaufzeit charakteristisch (Hartmann und Kaiser 2005). Um derartige Prozesskennzahlen optimieren zu können, bedarf es einer Möglichkeit der Steuerung. Für diese Aufgaben steht die Produktionssteuerung.

Hartmann und Kaiser (2005) sprechen die Terminierung von Produkten an. Die vereinbarten Termine mit den Kunden sollen eingehalten werden. Dafür wird die Belegung der vorhandenen Maschinen geplant. Bei nur einer Maschine folgen hierfür intuitive Einordnungen der Produktaufträge. Die einzelnen Produktaufträge können nacheinander zu ihren festgelegten Terminierungen gefertigt werden. Bei mehreren Maschinen (bereits ab drei) entsteht bereits ein Problem, welches nicht mehr intuitiv effizient lösbar ist. Es werden Heuristiken (meistens Suchalgorithmen) angewendet, die auf Algorithmen basieren. (Hartmann und Kaiser 2005, 117–133)

Ein heutzutage oft verwendetes Programm zur Unterstützung des Produktionssteuerung ist das MRP II (Manufacturing Ressource Planning) (Hartmann und Kaiser 2005; Kummer 2019; Nebl 2007). Das MRP II verbindet rechengestützt Termin- und Ablaufplanungen (Nebl 2007). Um den Produktionsprozess sauber abzuschließen, gilt es, die Terminierung einzuhalten und den Kunden zu beliefern. Diese Belieferung wird mit dem Begriff der Logistik assoziiert. Als Leistung bzw. Prozess zur Versorgung des Heeres galt die Logistik – mithin aus dem militärischen Bereich stammend – lange Zeit als Kriegskunst (Kummer 2019). Der Begriff Logistik an sich findet in der Wirtschaft jedoch erst seit 1950 Gebrauch (Fleischmann 2018). Sie besitzt in der Literatur zahlreiche Definitionen und wird in den verschiedenen Werken je nach Einsatzgebiet anders beschrieben. Gemeinsamkeiten der Definitionen listet Fleischmann (2018) wie folgt auf: „Logistische Prozesse sind alle Transport- und Lagerprozesse sowie das zugehörige Be- und Entladen, Ein- und Auslagern (Umschlag) und das Kommissionieren." Dabei zu beachten ist, dass die Logistik Flüsse entwickelt und verwaltet (Göpfert 2019). Unter einem Fluss wird der Güterfluss verstanden. Die Güter können je nach Anwendungsgebiet verschiedene Objekte darstellen: „Sachgüter, Energie, Informationen oder Menschen" werden nach (Pfohl 2004) darunter verstanden. Die vierte industrielle Revolution (Industrie 4.0) findet u. a. in Bereichen der Logistik statt oder wird dort geplant. In der Logistik ist demnach ein nachhaltiger Wandel zu erwarten (Bauernhansl 2014). Autonome Fahrzeuge werden für den Transport von Ware verwendet, und zwischen den Transportmitteln entsteht ein eigener Austausch an Flüssen (Bauernhansl 2014). Da mehr Akteure vorhanden sind, die Flüsse austauschen, gibt es auch mehr Möglichkeiten, Aufgaben umzusetzen. Das erhöht wiederum die Komplexität der Aufgaben. Unter den Aufgaben der Logistik ist die Auslieferung eines Produktes in der richtigen Menge im richtigen Zustand zur richtigen Zeit an den richtigen Empfänger mit den richtigen Kosten zu verstehen (Pfohl 2004). In diesem Bereich der Logistik finden sich auch Entscheidungsprobleme und die Verwendung von Algorithmen wieder (Iskan und Obst). Logistik ist dabei nicht auf den Boden beschränkt und findet auch im Bereich der Luft und auf See Anwendung (Iskan und Obst).

Aus diesen Erkenntnissen wird folgende Definition für die Produktion und Logistik verwendet:

Die Produktion wird als Wertsteigerungsprozess angesehen. Sie beinhaltet nicht nur das Erschaffen oder Modifizieren eines neuen Produktes, sondern beschäftigt sich auch mit den Bearbeitungszeiten und somit mit dem Management der Produktion.

Die Logistik wird als Güterfluss angesehen und beinhaltet die Gestaltung sowie Terminierungen dieses Flusses. Dabei ist die Logistik nicht an den Transport und die Verteilung an Land gebunden. Sie kann auch in der Luft, im und auf dem Meer wie auch im Weltraum stattfinden.

3 Vorgehensweise bei der und Ergebnisse der Literaturrecherche

Dieser Abschnitt betrachtet die Vorgehensweise bei der Literaturrecherche. Die gewählte Methode wird erläutert und eingehend diskutiert. Die Suchstrings für die Literaturrecherche werden präsentiert und die Aufnahme der zu betrachtenden Publikationen anhand von Auswahlkriterien eingeschränkt. Die analysierte, aufbereitete Darstellung der Publikationen erfolgt im nächsten Schritt. Abschließend werden Gemeinsamkeiten und Unterschiede aufgezeigt, sodass ein Vergleich möglich ist. Ein Fazit rundet die Literaturrecherche ab.

3.1 Die Literaturrecherche

Der systematischen Literaturrecherche liegt die Vorgehensweise nach vom Brocke et al. (2009) zugrunde. Ziel des Vorgehens ist, die Vollständigkeit der Literaturrecherche widerzuspiegeln, sodass auch andere Forschende die Erkenntnisse als zuverlässig verwenden können (vom Brocke et al. 2009).

Das Vorgehen nach vom Brocke et al. (2009) gliedert sich in fünf Phasen. Für einen besseren Überblick sind die Phasen in Abbildung 6 dargestellt.

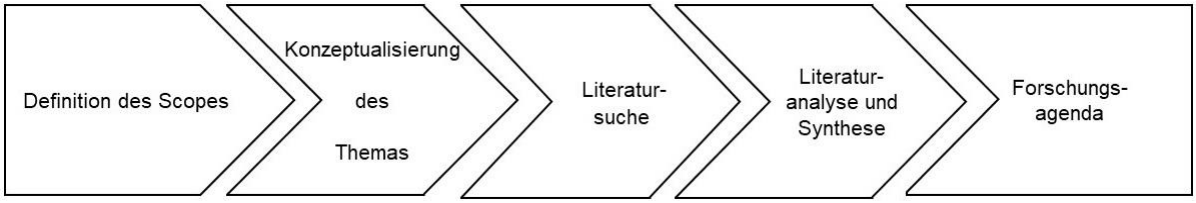


Abbildung 6: Vorgehen bei der Literaturrecherche (in Anlehnung an vom Brocke et al. 2009, S. 7)

Die erste Phase beschreibt die Definierung eines geeigneten Scope. Der Scope ordnet die Literaturrecherche in ihren Themenbereich ein und definiert ihren Umfang. Für eine Definition des Scope empfehlen vom Brocke et al. (2009) eine Taxonomie, präsentiert von Cooper (1988). Diese Taxonomie gliedert sich in sechs Unterpunkte und ist von vom Brocke et al. (2015) auf eine Taxonomie mit vier Punkten reduziert worden, welche in Tabelle 4 dargestellt ist.

1	**Prozess**	Sequenziell	Iterativ	
2	**Quelle**	Zitationsdaten-bank	Datenbanken	Publikationen
3	**Abdeckung**	Umfassend	Repräsentativ	Hauptwerke
4	**Techniken**	Schlüsselwort-suche	Vorwärtssuche	Rückwärtssuche

Tabelle 3: Taxonomie für den Scope (in Anlehnung an vom Brocke et al. 2015, S. 214)

Phase zwei verschafft einen Überblick über den aktuellen Wissensstand und Schlüsselthemen. Die dritte Phase umfasst den eigentlichen Suchprozess. Dieser beinhaltet die Recherche in Datenbanken, die Verwendung von Schlüsselwörtern und Rückwärts- und Vorwärtssuchen. Nach Webster und Watson (2002) betrifft die Rückwärtssuche die Artikel, welche im betrachteten Werk zitiert sind. Eine Vorwärtssuche hingegen ermöglicht es, Artikel einzusehen, welche das betrachtete Werk zitiert haben. Vom Brocke et al. (2009) empfehlen wissenschaftliche Journale oder Artikel aus renommierten Konferenzen als Quellen zu verwenden. Dabei sollte

sich auf hochrangige Artikel, also peer-reviewed Artikel mit großen Auswirkungen für den Wissenschaftsbereich, konzentriert und eine geeignete Datenbank mit spezifischen Suchbegriffen verwendet werden. Dadurch können irrelevante Artikel umgangen werden. In Phase vier wird die recherchierte Literatur analysiert und zusammengefasst. Phase fünf beinhaltet eine Forschungsagenda. Hier werden Fragen für zukünftige Forschungen formuliert; sie dienen somit als Denkanstoß. (vom Brocke et al. 2009, 1–9; Cooper 1988, 109; Webster und Watson 2002, xvi; vom Brocke et al. 2015, 209)

Für die Definition des Scopes wird die in Tabelle 3 gezeigte Taxonomie nach vom Brocke et al. (2015) verwendet. Die Taxonomie gliedert sich in vier Teilschritte auf, die dem Prozess, den Quellen, der Abdeckung und den Techniken gewidmet sind.

Prozess: Der Prozess wird anhand des Ziels der Literaturrecherche festgelegt. Es stehen zwei mögliche Herangehensweisen zur Verfügung: sequenziell oder iterativ. Der sequenzielle Prozess ist ein festgelegter Schritt am Anfang, welcher oftmals innerhalb der Recherche angepasst wird. Der iterative Prozess ist eine fortlaufende Suche und Analyse der Literatur. (vom Brocke et al. 2015, 214)

Quelle: Als Quellen können Zitationsdatenbanken, Datenbanken und Publikationen verwendet werden. Auch eine Kombination der genannten Quellen ist möglich. Erneut bestimmt das Ziel der Literaturrecherche die zu verwendenden Quellen. (vom Brocke et al. 2015, 214)

Abdeckung: Zur Abdeckung kann eine umfassende oder eine repräsentative Literaturrecherche, die sich auf die Hauptwerke (z. B. aus renommierten Zeitschriften) beschränkt, dienlich sein. Obgleich eine umfassende Literaturrecherche ideal klingen mag, so ist diese oft nicht praktikabel durchführbar, wenn eine zu große Anzahl von Veröffentlichungen existiert. In diesem Fall wird eine plausible Begründung für das Auslassen von Literatur notwendig. (vom Brocke et al. 2015, 214)

Techniken: Die gebräuchlichsten Techniken sind die Schlüsselwort-, Vorwärts- und Rückwärtssuche. Bei der Rückwärtssuche werden Quellen aus der Referenzierliste der gefundenen Literatur berücksichtigt. Mit der Vorwärtssuche hingegen wird auf Quellen geblickt, die die gefundene Literatur zitiert haben. (vom Brocke et al. 2015, 214)

Für die vorliegende Arbeit werden jeweils die blau hinterlegten Bereiche der Taxonomie aus Tabelle 3 verwendet. Es wird ein iterativer Prozess angewendet, welcher durch vordefinierte Schlüsselbegriffe eingegrenzt ist. Die Schlüsselbegriffe werden hierzu durch Synonyme und Unterthemen ergänzt. Es resultiert eine Tabelle mit der unterschiedlich starken Aussagekraft einzelner Schlüsselbegriffe. Tabelle 4 zeigt die für die Arbeit relevanten Schlüsselbegriffe. Um mehr relevante Literatur finden zu können, werden die Schlüsselwörter auf Englisch verwendet, da sich in der Forschung das Englische als weltweit verwendete Sprache für wissenschaftliche Artikel durchgesetzt hat.

Schlüsselwörter	Synonyme	Verwandte Wörter		
Computational Complexity	Complexity Theory	Combinatorial Optimization	Complexity Class	Calculability
Operations Research	Operational Research	/	/	/
Practice	Use	/	/	/
Turing Machine	/	Universal	Non-/Deterministic	Space/Time Complexity
Algorithm	/	Asymptotic Notation	Big O/Big Theta	Upper/Lower Bound
Production	/	/	/	/
Logistics	/	/	/	/

Tabelle 4: Schlüsselbegriffe für die Literatursuche

Aus den Schlüsselbegriffe aus Tabelle 4 werden Suchstrings erstellt. Über diese Suchstrings lässt sich in der Datenbank einschlägig relevante Literatur finden. Sobald ein Suchstring eingegeben und darüber Literatur angezeigt wird, werden die recherchierten Quellen als Treffer bezeichnet. Die Datenbanken besitzen die Möglichkeit, boolesche Operatoren zu verwenden, um eine gezieltere Suche zu ermöglichen.

Die booleschen Operatoren lauten AND, OR und NOT. Der Operator AND verknüpft zwei Schlüsselwörter und bildet die Schnittmenge zwischen den Schlüsselwörtern. Folglich werden Treffer die gewählten Schlüsselwörter aufweisen. Der Operator OR hingegen bildet die Vereinigungsmenge. Das heißt, es werden alle Treffer angezeigt, die mindestens eines der gewählten Schlüsselwörter beinhalten. Der Operator NOT exkludiert gezielt Schlüsselwörter. (Büttcher et al. 2016, 63)

Weitere Möglichkeiten der Eingrenzung bei der Suche nach Literatur bieten die Phrasensuche und die Trunkierung sowie die Verwendung von Klammern. Bei der Phrasensuche werden Anführungszeichen "..." verwendet. Zwischen ihnen steht das gewünschte Schlüsselwort. Über die Trunkierung werden ausschließlich Treffer gefunden, welche eine genaue Übereinstimmung mit der Suche aufweisen. Somit werden Wörter, die das gesuchte Schlüsselwort beinhalten, ausgeschlossen. Bei der Trunkierung wird der Asterisk * verwendet. Häufig werden auf diese Art Endungen eines Schlüsselwortes miteinbezogen. Der Asterisk steht für beliebige Zeichen mit unbestimmter Länge, welche in Verbindung mit dem angeführten Schlüsselwort stehen. Runde Klammern () gruppieren Wörter und ermöglichen dadurch, eine Reihenfolge, in welcher die anderen Operatoren angewendet werden sollen, vorzugeben. (Büttcher et al. 2016, 6, 64)

Aus den Schlüsselwörtern der Tabelle 4 und den booleschen Operatoren ergeben sich die Suchstrings, die für die Literaturrecherche dieser Arbeit verwendet worden sind. Tabelle 5 zeigt, welche Suchstrings dies sind.

Nummerierung	Suchstring
1	(“Computational Complexity “ OR “Combinatorial Optimization “) AND Practice
2	(“Computational Complexity “ OR “Combinatorial Optimization “) AND (“Operations Research “ OR “Operational Research “)
3	“Turing Machine “ AND Practice
4	“Computational Complexity “ AND Logistics
5	“Complexity Class “ AND (Use OR Practice)
6	“Turing Machine “ AND (“Computational Complexity “ OR „ Complexity Theory “) AND Algorithm
7	((Turing AND Calculability) OR “Complexity class* “) AND (production OR logistics)

Tabelle 5: Verwendete Suchstrings

Diese Suchstrings werden in den Datenbanken als Hilfsmittel zur geeigneten Literatursuche verwendet. Um eine möglichst große Auswahl an Quellen zu erreichen, werden verschiedene Datenbanken benutzt. Die hier gewählten Datenbanken lauten:

- OPAC der TU Dortmund
- IEEE Xplore
- Science Direct
- ACM Digital Library
- Scopus
- Web of Science

Bei der Benutzung der Datenbanken ist zu beachten, dass die Suchstrings in verschiedenen Kategorien angewendet werden können. So sind die Suchstrings in dieser Arbeit jeweils auf den Titel, die Schlüsselwörter und den Abstract angewendet worden. Bei den hier aufgelisteten Datenbanken ist zu berücksichtigen, dass der OPAC der TU Dortmund auch Treffer auf anderen Datenbanken ermöglicht und so eine große Anzahl an Quellen findet. Dies kann auch Tabelle 6 entnommen werden. Die Anzahl an Treffern, bezogen auf die angegebenen Suchstrings aus Tabelle 5, variiert stark.

Datenbank	Anzahl der Treffer aller Suchstrings
OPAC der TU Dortmund	3330749
IEEE Xplore	797
Science Direct	455
ACM Digital Library	102406
Scopus	9474
Web of Science	2909

Tabelle 6: Verwendete Datenbanken

Durch diese hohe Anzahl an Treffern ist es notwendig, Einschränkungen in der Suche vorzunehmen. Die Datenbanken bieten dafür verschiedene Möglichkeiten an, etwa den Zeitraum der Veröffentlichungen einzugrenzen oder die Literaturart auszuwählen. Bei der Suche nach passender Literatur ist aufgefallen, dass die Anzahl an Publikationen für die Suchstrings aus Tabelle 5 in den frühen 2000er Jahren anstieg. Daraus wird die Hypothese abgeleitet, dass die Forschung zur Komplexitätsklassentheorie für die Produktion und Logistik an Bedeutung gewinnt. Aus diesem Grund wird die gesuchte Literatur vorzugsweise in diesem Zeitraum ausgewählt. Damit wird nicht nur die Suche auf ein verwertbares Maß an Ergebnissen eingegrenzt, sie liefert auch aktuellere Quellen. Damit wird ein Wissensstand mit moderneren Einflüssen und Forschungstrends erarbeitet. Da aber die meisten Publikationen in diesem Zeitraum veröffentlicht worden sind, bleibt weiterhin eine große Anzahl an Quellen übrig, die sinnvoll selektiert werden müssen. Daher wird versucht, bei einer sehr großen Anzahl an Treffern das Veröffentlichungsdatum so aktuell wie möglich einzustellen. Für eine weitere Filterung der Suchergebnisse wird daher der Themenbereich auf die Ingenieurswissenschaften und die Informatik eingegrenzt. Diese Möglichkeit bietet z. B. die Datenbank Scopus. Derartige Einschränkungen machen die Recherche repräsentativ. Eine umfassende Literaturrecherche würde bedeuten, eine zu große Anzahl an Treffern zu berücksichtigen.

Nummer	Autor (Jahr)	Titel	Art der Publikation
1	Anacleto et al. (2021)	Fast r-flip move evaluations via closed-form formulae for Boolean quadratic programming problems with generalized upper bound constraints	Zeitschriftenartikel
2	Arkin et al. (1989)	Computational complexity of uncapacitated multi-echelon production planning problems	Zeitschriftenartikel
3	Boysen et al. (2019)	Matching supply and demand in a sharing economy: Classification, computational complexity, and application	Zeitschriftenartikel
4	Caliwag et al. (2019)	Bin Packing Problem with Compatible Categories Utilizing an Enhanced Variable Neighborhood Search Algorithm	Konferenzbeitrag
5	Caprara et al. (2014)	A Study on the Computational Complexity of the Bilevel Knapsack Problem	Zeitschriftenartikel
6	Çelik und Süral (2019)	Order picking in parallel-aisle warehouses with multiple blocks: complexity and a graph theory-based heuristic	Zeitschriftenartikel
7	Chen und Xu (2008)	A hybrid algorithm for multi-depot vehicle routing problem	Konferenzbeitrag
8	Chen et al. (2023)	Discrete Whale Optimization Algorithm for Capacitated Vehicle Routing Problem	Konferenzbeitrag
9	Coene et al. (2012)	On the Computational Complexity of Peer-to-Peer Satellite Refueling Strategies	Zeitschriftenartikel
10	Danancier et al. (2019)	Comparison of Path Planning Algorithms for an Unmanned Aerial Vehicle Deployment Under Threats	Zeitschriftenartikel
11	de Weerdt et al. (2018)	Complexity of Scheduling Charging in the Smart Grid	Konferenzbeitrag

12	Ding et al. (2021)	Implementation of a Hybrid Classical-Quantum Annealing Algorithm for Logistic Network Design	Zeitschriftenartikel
13	Faigle und Kern (1994)	Computational complexity of some maximum average weight problems with precedence constraints	Zeitschriftenartikel
14	Felinskas und Sakalauskas (2008)	Variable neighborhood Simulated Annealing method and application for design	Konferenzbeitrag
15	Fliedner et al. (2016)	Vehicle scheduling under the warehouse-on-wheels policy	Zeitschriftenartikel
16	Gadhikar et al. (2023)	Comparative Analysis of Algorithms for Cargo Container Space Optimization	Konferenzbeitrag
17	Ghelichi et al. (2022)	Drone logistics for uncertain demand of disaster-impacted populations	Zeitschriftenartikel
18	Gopalan (2015)	Computational complexity of convoy movement planning problems	Zeitschriftenartikel
19	Guo et al. (2019)	Parallel machine scheduling with job synchronization to enable efficient material flows in hub terminals	Zeitschriftenartikel
20	Hall et al. (1998)	Scheduling with finite capacity input buffers	Zeitschriftenartikel
21	Ivanova et al. (2021)	Methods for optimizing routes in digital logistics	Konferenzbeitrag
22	Kitamura und Okamoto (2012)	Automated Route Planning for Milk-run Transport Logistics Using Model Checking	Konferenzbeitrag
23	Keith et al. (2024)	Machine Learning-Based Problem Space Reduction in Stochastic Programming Models: An Application in Biofuel Supply Chain Network Design	Zeitschriftenartikel
24	Kolen und Kroon (1993)	On the computational complexity of (maximum) shift class scheduling	Zeitschriftenartikel

25	Kovalyov et al. (2004)	Batching decisions for assembly production systems	Zeitschriftenartikel
26	Lee und Qi (2009)	Vehicle scheduling with combinable delivery and pickup operations	Zeitschriftenartikel
27	Lee et al. (2013)	Operations scheduling with multiple resources and transportation considerations	Zeitschriftenartikel
28	Li (2015)	Container terminal logistics scheduling and decision-making within the conceptual framework of computational thinking	Konferenzbeitrag
29	Li et al. (2019)	Container Terminal Oriented Logistics Generalized Computational Complexity	Zeitschriftenartikel
30	Li et al. (2005)	Machine scheduling with deliveries to multiple customer locations	Zeitschriftenartikel
31	Li et al. (2021)	An Algorithm to Solve Heterogeneous Vehicle Routing Problem With Second Trip	Zeitschriftenartikel
32	Liu et al. (2023)	Route planning for last-mile deliveries using mobile parcel lockers: A hybrid q-learning network approach	Zeitschriftenartikel
33	Ma (2023)	Simulation of Logistics Route Optimization Model Based on Genetic Algorithm Optimization Neural Network	Konferenzbeitrag
34	Manzini und Gebennini (2008)	Optimization models for the dynamic facility location and allocation problem	Zeitschriftenartikel
35	Matsutani und Kimura (2022)	Delivery Routing to Reduce Calculation Load of Drones on Divided Logistics Areas for Drone Logistics Networks	Konferenzbeitrag
36	Monteiro et al. (2024)	Computational complexity and linear formulations for optimizing the location of shared mobility stations	Zeitschriftenartikel
37	Morim et al. (2024)	The drone-assisted vehicle routing problem with robot stations	Zeitschriftenartikel

38	Murray und Raj (2020)	The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones	Zeitschriftenartikel
39	Ng et al. (2010)	'Product Partition' and related problems of scheduling and systems reliability: computational complexity and approximation	Zeitschriftenartikel
40	Park et al. (2018)	Two-agent scheduling with generalized due dates	Zeitschriftenartikel
41	Rinaldi et al. (2023)	Development of Heuristic Approaches for Last-Mile Delivery TSP with a Truck and Multiple Drones	Zeitschriftenartikel
42	Rudek (2012)	Scheduling problems with position dependent job processing times: computational complexity results	Zeitschriftenartikel
43	Sarid und Shapiro (2008)	Classifying the multi robot path finding problem into a quadratic competitive complexity class	Zeitschriftenartikel
44	Tian et al. (2022)	Swarm Intelligence Technique for Supply Chain Market in Logistic Analytics Management	Zeitschriftenartikel
45	Vanajakumari et al. (2022)	Supply chain planning: A case for Hybrid Cross-Docks	Zeitschriftenartikel
46	Villar-Medina et al. (2009)	Implementation of a supervised learning technique in a multi-agent system for building production orders	Zeitschriftenartikel
47	Walter et al. (2013)	The discrete forward‐reserve problem ‐ Allocating space, selecting products, and area sizing in forward order	Zeitschriftenartikel
48	Peng Wang und Stephan Bohacek (2008)	On the practical complexity of solving the maximum weighted independent set problem for optimal scheduling in wireless networks	Konferenzbeitrag
49	Weyland (2014)	On the computational complexity of the Probabilistic Traveling Salesman Problem with Deadlines	Zeitschriftenartikel

50	Xie et al. (2021)	Introducing split orders and optimizing operational policies in robotic mobile fulfillment systems	Zeitschriftenartikel
51	Ying et al. (2016)	Exact algorithms for single-machine scheduling problems with a variable maintenance	Zeitschriftenartikel
52	Zachariadis und Kiranoudis (2010)	A strategy for reducing the computational complexity of local search-based methods for the vehicle routing problem	Zeitschriftenartikel
53	Zawadzki et al. (2013)	A generalization of SAT and #SAT for robust policy evaluation	Konferenzbeitrag
54	Zhang et al. (2021)	Optimal scheduling of emergency resources for major maritime oil spills considering time-varying demand and transportation networks	Zeitschriftenartikel

Tabelle 7: Ergebnisse der Datenbankrecherche (in Anlehnung an Blanke 2023, S. 32-33)

Als erstes werden die Titel aus Tabelle 7 überprüft. Diese sollen zwingend einen Bezug zum Thema der Komplexitätsklassentheorie, der Optimierung, der Produktion oder der Logistik aufweisen. Um keine relevanten Publikationen zu verpassen, werden bei dem Kriterium des Titels uneindeutige Titel toleriert, die im nächsten Schritt strenger begutachtet werden. Das zweite Kriterium bezieht sich auf den Abstract. Hier müssen eindeutige Merkmale von praktischen Anwendungen in der Produktion oder Logistik erkennbar sein. Weiterhin muss ein Bezug zur Optimierung oder zur Komplexitätsklassentheorie bestehen. Durch dieses Vorgehen und die Elimination der Duplikate können 60 Publikationen recherchiert werden, von denen 54 frei zugänglich sind. Tabelle 7 listet die übernommenen Publikationen auf. Dem Beispiel von vom Brocke et al. (2009) folgend, wird hier ein Augenmerk auf Artikel aus wissenschaftlichen Zeitschriften gelegt. Da die Publikationen bereits im Rahmen der Recherche geringfügig gefiltert worden sind, werden spezifischere Aspekte erstellt. Die Publikationen aus Tabelle 7 werden dahingehend unter Bezugnahme auf Schlüsselaspekte analysiert. Die Schlüsselaspekte begünstigen den Entscheidungsprozess zur Auswahl der Publikationen, welche diese Arbeit am besten unterstützen. Tabelle 8 zeigt die gewählten Schlüsselaspekte und erläutert ihre Bedeutung im Kontext der Publikationsanalyse.

Schlüsselaspekte	Erläuterung
Anwendung in der Praxis	Die Publikation soll sich auf in der Praxis anwendbare Fälle beziehen. Ist dies nicht der Fall, bezieht sich die Publikation auf theoretische Ansätze.
Domäne	Die Domäne zeigt, in welchen Bereichen sich die Anwendungen der Produktion und Logistik wiederfinden. So können womöglich Trends ermittelt werden.
Präsentation von Algorithmen	Es werden Algorithmen entwickelt oder es wird sich auf bereits bestehende Algorithmen gestützt.
Art der Komplexitätsklasse	Die gegebene Komplexitätsklasse wird aufgeführt, um die in der Praxis häufig auftretenden Komplexitätsklassen identifizieren zu können

Tabelle 8: Schlüsselaspekte zur Ermittlung der relevantesten Publikationen

Die Schlüsselaspekte sind dabei in ihrer Bedeutsamkeit unterschiedlich. Das liegt an den Unterschiedlichen Bezügen der Schlüsselaspekte zur Komplexitätsklassentheorie. Dadurch erreicht man eine genauere Übereinstimmung zur Auswahl der Publikationen. Die Anwendung in der Praxis hat einen hohen Stellenwert und ist ausschlaggebend für die Verwendung der Publikationen in dieser Arbeit. Die Domäne ist hierbei weniger wichtig, da sie keine Abhängigkeit zur Komplexitätsklassentheorie aufweist. Sie hat jedoch einen stärkeren Bezug zu der hier untersuchten Thematik, sofern Produktion oder Logistik erkennbar sind. Da Teilbereiche der Produktion und Logistik aber auch auf anderen Feldern auftreten können, werden auch andere Domänen mitberücksichtigt. Die Präsentation von Algorithmen ist ebenfalls weniger wichtig. Denn ein Algorithmus ist kein Indikator für die Komplexitätsklassentheorie, sondern wird auch in anderen Forschungsfeldern untersucht. Doch durch die dargelegten Algorithmen wird ein stärkerer Bezug zur Praxis erreicht, zumal diese oft mit Experimenten einhergehen. Da ein Versuch in der Praxis aber nicht von der Präsentation eines Algorithmus abhängig sein muss, wird diesem Schlüsselaspekt eine geringere Bedeutsamkeit zugeordnet. Die Art der Komplexitätsklasse soll eine Übersicht geben, welche Komplexitätsklassen vermehrt auftreten. Hierbei ist die Verwendung von Komplexitätsklassen essenziell. Die Wahl der Komplexitätsklasse ist für die reine Einordnung der Publikationen in dieser Arbeit unbedeutend. Durch Zuteilung von Bedeutsamkeit zu einzelnen Schlüsselaspekten können die passenden Publikationen leichter ermittelt werden. Das Fehlen von bedeutsamen Schlüsselaspekten verschlechtert die Bewertung einer Publikation, das Vorhandensein relevanter Schlüsselaspekte verbessert

diese. Da auch experimentelle Analysen in den Publikationen auftreten, können Kombinationen etwa mit der Domäne Logistik ein passenderes Match für diese Arbeit hervorbringen.

Um eine Bewertung zu erhalten und zu ermitteln, welche Publikationen am zutreffendsten sind, wird ein Bewertungsschema benötigt. Hierfür wird die von Gerdt (2023) zitierte Bewertungsmatrix aus dem Bericht der Bundesanstalt für Arbeitsschutz und Arbeitsmedizin (BAuA) verwendet.

Kategorie	Symbol	Beschreibung
A	++	Alle oder die meisten der Kriterien werden erfüllt. Es ist sehr unwahrscheinlich, dass die Kriterien, die nicht erfüllt werden, die Schlussfolgerungen der Studie ändern.
B	+	Einige der Kriterien werden erfüllt. Es ist unwahrscheinlich, dass die Kriterien, die nicht erfüllt werden, die Schlussfolgerungen der Studie ändern.
C	-	Wenige oder keine der Kriterien werden erfüllt. Es ist wahrscheinlich oder sehr wahrscheinlich, dass die Kriterien, die nicht erfüllt werden, die Schlussfolgerungen der Studie ändern.

Tabelle 9: Bewertungsmatrix (in Anlehnung an Gerdt 2023, S. 26)

Die Bewertungsmatrix besteht aus den drei Kategorien A, B und C. Kategorie A beinhaltet Publikationen, welche alle oder die meisten der aufgestellten Aspekte aus Tabelle 8 erfüllen. Zur Kennzeichnung wird ein “++“ verwendet. Kategorie B steht für Publikationen, die einige Kriterien erfüllen. Dementsprechend wird ein “+“ zur Kennzeichnung verwendet. Die Kategorie C inkludiert Publikationen, die wenige oder keine Kriterien erfüllen. Hierfür wird ein “-“ zur Kennzeichnung verwendet. (Gerdt 2023, 26)

3.2 Analyse der Schlüsselaspekte

Um die für die Fragestellung dieser Arbeit am besten geeigneten Publikationen ausfindig zu machen, werden nun die Schlüsselaspekte in den Publikationen untersucht. Dafür wird eine neue Matrix erstellt (Tabelle 10). Diese Matrix verbindet die Tabellen 7, 8 und 9. Es wird die Nummerierung der Publikationen aus Tabelle 7 verwendet. Korrelierend der Nummern kann somit aus Tabelle 7 entnommen werden um welchen Titel es sich hierunter handelt.

Schlüsselaspekte / Publikationen	Anwendung in der Praxis	Domäne	Präsentation von Algorithmen	Art der Komplexitätsklassen	Bewertung nach Bewertungsmatrix
1	Nein	Informatik, Mathematik	Ja	NP	-
2	Nein	Logistik	Nein	NP, NP-vollständig	-
3	Digitales Experiment	Verkehr	Nein	P, NP-hart	+
4	Nein	Informatik	Ja	BPP, NP-hart	-
5	Nein	Mathematik	Nein	P, NP-vollständig	-
6	Digitales Experiment	Logistik	Ja	NP-hart, NP-vollständig	+
7	Digitales Experiment	Logistik	Ja	NP-hart	+
8	Digitales Experiment	Logistik	Ja	NP-hart	+
9	Digitales Experiment	Raumfahrt	Ja	NP-hart	+
10	Digitales Experiment	Militär	Ja	Nicht angegeben	-
11	Digitales Experiment	Verkehr	Ja	P, NP-hart, NP-vollständig	+
12	Digitales Experiment	Logistik	Ja	NP-hart	+
13	Nein	Mathematik	Nein	NP-vollständig	-
14	Digitales Experiment	Produktion	Ja	NP, NP-vollständig	+
15	Realer Bezug	Logistik, Mathematik	Ja	NP-hart	+

16	Digitales Experiment	Logistik	Ja	Nicht angegeben	+
17	Digitales Experiment	Logistik, Luftfahrt	Ja	NP-hart	+
18	Nein	Militär	Ja	NP-vollständig	+
19	Digitales Experiment	Produktion, Logistik	Ja	NP-hart, NP-vollständig	+
20	Nein	Produktion	Ja	P, NP-vollständig	-
21	Digitales Experiment	Logistik	Ja	NP	+
22	Digitales Experiment	Logistik	Ja	P	+
23	Digitales Experiment	Logistik	Ja	Nicht angegeben	-
24	Nein	Luftfahrt	Ja	NP-hart, NP-vollständig	-
25	Nein	Produktion	Ja	NP-hart	+
26	Realer Bezug	Logistik	Ja	NP-hart	+
27	Nein	Logistik, Medizin	Ja	P, NP-hart	-
28	Digitales Experiment	Logistik	Ja	Nicht angegeben	-
29	Digitales Experiment	Logistik	Ja	NP-vollständig	+
30	Nein	Produktion, Logistik	Ja	NP-hart	+
31	Digitales Experiment	Logistik	Ja	NP-hart	+
32	Digitales Experiment	Logistik	Ja	NP-hart	+
33	Digitales Experiment	Logistik	Ja	Nicht angegeben	-
34	Ja	Logistik	Ja	NP-hart	++
35	Digitales Experiment	Logistik, Luftfahrt	Ja	P	+

36	Digitales Experiment mit realem Bezug	Verkehr	Ja	NP-hart	+
37	Digitales Experiment	Logistik, Luftfahrt	Ja	Nicht angegeben	-
38	Digitales Experiment	Logistik, Luftfahrt	Ja	NP-hart	+
39	Nein	Mathematik	Ja	NP-hart, NP-vollständig	-
40	Nein	Produktion, Logistik, Mathematik	Nein	NP-hart	-
41	Digitales Experiment	Logistik, Luftfahrt	Ja	NP-hart	+
42	Nein	Produktion	Ja	NP-hart	-
43	Digitales Experiment	Informatik, Mathematik	Ja	Nicht angegeben	-
44	Digitales Experiment	Logistik	Ja	Nicht angegeben	-
45	Digitales Experiment	Logistik	Ja	NP-hart	+
46	Ja	Logistik	Ja	NP-hart	++
47	Digitales Experiment	Logistik	Ja	NP-hart	+
48	Digitales Experiment	Kommunikation	Ja	NP-vollständig	+
49	Nein	Logistik, Mathematik	Nein	#P-hart	-
50	Digitales Experiment mit realem Bezug	Logistik	Ja	NP-hart	+
51	Digitales Experiment	Produktion	Ja	P	+
52	Digitales Experiment mit realem Bezug	Logistik	Ja	P	+

53	Nein	Mathematik	Ja	$\#P^{NP}$	-
54	Digitales Experiment mit realem Bezug	Logistik (maritim)	Ja	NP-hart	+

Tabelle 10: Matrix zur Bewertung der Publikationen (in Anlehnung an Gerdt 2023, S. 48)

Für diese Arbeit relevante Publikationen sind mit einem “++“ in Tabelle 10 gekennzeichnet. Wesentliche Merkmale für das Erreichen einer Übereinstimmung mit dem Forschungsthema der Komplexitätsklassentheorie in der Produktion und Logistik sind die Existenz einer Anwendung in der Praxis und die Einordnung in eine Komplexitätsklasse. Fehlt einer dieser Aspekte, wird die Publikation nicht mehr mit einem “++“ gekennzeichnet. Die Unterscheidung, ob ein “-“ oder “+“ verwendet wird, hängt vom Bezug zur Praxis sowie dem Anwendungsgebiet in der Forschung ab. So erreichen Publikationen mit praxisbezogenen Impulsen ein “+“ in der Auswertung, auch wenn keine selbstständige Anwendung erfolgt. In Tabelle 11 wird die Statistik der Auswertung aufgelistet und im Folgenden diskutiert.

Schlüsselaspekte	Merkmale	Häufigkeit
Anwendungen in der Praxis	Ja	5
	Nein	16
	Digitales Experiment	33
	Realer Bezug	6
Domäne	Informatik	3
	Kommunikation	1
	Logistik	33
	Luftfahrt	6
	Mathematik	9
	Medizin	1
	Militär	2
	Produktion	8
	Raumfahrt	1
	Verkehr	3
Präsentation von Algorithmen	Ja	48
	Nein	6
Art der Komplexitätsklasse	BPP	1
	NP	4
	NP-hart	30
	NP-vollständig	13
	P	7
	#P-hart	1
	$\#P^{NP}$	1
	Nicht angegeben	8

Tabelle 11: Statistiktabelle zur Auswertung der Matrix für alle 54 Publikationen

Nur wenige Publikationen befassen sich mit der selbstständigen Anwendung. Dafür wird eine Vielzahl von digitalen Experimenten verwendet. Aus der Literaturrecherche ist ersichtlich, dass die meisten der gefundenen Publikationen einen Bezug zur Praxis aufweisen oder Probleme aufgreifen, welche in der Praxis auffindbar oder anwendbar sind. Nur wenige beziehen sich

konkret auf einen in einem Unternehmen auftretenden Sachverhalt. Es wird häufig versucht, ein bereits bekanntes Problem zu vereinfachen, um auf diese Weise die Komplexitätsklasse zu verringern. Dadurch gelangen auch Publikationen ohne Praxisbezug in die Analyse, denn auch hier wird meistens versucht, einen Algorithmus zu vereinfachen. Sowohl praxisbezogene als auch theoretische Forschungen versuchen oft, die Zeitkomplexität zu senken. Zur besseren Veranschaulichung werden fast immer Algorithmen präsentiert. Entweder werden bekannte Algorithmen aufgegriffen oder neue definiert, die das Problem effizienter lösen sollen. Die am häufigsten verwendete Komplexitätsklasse ist NP-hart. Den Erwartungen entsprechend, werden in den aktuellen Publikationen vorzugsweise schwere Probleme in nichtdeterministischer Polynomialzeit erforscht. Am häufigsten wird die Domäne der Logistik untersucht. Trivialerweise erfolgt dies durch die Filterung nach produktionstechnischen und logistischen Publikationen. Jedoch lassen sich Mischformen der Logistik erkennen, da sie beispielweise zusammen mit der Luftfahrt auffindbar sind. Die Ursprünge des Militärs (Kummer 2019) stehen weniger im Fokus, wobei die Luftfahrt auch in diesem Kontext umfunktioniert werden könnte.

Somit bleiben drei Publikationen übrig, die im nächsten Abschnitt näher betrachtet und vorgestellt werden.

3.3 Vorstellung und Ergebnisanalyse

In diesem Abschnitt werden die Publikationen näher erläutert. Ihre Vorgehensweise wird dargestellt und die wichtigsten Punkte ihrer Forschungsergebnisse zusammengefasst. Anschließend wird eine Tabelle erstellt, die die Gemeinsamkeiten und Unterschiede zwischen den Publikationen darstellen soll. Es wird herausgestellt, welche Konzepte der Komplexitätsklassentheorie verwendet werden.

Optimization models for the dynamic facility location and allocation problem (Manzini und Gebennini 2008)

Manzini und Gebennini (2008) untersuchen das Facility Location und Allocation Problem. Hierbei wird über die geeignetsten Standorte von Firmen und Verteilungszentren, sowie deren Zuordnung von Waren, Aufträgen und Kunden entschieden. Es ist ein wichtiges Problem in der Logistik und gehört zu den NP-harten Problemen. Diese Publikation verfolgt die Zielsetzung, Fragen, die das Facility Location und Allocation Problem aufwirft, zu beantworten. Eine derartige Frage lautet z. B.: „Wo produziere ich am rentabelsten?“ Die Arbeit beschäftigt sich zuerst mit der Sichtung der Literatur. Im Anschluss werden vier Modelle aufgestellt, die auf das dynamische Location Allocation Problem (LAP) einer generischen, weltweit agierenden Firma angewendet werden. Abschließend werden die Ergebnisse einer Fallstudie erläutert. (Manzini und Gebennini 2008, 2061–2062)

Die Autoren geben die in der Literatur auffindbaren Varianten des LAPs wieder. Es existieren *das single facility minimum location problem*, welches den optimalen Ort für die Verteilung von Gütern unterstützt, und das *multiple facility location problem*, welches Kunden von mehreren Standorten aus beliefert. Auch hier sollen die optimalen Standorte für die Werke gefunden werden. Des Weiteren besteht das *facility location allocation problem*, bei dem Standorte für neue Werke zu den bereits bestehenden Werken lokalisiert und der Bedarf zwischen ihnen geplant werden soll. Schlussendlich besteht noch das *network location problem*, welches dem *facility location allocation problem* gleicht, bei dem jedoch der Güterfluss miteinbezogen wird, sodass ein zusätzliches Netzwerk zum Tragen kommt. Weitere Probleme werden der Vollständigkeit halber eingeführt, wobei es sich um Erweiterungen der bereits geschilderten Probleme handelt. (Manzini und Gebennini 2008, 2062–2064)

Die in der Arbeit verwendeten Modelle sind innovative linear-integer programming Modelle. Das erste vorgestellte Modell ist das *single-commodity, multi-period, two-stage model (SCMP2S).* Das Modell ist passend für ein Logistiknetzwerk, welches einen

Hauptproduktiosort, mehrere Unterwerke und eine Reihe von Kunden besitzt. Im Fokus stehen die Lieferketten zwischen den Werken untereinander und den Kunden,; diese Lieferketten sollen minimiert werden. Außerdem wird die Produktionszeit gemanaged. Unter der Annahme von konstanten Lagerkosten, dem Nichtbetrachten der Lagerkapazität und exakten Angebots- und Nachfrage-Voraussetzungen wird eine mathematische Formulierung entwickelt. Die Rechenkomplexität wird dabei minimiert, indem stochastische Ereignisse missachtet werden. Das zweite Modell ist das *multi-commodity, multi-period, two-stage model (MCMP2S)*. Es unterscheidet sich vom ersten Modell darin, dass mehr als ein Produkt produziert und angeboten wird. Modell drei, das *single-commodity, multi-period, two-stage open/closed model (SCMP2SOC)* beinhaltet die weitere Möglichkeit, dass die Werke zu verschiedenen Zeiten geschlossen oder geöffnet sein können. Das letzte Modell, das *single-commodity, multi-period, three-stage model (SCMP3S)* bezieht die Produktionsanlagen mit ein. (Manzini und Gebennini 2008, 2065–2069)

Die Fallstudie bezieht sich auf ein Unternehmen mit Hauptsitz in Italien, Subwerken in Frankreich, Deutschland, Taiwan, USA und dem Vereinigten Königreich sowie über 1100 Kunden. Es werden die geographischen Koordinaten der Kunden in kartesische Koordinaten umgewandelt und somit 50 bayzentrische Nachfragepunkte ermittelt, die in insgesamt 254 virtuelle Punkte umgewandelt werden. Nach Anwendung des SCMP2S stellt sich heraus, dass die beste Konstellation für den Güterfluss die Schließung der Werke in Frankreich, Deutschland und dem Vereinigten Königreich erfordert. Das bringt eine Ersparnis von 900.000 € pro Jahr mit sich. Weitere Anwendungen des SCMP2S auf andere Orte schlagen ein weiteres Werk in der Türkei vor. Für weitere Rationalisierungen wird daraufhin das MCMP2S angewendet und die Schließung und Wiederöffnung eines Werkes mit berücksichtigt. Es stellt sich heraus, dass das Werk in Deutschland nur zu den vier zentralen Monaten des Jahres geöffnet sein soll. (Manzini und Gebennini 2008, 2074–2086)

Implementation of a supervised learning technique in a multi-agent system for building production orders (Villar-Medina et al. 2009)

In dieser Publikation wird die Implementierung eines Algorithmus für die Erstellung von Fertigungsaufträgen beschrieben. Versucht wird dies mittels eines Multi-Agenten-Systems (MAS). Diese Art von Problemen gehört ebenfalls zur Klasse NP-hart. Die Verwendung von MAS ist in verschiedenen Formen bereits in Gebrauch und wird zur Optimierung verwendet. Neuronale Netze und Systeme zur Auftragsabwicklung werden ebenfalls konsultiert. Der Vorschlag von Villar-Medina et al. (2009) unterscheidet sich in drei Punkten zu den vorhandenen Forschungsarbeiten auf diesem Gebiet. Erstens wird ein agentbasierendes Produktionssystem angestrebt, zweitens ein überwachtes Lernen integriert und drittens eine Kommunikation asynchroner Entscheidungen gefordert. (Villar-Medina et al. 2009, 808–809)

Die Studie betrachtet einen übergeordnete Produktionsplan, mit dem durch zeitliche Auflösung eine quantitative Bedarfsermittlung prognostiziert werden soll. So kann ausgeglichen werden, was produziert werden soll und was die Kapazität zulässt. Entsteht nun eine Änderung, müssen zeitintensive neue Rechnungen bemüht werden. Agent-basierende Entscheidungen sollen genau das verhindern. (Villar-Medina et al. 2009, 809–812)

Es werden fünf verschiedene Agenten verwendet: Ein Agent für die Koordination (i), ein Agent für die Maschinen (ii), ein Agent für die Werkzeuge (iii), ein Agent als Beobachter (iv) und ein Agent als Planer (v). Sowohl (ii) als auch (iii) besitzen intelligente Fähigkeiten. (i) koordiniert die Mitteilungen zwischen den Agenten. (iv) untersucht in festen Intervallen das System. Anschließend entscheidet (v) die Reihenfolge der Fertigungsaufträge. Die Kommunikation zwischen (i) und (ii) erfolgt durch fünf Datensätze, jedoch erst nach Berichterstattung von (iii). (ii) berechnet nun, welche Maschine in Betrieb genommen werden soll, und informiert (i). Sobald die Daten für die Werkzeuge und die Maschine festgelegt sind, wird dem Fertigungsauftrag eine Priorität zugewiesen, und er wird abgespeichert. Als nächstes muss die Fertigungsabteilung die Produktion freigeben. (Villar-Medina et al. 2009, 813)

Auch hier wird eine Fallstudie umgesetzt. Die produzierende Firma stellt Etiketten für verschiedene Produkte her. Durch die vielen wählbaren Farben und Etikettenmaterialien entsteht eine hohe Anzahl an möglichen Produkten. Einhergehend steigt auch die Komplexität der Aufgabe, das korrekte Werkzeug an der richtigen Maschine zeitlich einzuteilen. Es wird dafür ein neuronales Netz verwendet, welches dem Agenten durch Java-Code hinzugefügt wird. Eine Trainingsmatrix unterstützt den Prozess des Lernens. (Villar-Medina et al. 2009, 813–817)

Gemeinsamkeiten	Unterschiede
Komplexitätsklasse NP-hart	Verwendung eines programming Modells/Agent-Algorithmus
Algorithmen werden entworfen	Untersuchung ähnlicher Problemstellungen/Arbeiten
Produktion und Logistik	Standortproblem/Terminplanungsproblem
Optimierung eines Problems	
Verwendung ähnlicher Ansätze	
Keine Analyse der Rechenkomplexität	

Tabelle 12: Gegenüberstellung der ausgewählten Publikationen

Für eine bessere Analyse werden in Tabelle 12 Gemeinsamkeiten und Unterschiede aufgelistet. Beide Publikationen weisen eine NP-harte Komplexitätsklasse auf, welche nicht näher analysiert wird. Sie wird lediglich angegeben und als bekannt vorausgesetzt. Analysen bezüglich der Rechenkomplexität treten nicht auf. Sowohl die Einbeziehung einer Turingmaschine als auch die mathematischen Beweismethoden für Komplexitätsklassen werden nicht angewendet. Trotz alledem werden Optimierungsprobleme in der Produktion und Logistik untersucht. Dafür werden in beiden Publikationen neue Algorithmen entworfen. Die Entwicklung dieser Algorithmen steht im Vordergrund beider Forschungsarbeiten und wird hinreichend getestet. In der Analyse ähnlicher Ansätze werden Unterschiede deutlich und die eigene Forschung wird gerechtfertigt. Ein Unterschied hierbei ist die Untersuchung ähnlicher Problemstellungen in (Manzini und Gebennini 2008) und ihre Bedeutung im Kontext der Forschung, wohingegen (Villar-Medina et al. 2009) ganze Forschungsarbeiten abgrenzt. Weitere Unterschiede entstehen bei der Verwendung der Arten von Algorithmen. Hier wird einmal ohne und einmal mit Agenten und damit einer eigenen Entscheidungsfindung des Systems gearbeitet. Schlussendlich werden auch andere Optimierungsprobleme untersucht, z. B. das LAP, welches sich auf die Standorte für die beste Produktion und Logistik bezieht, und die Terminplanung von Fertigungsaufträgen.

Da sich ein Großteil der Publikationen mit digitalen Experimenten beschäftigt wird auch eine dieser Forschungsarbeiten kurz präsentiert.

Optimal scheduling of emergency resources for major maritime oil spills considering time-varying demand and transportation networks (Zhang et al. 2021)

Zhang et al. (2021) untersuchen die Planung von Notfallressourcen bei Ölkatastrophen im Meer. Dieses Problem befindet sich ebenfalls in der Komplexitätsklasse NP-hart. Das Problem ist nicht nur relevant für die effiziente Ressourcennutzung und das Budget, sondern verhindert auch die Ausbreitung negativer ökologischer Folgen. Es wird innerhalb der Fachliteratur wenig zur Logistik nach Auftreten des Vorfalls gefunden. Weitere Gespräche mit der maritimen Verwaltung Chinas zeigen die unkoordinierte Vorghensweise während einiger Unfälle,

weswegen dieser Artikel die Lücke realitätsbezogen ergänzen soll. (Zhang et al. 2021, 529–530)

Zuerst werden drei landbasierte, notfallressourcenlogistische Ansätze betrachtet, darunter die Zeitplanung, welche das komplexe dynamische Problem in ein relativ statisches Problem umwandelt. Die rollende Planung aktualisiert die Parameter, die der dynamischen Natur eines landbasierten Vorfalls nahekommt. Zusätzlich werden probabilistische Szenarien für die Unsicherheit addressiert. Weiterhin in der Literatur vorzufinden sind Chemieunfälle und maritime Szenarien, die landbasierte Ansätze verwenden. Eine Analyse der Literatur gibt Auskunft über die unzureichende Forschungslage in der maritimen Notfallressourcenplanung. Ausschlaggebend ist die fehlende Betrachtung der Wechselbeziehungen von Katastrophenausbreitung und -eindämmung bei schnellem Einschreiten und damit dem Verhindern, dass sich ein Ölfilm ausbreiten kann. Schließlich wird eine eigenst entwickelte Methode zur Planung von Notfallressourcen vorgestellt. Sie beinhaltet einen neuen Planungsansatz zur optimierten Berücksichtigung dynamischer Vorgänge. Die Wechselbeziehungen zwischen der Ölfilmausbreitung und dem Notfallhandeln werden betrachtet. Ein effizienterer Algorithmus wird implementiert; Kosten-Leistungs-Verhältnisse lassen die beste Lösung ermitteln. (Zhang et al. 2021, 530–533)

Nach Aufstellung eines mathematischen Modells und der Annahmen für einen realgetreuen maritimen Notfallressourceneinsatz wird der Algorithmus anhand einer Fallstudie getestet. Die Ölkatastrophe ereignete sich in Bohai Bay, China. Für die Tests werden die genauen geographischen Koordinaten sowie die Ausbreitungsgeschwindigkeit des Öls und seine Masse miteinbezogen. Es stellt sich heraus, dass der neu aufgestellte Algorithmus einen älteren Algorithmus übertrifft. Weitere Tests mit zufällig generierten Szenarien ergeben eine Sensibilität der Zeitabhängigkeit. Somit besteht auch auf diesem Feld weiterer Forschungsbedarf. (Zhang et al. 2021, 534–545)

Der Bezug zur Praxis ist in der Publikation erkennbar, jedoch werden keine direkten Fallstudien vorgestellt.

3.4 Diskussion und Fazit

Abschließend ist zu sagen, dass wenig bis gar keine Theorien der Komplexitätsklassentheorie in der Praxis angewendet werden. Es ließen sich nur wenige Publikationen mit relevanten Fallstudien auffinden, weshalb eine genauere Evaluation erschwert ist. Die in Abschnitt 2.1 vorgestellten Turingmaschinen werden nicht berücksichtigt. Hingegen stellen die Forschungarbeiten die Algorithmen (Abschnitt 2.2) in den Vordergrund. Sie werden teilweise innerhalb digitaler Experimente untersucht. Praxisnah werden sie, indem real antreffende Strukturen eingefügt werden. Ein Beispiel dafür ist (Zachariadis und Kiranoudis 2010), wo vier griechische Städte als Ausgangssituation eines Logistiknetzes verwendet werden. Darüberhinaus wird versucht, die Rechenzeit zu minimieren (siehe auch Abschnitt 2.1.1). Es werden praktische Anwendungen von Optimierugsproblemen gennant, aber eine Anwendung erfolgt nicht. Bezüglich der Analyse von Komplexitätsklassen besteht in der Sammlung der gefundenen Publikationen Forschungsmaterial mit mathematischen Beweismethoden. Diese Beweismethoden widmen sich meistens der Theorie. Hier wird lediglich die Zugehörigkeit eines Problems in eine Komplexitätsklasse dargestellt. Anwendungsbezogene Aussagen werden in Bezug auf die Laufzeit und damit das Worst-Case-Szenario getätigt, etwa in (Kovalyov et al. 2004) (vgl. Abschnitt 2.1). Aus diesem Grund wird, wie in Abschnitt 2.1 erläutert und Abschnitt 2.3 dargestellt, die O-Notation verwendet, was sich bereits in der Literaturrecherche abzeichnete. Die Einteilung in die Komplexitätsklassen dient in den untersuchten Publikationen in erster Linie zur Bewertung der reinen Komplexität eines Problems. So ist es möglich abzuwiegen, wie schwer ein Problem sein kann und welche Methoden zur Algorithmusentwicklung angewendet werden können.

Eine Hypothese zur geringen praktischen Verwendung steht im Zusammenhang mit den großen Kosten und auch möglichen negativen Auswirken, sollte der Optimierungsvorschlag fehlschlagen, wie z. B. in (Zhang et al. 2021). Hier geht nicht nur Geld verloren, sondern auch das Ökosystem und Menschenleben stehen auf dem Spiel. Simulationen bieten im Hinblick auf den Sicherheitsaspekt bessere Bedingungen und werden wahrscheinlich deshalb präferiert.

4 Zusammenfassung und Ausblick

Diese Arbeit hat zum Ziel, die Praxisrelevanz der Komplexitätsklassentheorie zu ermitteln. Sie konzentriert sich auf Fallstudien und ermittelt Anwendungsfälle. Um das Vorgehen darzustellen, sind vorangehend die essentiellen Aspekte der Komplexitätsklassentheorie zusammengefasst und die wichtigsten Komplexitätsklassen aufgelistet worden. Als Arbeitsmethode bietet sich eine systematische Literaturrecherche nach vom Brocke et al. (2009) an; diese wird folglich gewählt. Anhand von Suchstrings haben sich Publikationen ermitteln lassen, die mit eigens definierten Schlüsselaspekten ausgewertet werden konnten. Anschließend werden drei der bestehenden Publikationen analysiert und dargestellt. Abschließend vergleicht eine Tabelle die Gemeinsamkeiten sowie Unterschiede in den Forschungen. Da nur wenige Publikationen mit den Anforderungen übereinstimmen und viele digitale Experimente verwendet worden sind, ist beispielhaft auch eine solche Forschungsarbeit zur Gegenüberstellung vorgestellt worden. Wegen der mangelnden Existenz von Fallbeispielen wird das gewünschte Ergebnis nicht erreicht, sodass davon auszugehen ist, dass die Komplexitätsklassentheorie keine Anwendung in der Praxis findet. Sie dient lediglich der Einordnung und Orientierung hinsichtlich der Komplexität von Problemen. Weiterhin werden Probleme in ihrer Komplexität vereinfacht; dafür werden theoretische Konzepte und mathematische Beweismethoden angewendet. Die Verbindung zu Algorithmen zeigt auf, dass diese meistens im Fokus von Forschungarbeiten liegen, aber keinen direkten Bezug zur Komplexitätsklassentheorie aufzeigen müssen.

Zukünftige Forschungen könnten sich mit den Analysen der digitalen Experimente auseinandersetzen und eruieren, inwieweit diese für praktische Einsätze verwendet werden können. Weiterhin ließe sich darin die Bedeutung der Komplexitätsklassen im Hinblick auf Experimente evaluieren. Darüber hinaus ließe sich eine Formel entwerfen, mit der die Güte von recherchierten Publikationen besser bewertet werden könnte. Über eine solche Formel könnten auch die Forschenden anderer Fachgebiete passendere Artikel für ihre Forschung finden, wenn die Anforderungen und Aspekte angepasst werden. Dies ließe im Endeffekt eine direkte Bewertung von Publikation mittels eines Punktesystems oder einer Prozentangabe realistisch erscheinen.

Literaturverzeichnis

Literaturverzeichnis

Anacleto, Eduardo A. J./Meneses, Claudio N./Liang, Ricardo N. (2021). Fast r-flip move evaluations via closed-form formulae for Boolean quadratic programming problems with generalized upper bound constraints. Computers & Operations Research ; 37 132. https://doi.org/10.1016/j.cor.2021.105297.

Arkin, E./Joneja, D./Roundy, R. (1989). Computational complexity of uncapacitated multi-echelon production planning problems. Operations Research Letters ; 8 8 (2), 61–67.

Arora, Sanjeev/Barak, Boaz (2010). Computational complexity. A modern approach. Cambridge, Cambridge Univ. Press.

B. Li (2015). Container terminal logistics scheduling and decision-making within the conceptual framework of computational thinking. In: 2015 54th IEEE Conference on Decision and Control (CDC), 2015 54th IEEE Conference on Decision and Control (CDC), 330–337.

Bauernhansl, Thomas et al. (2014). Industrie 4.0 in Produktion, Auttomatisierung und Logistik. Anwendung · Technologien · Migration. Dordrecht, Springer.

Bovet, Daniel Pierre/Crescenzi, Pierluigi/Bovet, D. (1994). Introduction to the Theory of Complexity. Prentice Hall London.

Boysen, Nils/Briskorn, Dirk/Schwerdfeger, Stefan (2019). Matching supply and demand in a sharing economy: Classification, computational complexity, and application. European Journal of Operational Research ; 11 278 (2), 578–595. https://doi.org/10.1016/j.ejor.2019.04.032.

Büttcher, Stefan/Clarke, Charles L. A./Cormack, Gordon V. (2016). Information retrieval. Implementing and evaluating search engines. Cambridge (Mass.)/London, ǂThe ǂMIT Press.

Caliwag, Jasmin A./Aragon, Ma. Christina C./Medina, Ruji P. (2019). Bin Packing Problem with Compatible Categories Utilizing an Enhanced Variable Neighborhood Search Algorithm. In: Proceedings of the 2019 IEEE 4th International Conference on Technology, Informatics, Management, Engineering & Environment (TIME-E). Bali, Indonesia, November 13-15, 2019. [Piscataway, NJ], IEEE, 6–11.

Caprara, Alberto/Carvalho, Margarida/Lodi, Andrea/Woeginger, Gerhard J. (2014). A Study on the Computational Complexity of the Bilevel Knapsack Problem. SIAM Journal on Optimization ; 24 24 (2), 823–838. https://doi.org/10.1137/130906593.

Çelik, Melih/Süral, Haldun (2019). Order picking in parallel-aisle warehouses with multiple blocks: complexity and a graph theory-based heuristic. International Journal of Production Research ; 57 57 (3), 888–906. https://doi.org/10.1080/00207543.2018.1489154.

Chen, Huiyu/Sun, Lijun/Chen, Tianfei. Discrete Whale Optimization Algorithm for Capacitated Vehicle Routing Problem. In: 2023 China Automation Congress CAC 2023, 5231–5236.

Coene, Sofie/Spieksma, Frits C. R./Dutta, Atri/Tsiotras, Panagiotis (2012). On the Computational Complexity of Peer-to-Peer Satellite Refueling Strategies. INFOR. Information Systems and Operational Research ; 50 50 (2), 88–94. https://doi.org/10.3138/infor.50.2.088.

Cooper, Harris M. (1988). Organizing knowledge syntheses: A taxonomy of literature reviews. Knowledge in Society 1 (1), 104–126. https://doi.org/10.1007/BF03177550.

Cormen, Thomas H./Leiserson, Charles E./Rivest, Ronald L./Stein, Clifford (2022). Introduction to algorithms. MIT press.

Danancier, Kevin/Ruvio, Delphine/Sung, Inkyung/Nielsen, Peter (2019). Comparison of Path Planning Algorithms for an Unmanned Aerial Vehicle Deployment Under Threats. IFAC-PapersOnLine 52 (13), 1978–1983. https://doi.org/10.1016/j.ifacol.2019.11.493.

Davis, Martin (1958). Computability & unsolvability. New York, Dover Publications.

Ding, Y./Chen, X./Lamata, L./Solano, E./Sanz, M. (2021). Implementation of a Hybrid Classical-Quantum Annealing Algorithm for Logistic Network Design. SN Computer Science 2 (2). https://doi.org/10.1007/s42979-021-00466-2.

Faigle, U./Kern, W. (1994). Computational complexity of some maximum average weight problems with precedence constraints. Operations Research ; 42 42 (4), 688–694.

Felinskas, Grazvydas/Sakalauskas, Leonidas (2008). Variable neighborhood Simulated Annealing method and application for design. In: E. K. Zavadskas/A. Kaklauskas/M. J. Skibniewski (Hg.). 25TH INTERNATIONAL SYMPOSIUM ON AUTOMATION AND ROBOTICS IN CONSTRUCTION - ISARC-2008, 566–572.

Fleischmann, Bernhard (2018). Begriffliche Grundlagen der Logistik. In: Various Authors (Hg.). Begriff der Logistik, logistische Systeme und Prozesse. [United States], Springer Berlin Heidelberg, 1–16.

Fliedner, Malte/Briskorn, Dirk/Boysen, Nils (2016). Vehicle scheduling under the warehouse-on-wheels policy. Discrete Applied Mathematics 205, 52–61. https://doi.org/10.1016/j.dam.2015.11.015.

G. Strawn (2014). Alan Turing. IT Professional 16 (1), 5–7. https://doi.org/10.1109/MITP.2014.2.

Gadhikar, Lakshmi/Bomble, Sanskar/Chatur, Swayam/Karekar, Sarvesh/Khamkar, Sarvesh. Comparative Analysis of Algorithms for Cargo Container Space Optimization. In: 2023 6th International Conference, 319–325.

Garey, Michael R./Johnson, David S. (1979). Computers and intractability. San Francisco, Freeman.

Gerdt (2023). Systematische Literaturanalyse zu bestehenden Vorgehens- und Referenzmodellen zur Zuverlässigkeitsprognose von Heizsystemen. Online verfügbar unter https://www.itpl-vw.mb.tu-dortmund.de/publikationen/FP_2023_Gerdt.pdf (abgerufen am 13.08.2024).

Ghelichi, Zabih/Gentili, Monica/Mirchandani, Pitu B. (2022). Drone logistics for uncertain demand of disaster-impacted populations. Transportation Research Part C: Emerging Technologies 141, 103735. https://doi.org/10.1016/j.trc.2022.103735.

Gopalan, Ram (2015). Computational complexity of convoy movement planning problems. Mathematical Methods of Operations Research ; 82 82 (1), 31–60. https://doi.org/10.1007/s00186-015-0503-3.

Göpfert (2019). Logistik der Zukunft - Logistics for the Future. Springer Fachmedien Wiesbaden.

Guo, Peng/Weidinger, Felix/Boysen, Nils (2019). Parallel machine scheduling with job synchronization to enable efficient material flows in hub terminals. Omega 89, 110–121. https://doi.org/10.1016/j.omega.2018.10.003.

Gurari, Eitan (1989). An introduction to the theory of computation. Computer Science Press Rockville.

Hall, N. G./Posner, M. E./Potts, C. N. (1998). Scheduling with finite capacity input buffers. Operations Research 46 (3 SUPPL. 1), S154-S159. https://doi.org/10.1287/opre.46.3.s154.

Hartmanis, J./Stearns, R. E. (1965). On the Computational Complexity of Algorithms. Transactions of the American Mathematical Society 117, 285. https://doi.org/10.2307/1994208.

Hartmann, Evi/Kaiser, Gernot (2005). Produktion. Problemfelder der Produktion im Überblick. St. Gallen/[Sternenfels], SMG Publ; Verl. Wiss. und Praxis.

Helmold, Marc (2024). Erfolgreiche Transformation Zum Digitalen Champion. Wettbewerbsvorteile Durch Digitalisierung und Künstliche Intelligenz. Wiesbaden, Springer Vieweg. in Springer Fachmedien Wiesbaden GmbH.

Hopcroft, J. (2012). On the impact of Turing machines. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1–2.

Hou (2021). Fundamentals of Logic and Computation. [S.l.], Springer International Publishing.

Hromkovič, Juraj (2014). Theoretische Informatik. Formale Sprachen, Berechenbarkeit, Komplexitätstheorie, Algorithmik, Kommunikation und Kryptographie. 5. Aufl. Wiesbaden, Springer Fachmedien Wiesbaden.

Iskan, Stefan/Obst, Thomas. Logistik 4.0 Industrie 4.0 und die Konsequenzen für Logistikdienstleister. Speditionen zukunftsfähig machen.

Ivanova, Lubov/Kurkin, Andrey/Ivanov, Sergei. Methods for optimizing routes in digital logistics. In: E3S Web of Conferences 258, 02015 (2021), 2015.

Jan Vom Brocke/Alexander Simons/Bjoern Niehaves/Kai Riemer/Anne Cleven (2009). Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process. In: A COMPREHENSIVE STUDY OF INTERNET OF THINGS AND DIGITAL BUSINESS ON THE ECONOMIC GROWTH AND ITS IMPACT ON HUMAN RESOURCE MANAGEMENT, 17th European Conference on Information Systems (ECIS).

K. Rüdiger Reischuk. Komplexitätstheorie Band I: Grundlagen. Maschinenmodelle, Zeit- und Platzkomplexität, Nichtdeterminismus.

Karp, R. M. (1975). On the Computational Complexity of Combinatorial Problems. Networks 5 (1), 45–68. https://doi.org/10.1002/net.1975.5.1.45.

Keith, Kolton/Castillo-Villar, Krystel K./Alaeddini, Adel (2024). Machine Learning-Based Problem Space Reduction in Stochastic Programming Models: An Application in Biofuel Supply Chain Network Design. IEEE Access ; 11 12, 33852–33866. https://doi.org/10.1109/ACCESS.2024.3372516.

Knuth, Donald E. (1976). Big Omicron and big Omega and big Theta. SIGACT News 8 (2), 18–24. https://doi.org/10.1145/1008328.1008329.

Kolen, Antoon W.J./Kroon, Leo G. (1993). On the computational complexity of (maximum) shift class scheduling. European Journal of Operational Research ; 11 64 (1), 138–151. https://doi.org/10.1016/0377-2217(93)90014-E.

Kovalyov, M. Y./Potts, C. N./Strusevich, V. A. (2004). Batching decisions for assembly production systems. European Journal of Operational Research ; 11 157 (3), 620–642. https://doi.org/10.1016/S0377-2217(03)00250-9.

Kummer, Sebastian (Hg.) (2019). Grundzüge der Beschaffung, Produktion und Logistik. 4. Aufl. Hallbergmoos, Pearson.

Lee, Chung-Yee/Qi, Xiangtong (2009). Vehicle scheduling with combinable delivery and pickup operations. Operations Research Letters ; 8 37 (6), 399–404. https://doi.org/10.1016/j.orl.2009.06.002.

Lee, Kangbok/Lei, Lei/Pinedo, Michael/Wang, Shengbin (2013). Operations scheduling with multiple resources and transportation considerations. International Journal of Production Research ; 57 51 (23-24), 7071–7090. https://doi.org/10.1080/00207543.2013.781283.

Li, Bin/Sun, Bing/Yao, Wei/He, Yuqing/Song, Guanggang (2019). Container Terminal Oriented Logistics Generalized Computational Complexity. IEEE Access ; 11 7, 94737–94756. https://doi.org/10.1109/ACCESS.2019.2928684.

Li, Chung-Lun/Vairaktarakis, George/Lee, Chung-Yee (2005). Machine scheduling with deliveries to multiple customer locations. European Journal of Operational Research ; 11 164 (1), 39–51. https://doi.org/10.1016/j.ejor.2003.11.022.

Li, Penglin/Zhi, Chen/Li, Wei (2021). An Algorithm to Solve Heterogeneous Vehicle Routing Problem With Second Trip. IEEE Access ; 11 9, 12241–12255. https://doi.org/10.1109/ACCESS.2021.3049510.

Liu, Yubin/Ye, Qiming/Escribano-Macias, Jose/Feng, Yuxiang/Candela, Eduardo/Angeloudis, Panagiotis (2023). Route planning for last-mile deliveries using mobile parcel lockers: A hybrid q-learning network approach. Transportation Research Part E: Logistics and Transportation Review 177, 103234. https://doi.org/10.1016/j.tre.2023.103234.

Ma, Hao. Simulation of Logistics Route Optimization Model Based on Genetic Algorithm Optimization Neural Network. In: 2023 2nd International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS), 105–110.

Manzini, Riccardo/Gebennini, Elisa (2008). Optimization models for the dynamic facility location and allocation problem. International Journal of Production Research ; 57 46 (8), 2061–2086. https://doi.org/10.1080/00207540600847418.

Mathijs M. de Weerdt/Michael Albert/Vincent Conitzer/Koos van der Linden (2018). Complexity of Scheduling Charging in the Smart Grid. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. Stockholm, Sweden, International Foundation for Autonomous Agents and Multiagent Systems, 1924–1926.

Matsutani, Kazuya/Kimura, Shigetomo (2022). Delivery Routing to Reduce Calculation Load of Drones on Divided Logistics Areas for Drone Logistics Networks. In: 2022 Tenth International Symposium on Computing and Networking workshops. CANDARW 2022 : 21-22 November 2022, Himeji, Japan : proceedings. Piscataway, NJ, IEEE, 44–48.

Mich, Luisa/Berry, Daniel M. (2019). A Gentle Introduction to Computational Complexity Through an Examination of Noodle Making. https://doi.org/10.20368/1971-8829/1399.

Monteiro, Cristiano Martins/dos Santos, Vinicius Fernandes/Davis Junior, Clodoveu Augusto (2024). Computational complexity and linear formulations for optimizing the location of shared mobility stations. International Transactions in Operational Research ; 31 31 (5), 2903–2929. https://doi.org/10.1111/itor.13411.

Morim, André/Campuzano, Giovanni/Amorim, Pedro/Mes, Martijn/Lalla-Ruiz, Eduardo (2024). The drone-assisted vehicle routing problem with robot stations. Expert Systems with Applications 238, 121741. https://doi.org/10.1016/j.eswa.2023.121741.

Motwani, Rajeev/Raghavan, Prabhakar (op. 2007). Randomized algorithms. 9. Aufl. Cambridge, Cambridge Univ.

Murray, Chase C./Raj, Ritwik (2020). The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. Transportation Research Part C: Emerging Technologies 110, 368–398. https://doi.org/10.1016/j.trc.2019.11.003.

Nebl, Theodor (2007). Produktionswirtschaft, 6., vollständig überarbeitete und erweiterte Auflage. De Gruyter.

Ng, C. T./Barketau, M. S./Cheng, T. C. E./Kovalyov, Mikhail Y. (2010). 'Product Partition' and related problems of scheduling and systems reliability : computational complexity and approximation. European journal of operational research ; 207 207 (2), 601–605.

Papadimitriou, Christos H. (2003). Computational complexity. In: Encyclopedia of computer science, 260–265.

Park, Myoung-Ju/Min, Yun-Hong/Choi, Byung-Cheon (2018). Two-agent scheduling with generalized due dates. The Asian Journal of Shipping and Logistics 34 (4), 345–350. https://doi.org/10.1016/j.ajsl.2018.12.008.

Peiyou Chen/Xinming Xu (2008). A hybrid algorithm for multi-depot vehicle routing problem. In: 2008 IEEE International Conference on Service Operations and Logistics, and Informatics, 2008 IEEE International Conference on Service Operations and Logistics, and Informatics, 2031–2034.

Peng Wang/Stephan Bohacek (2008). On the practical complexity of solving the maximum weighted independent set problem for optimal scheduling in wireless networks. In: Proceedings of the 4th Annual International Conference on Wireless Internet. <conf-loc>, <city>Maui</city>, <state>Hawaii</state>, <country>USA</country>, </conf-loc>, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Article 15.

Pereira, L. M. (2012). Turing is among us. JOURNAL OF LOGIC AND COMPUTATION 22 (6), 1257–1277. https://doi.org/10.1093/logcom/exs035.

Pfohl, Hans-Christian (2004). Logistiksysteme. Betriebswirtschaftliche Grundlagen. 7. Aufl. Berlin/Heidelberg/New York/Hongkong/London/Mailand/Paris/Tokio, Springer.

Riley, David D./Hunt, Kenny A. (2014). Computational thinking for the modern problem solver. Boca Raton, Fla, CRC Press.

Rinaldi, Marco/Primatesta, Stefano/Bugaj, Martin/Rostáš, Ján/Guglieri, Giorgio (2023). Development of Heuristic Approaches for Last-Mile Delivery TSP with a Truck and Multiple Drones. Drones ; 7 7 (7), 407. https://doi.org/10.3390/drones7070407.

Rudek, Radosław (2012). Scheduling problems with position dependent job processing times: computational complexity results. Annals of Operations Research ; 196 196 (1), 491–516. https://doi.org/10.1007/s10479-012-1098-1.

S. A. Cook (2000). The P versus NP problem.

S. Prabu/A. Tripathi/K. Kaur/M. M. Krishna/A. Bora/M. F. Hasan (2021). A Comprehensive Study of Internet of Things and Digital Business on the Economic Growth and its Impact on Human Resource Management. In: 2021 International Conference on Computing Sciences (ICCS), 2021 International Conference on Computing Sciences (ICCS), 91–94.

Sarid, Shahar/Shapiro, Amir (2008). Classifying the multi robot path finding problem into a quadratic competitive complexity class. Annals of Mathematics & Artificial Intelligence ; 62 52 (2-4), 169–203. https://doi.org/10.1007/s10472-009-9122-0.

Savitch, Walter J. (1970). Relationships between nondeterministic and deterministic tape complexities. Journal of Computer and System Sciences 4 (2), 177–192. https://doi.org/10.1016/S0022-0000(70)80006-X.

Sedgewick, Robert (2002). Algorithmen. 2. Aufl. München/Boston [u.a.], Addison-Wesley.

Sipser, Michael (2005). Introduction to the theory of competition. 2. Aufl. New York, Brooks Cole.

Stockmeyer, Larry J. (1992). Chapter 9 Computational complexity. In: Handbooks in Operations Research and Management Science : Computing. Elsevier, 455–517.

T. Kitamura/K. Okamoto (2012). Automated Route Planning for Milk-run Transport Logistics Using Model Checking. In: 2012 Third International Conference on Networking and Computing, 2012 Third International Conference on Networking and Computing, 240–246.

Tian, Qian/Yin, Qingwei/Meng, Yagang (2022). Swarm Intelligence Technique for Supply Chain Market in Logistic Analytics Management. International Journal of Information Systems and Supply Chain Management 15 (4), 1–20. https://doi.org/10.4018/IJISSCM.305845.

Trefethen, Lloyd N. (1997). Numerical linear algebra. Society for Industrial and Applied Mathematics.

Vanajakumari, Manoj/Sun, Haoying/Jones, Ashley/Sriskandarajah, Chelliah (2022). Supply chain planning: A case for Hybrid Cross-Docks. Omega 108, 102585. https://doi.org/10.1016/j.omega.2021.102585.

Villar-Medina, Israel/López-Ortega, Omar/Hernández-Gómez, Roberto (2009). Implementation of a supervised learning technique in a multi-agent system for building production orders. International Journal of Advanced Manufacturing Technology ; 40 40 (7-8), 808–818. https://doi.org/10.1007/s00170-008-1387-7.

vom Brocke, Jan/Simons, Alexander/Riemer, Kai/Niehaves, Björn/Plattfaut, Ralf/Cleven, Anne (2015). Standing on the Shoulders of Giants: Challenges and Recommendations of Literature Search in Information Systems Research. Communications of the Association for Information Systems 37. https://doi.org/10.17705/1CAIS.03709.

Walter, Rico/Boysen, Nils/Scholl, Armin (2013). The discrete forward–reserve problem – Allocating space, selecting products, and area sizing in forward order picking. European

Journal of Operational Research ; 11 229 (3), 585–594. https://doi.org/10.1016/j.ejor.2013.02.047.

Webster, Jane/Watson, Richard T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly 26 (2), xiii–xxiii. Online verfügbar unter http://www.jstor.org/stable/4132319.

Wegener, Ingo (2003). Komplexitätstheorie. Grenzen der Effizienz von Algorithmen. Berlin, Heidelberg, Springer Berlin Heidelberg.

Weyland, D. (2014). On the computational complexity of the Probabilistic Traveling Salesman Problem with Deadlines. Theoretical Computer Science ; 540 540-541, 156–168. https://doi.org/10.1016/j.tcs.2013.11.028.

Xie, Lin/Thieme, Nils/Krenzler, Ruslan/Li, Hanyi (2021). Introducing split orders and optimizing operational policies in robotic mobile fulfillment systems. European Journal of Operational Research ; 11 288 (1), 80–97. https://doi.org/10.1016/j.ejor.2020.05.032.

Yamada, Hisao (1962). Real-Time Computation and Recursive Functions Not Real-Time Computable. IEEE Transactions on Electronic Computers EC-11 (6), 753–760. https://doi.org/10.1109/TEC.1962.5219459.

Ying, Kuo-Ching/Lu, Chung-Cheng/Chen, Jhao-Cheng (2016). Exact algorithms for single-machine scheduling problems with a variable maintenance. Computers & Industrial Engineering 98, 427–433. https://doi.org/10.1016/j.cie.2016.05.037.

Zachariadis, Emmanouil E./Kiranoudis, Chris T. (2010). A strategy for reducing the computational complexity of local search-based methods for the vehicle routing problem. Computers & Operations Research ; 37 37 (12), 2089–2105. https://doi.org/10.1016/j.cor.2010.02.009.

Zawadzki, E./Platzer, A./Gordon, G. J. (2013). A generalization of SAT and #SAT for robust policy evaluation. In: IJCAI International Joint Conference on Artificial Intelligence, 2583–2589.

Zelewski, Stephan (1988). Komplexitätstheorie.

Zhang, Lingye/Lu, Jing/Yang, Zaili (2021). Optimal scheduling of emergency resources for major maritime oil spills considering time-varying demand and transportation networks. European Journal of Operational Research ; 11 293 (2), 529–546. https://doi.org/10.1016/j.ejor.2020.12.040.

Anacleto, Eduardo A. J./Meneses, Claudio N./Liang, Ricardo N. (2021). Fast r-flip move evaluations via closed-form formulae for Boolean quadratic programming problems with generalized upper bound constraints. Computers & Operations Research ; 37 132. https://doi.org/10.1016/j.cor.2021.105297.

Arkin, E./Joneja, D./Roundy, R. (1989). Computational complexity of uncapacitated multi-echelon production planning problems. Operations Research Letters ; 8 8 (2), 61–67.

Arora, Sanjeev/Barak, Boaz (2010). Computational complexity. A modern approach. Cambridge, Cambridge Univ. Press.

B. Li (2015). Container terminal logistics scheduling and decision-making within the conceptual framework of computational thinking. In: 2015 54th IEEE Conference on Decision and Control (CDC), 2015 54th IEEE Conference on Decision and Control (CDC), 330–337.

Bauernhansl, Thomas et al. (2014). Industrie 4.0 in Produktion, Auttomatisierung und Logistik. Anwendung · Technologien · Migration. Dordrecht, Springer.

Blanke, Dominik. Systematische Literaturanalyse zu bestehenden Vorgehens- und Referenzmodellen im Bereich Datenanalyse zur Zuverlässigkeitsprognose von Heizsystemen. Online verfügbar unter https://www.itpl-vw.mb.tu-dortmund.de/publikationen/BA_2023_Blanke.pdf (abgerufen am 09.08.2024).

Bovet, Daniel Pierre/Crescenzi, Pierluigi/Bovet, D. (1994). Introduction to the Theory of Complexity. Prentice Hall London.

Boysen, Nils/Briskorn, Dirk/Schwerdfeger, Stefan (2019). Matching supply and demand in a sharing economy: Classification, computational complexity, and application. European Journal of Operational Research ; 11 278 (2), 578–595. https://doi.org/10.1016/j.ejor.2019.04.032.

Büttcher, Stefan/Clarke, Charles L. A./Cormack, Gordon V. (2016). Information retrieval. Implementing and evaluating search engines. Cambridge (Mass.)/London, ǂThe ǂMIT Press.

Caliwag, Jasmin A./Aragon, Ma. Christina C./Medina, Ruji P. (2019). Bin Packing Problem with Compatible Categories Utilizing an Enhanced Variable Neighborhood Search Algorithm. In: Proceedings of the 2019 IEEE 4th International Conference on Technology, Informatics, Management, Engineering & Environment (TIME-E). Bali, Indonesia, November 13-15, 2019. [Piscataway, NJ], IEEE, 6–11.

Caprara, Alberto/Carvalho, Margarida/Lodi, Andrea/Woeginger, Gerhard J. (2014). A Study on the Computational Complexity of the Bilevel Knapsack Problem. SIAM Journal on Optimization ; 24 24 (2), 823–838. https://doi.org/10.1137/130906593.

Çelik, Melih/Süral, Haldun (2019). Order picking in parallel-aisle warehouses with multiple blocks: complexity and a graph theory-based heuristic. International Journal of Production Research ; 57 57 (3), 888–906. https://doi.org/10.1080/00207543.2018.1489154.

Chen, Huiyu/Sun, Lijun/Chen, Tianfei. Discrete Whale Optimization Algorithm for Capacitated Vehicle Routing Problem. In: 2023 China Automation Congress CAC 2023, 5231–5236.

Coene, Sofie/Spieksma, Frits C. R./Dutta, Atri/Tsiotras, Panagiotis (2012). On the Computational Complexity of Peer-to-Peer Satellite Refueling Strategies. INFOR. Information Systems and Operational Research ; 50 50 (2), 88–94. https://doi.org/10.3138/infor.50.2.088.

Cooper, Harris M. (1988). Organizing knowledge syntheses: A taxonomy of literature reviews. Knowledge in Society 1 (1), 104–126. https://doi.org/10.1007/BF03177550.

Cormen, Thomas H./Leiserson, Charles E./Rivest, Ronald L./Stein, Clifford (2022). Introduction to algorithms. MIT press.

Danancier, Kevin/Ruvio, Delphine/Sung, Inkyung/Nielsen, Peter (2019). Comparison of Path Planning Algorithms for an Unmanned Aerial Vehicle Deployment Under Threats. IFAC-PapersOnLine 52 (13), 1978–1983. https://doi.org/10.1016/j.ifacol.2019.11.493.

Davis, Martin (1958). Computability & unsolvability. New York, Dover Publications.

Ding, Y./Chen, X./Lamata, L./Solano, E./Sanz, M. (2021). Implementation of a Hybrid Classical-Quantum Annealing Algorithm for Logistic Network Design. SN Computer Science 2 (2). https://doi.org/10.1007/s42979-021-00466-2.

Faigle, U./Kern, W. (1994). Computational complexity of some maximum average weight problems with precedence constraints. Operations Research ; 42 42 (4), 688–694.

Felinskas, Grazvydas/Sakalauskas, Leonidas (2008). Variable neighborhood Simulated Annealing method and application for design. In: E. K. Zavadskas/A. Kaklauskas/M. J. Skibniewski (Hg.). 25TH INTERNATIONAL SYMPOSIUM ON AUTOMATION AND ROBOTICS IN CONSTRUCTION - ISARC-2008, 566–572.

Fleischmann, Bernhard (2018). Begriffliche Grundlagen der Logistik. In: Various Authors (Hg.). Begriff der Logistik, logistische Systeme und Prozesse. [United States], Springer Berlin Heidelberg, 1–16.

Fliedner, Malte/Briskorn, Dirk/Boysen, Nils (2016). Vehicle scheduling under the warehouse-on-wheels policy. Discrete Applied Mathematics 205, 52–61. https://doi.org/10.1016/j.dam.2015.11.015.

G. Strawn (2014). Alan Turing. IT Professional 16 (1), 5–7. https://doi.org/10.1109/MITP.2014.2.

Gadhikar, Lakshmi/Bomble, Sanskar/Chatur, Swayam/Karekar, Sarvesh/Khamkar, Sarvesh. Comparative Analysis of Algorithms for Cargo Container Space Optimization. In: 2023 6th International Conference, 319–325.

Garey, Michael R./Johnson, David S. (1979). Computers and intractability. San Francisco, Freeman.

Gerdt, Angelina (2023). Systematische Literaturanalyse zu bestehenden Vorgehens- und Referenzmodellen zur Zuverlässigkeitsprognose von Heizsystemen. Online verfügbar unter https://www.itpl-vw.mb.tu-dortmund.de/publikationen/FP_2023_Gerdt.pdf (abgerufen am 13.08.2024).

Ghelichi, Zabih/Gentili, Monica/Mirchandani, Pitu B. (2022). Drone logistics for uncertain demand of disaster-impacted populations. Transportation Research Part C: Emerging Technologies 141, 103735. https://doi.org/10.1016/j.trc.2022.103735.

Gopalan, Ram (2015). Computational complexity of convoy movement planning problems. Mathematical Methods of Operations Research ; 82 82 (1), 31–60. https://doi.org/10.1007/s00186-015-0503-3.

Göpfert (2019). Logistik der Zukunft - Logistics for the Future. Springer Fachmedien Wiesbaden.

Guo, Peng/Weidinger, Felix/Boysen, Nils (2019). Parallel machine scheduling with job synchronization to enable efficient material flows in hub terminals. Omega 89, 110–121. https://doi.org/10.1016/j.omega.2018.10.003.

Gurari, Eitan (1989). An introduction to the theory of computation. Computer Science Press Rockville.

Hall, N. G./Posner, M. E./Potts, C. N. (1998). Scheduling with finite capacity input buffers. Operations Research 46 (3 SUPPL. 1), S154-S159. https://doi.org/10.1287/opre.46.3.s154.

Hartmanis, J./Stearns, R. E. (1965). On the Computational Complexity of Algorithms. Transactions of the American Mathematical Society 117, 285. https://doi.org/10.2307/1994208.

Hartmann, Evi/Kaiser, Gernot (2005). Produktion. Problemfelder der Produktion im Überblick. St. Gallen/[Sternenfels], SMG Publ; Verl. Wiss. und Praxis.

Helmold, Marc (2024). Erfolgreiche Transformation Zum Digitalen Champion. Wettbewerbsvorteile Durch Digitalisierung und Künstliche Intelligenz. Wiesbaden, Springer Vieweg. in Springer Fachmedien Wiesbaden GmbH.

Hopcroft, J. (2012). On the impact of Turing machines. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1–2.

Hou (2021). Fundamentals of Logic and Computation. [S.l.], Springer International Publishing.

Hromkovič, Juraj (2014). Theoretische Informatik. Formale Sprachen, Berechenbarkeit, Komplexitätstheorie, Algorithmik, Kommunikation und Kryptographie. 5. Aufl. Wiesbaden, Springer Fachmedien Wiesbaden.

Iskan, Stefan/Obst, Thomas. Logistik 4.0 Industrie 4.0 und die Konsequenzen für Logistikdienstleister. Speditionen zukunftsfähig machen.

Ivanova, Lubov/Kurkin, Andrey/Ivanov, Sergei. Methods for optimizing routes in digital logistics. In: E3S Web of Conferences 258, 02015 (2021), 2015.

Jan Vom Brocke/Alexander Simons/Bjoern Niehaves/Kai Riemer/Anne Cleven (2009). Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process. In: A COMPREHENSIVE STUDY OF INTERNET OF THINGS AND DIGITAL BUSINESS ON THE ECONOMIC GROWTH AND ITS IMPACT ON HUMAN RESOURCE MANAGEMENT, 17th European Conference on Information Systems (ECIS).

K. Rüdiger Reischuk. Komplexitätstheorie Band I: Grundlagen. Maschinenmodelle, Zeit- und Platzkomplexität, Nichtdeterminismus.

Karp, R. M. (1975). On the Computational Complexity of Combinatorial Problems. Networks 5 (1), 45–68. https://doi.org/10.1002/net.1975.5.1.45.

Keith, Kolton/Castillo-Villar, Krystel K./Alaeddini, Adel (2024). Machine Learning-Based Problem Space Reduction in Stochastic Programming Models: An Application in Biofuel Supply Chain Network Design. IEEE Access ; 11 12, 33852–33866. https://doi.org/10.1109/ACCESS.2024.3372516.

Knuth, Donald E. (1976). Big Omicron and big Omega and big Theta. SIGACT News 8 (2), 18–24. https://doi.org/10.1145/1008328.1008329.

Kolen, Antoon W.J./Kroon, Leo G. (1993). On the computational complexity of (maximum) shift class scheduling. European Journal of Operational Research ; 11 64 (1), 138–151. https://doi.org/10.1016/0377-2217(93)90014-E.

Kovalyov, M. Y./Potts, C. N./Strusevich, V. A. (2004). Batching decisions for assembly production systems. European Journal of Operational Research ; 11 157 (3), 620–642. https://doi.org/10.1016/S0377-2217(03)00250-9.

Kummer, Sebastian (Hg.) (2019). Grundzüge der Beschaffung, Produktion und Logistik. 4. Aufl. Hallbergmoos, Pearson.

Lee, Chung-Yee/Qi, Xiangtong (2009). Vehicle scheduling with combinable delivery and pickup operations. Operations Research Letters ; 8 37 (6), 399–404. https://doi.org/10.1016/j.orl.2009.06.002.

Lee, Kangbok/Lei, Lei/Pinedo, Michael/Wang, Shengbin (2013). Operations scheduling with multiple resources and transportation considerations. International Journal of Production Research ; 57 51 (23-24), 7071–7090. https://doi.org/10.1080/00207543.2013.781283.

Li, Bin/Sun, Bing/Yao, Wei/He, Yuqing/Song, Guanggang (2019). Container Terminal Oriented Logistics Generalized Computational Complexity. IEEE Access ; 11 7, 94737–94756. https://doi.org/10.1109/ACCESS.2019.2928684.

Li, Chung-Lun/Vairaktarakis, George/Lee, Chung-Yee (2005). Machine scheduling with deliveries to multiple customer locations. European Journal of Operational Research ; 11 164 (1), 39–51. https://doi.org/10.1016/j.ejor.2003.11.022.

Li, Penglin/Zhi, Chen/Li, Wei (2021). An Algorithm to Solve Heterogeneous Vehicle Routing Problem With Second Trip. IEEE Access ; 11 9, 12241–12255. https://doi.org/10.1109/ACCESS.2021.3049510.

Liu, Yubin/Ye, Qiming/Escribano-Macias, Jose/Feng, Yuxiang/Candela, Eduardo/Angeloudis, Panagiotis (2023). Route planning for last-mile deliveries using mobile parcel lockers: A hybrid q-learning network approach. Transportation Research Part E: Logistics and Transportation Review 177, 103234. https://doi.org/10.1016/j.tre.2023.103234.

Ma, Hao. Simulation of Logistics Route Optimization Model Based on Genetic Algorithm Optimization Neural Network. In: 2023 2nd International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS), 105–110.

Manzini, Riccardo/Gebennini, Elisa (2008). Optimization models for the dynamic facility location and allocation problem. International Journal of Production Research ; 57 46 (8), 2061–2086. https://doi.org/10.1080/00207540600847418.

Mathijs M. de Weerdt/Michael Albert/Vincent Conitzer/Koos van der Linden (2018). Complexity of Scheduling Charging in the Smart Grid. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. Stockholm, Sweden, International Foundation for Autonomous Agents and Multiagent Systems, 1924–1926.

Matsutani, Kazuya/Kimura, Shigetomo (2022). Delivery Routing to Reduce Calculation Load of Drones on Divided Logistics Areas for Drone Logistics Networks. In: 2022 Tenth International Symposium on Computing and Networking workshops. CANDARW 2022 : 21-22 November 2022, Himeji, Japan : proceedings. Piscataway, NJ, IEEE, 44–48.

Mich, Luisa/Berry, Daniel M. (2019). A Gentle Introduction to Computational Complexity Through an Examination of Noodle Making. https://doi.org/10.20368/1971-8829/1399.

Monteiro, Cristiano Martins/dos Santos, Vinicius Fernandes/Davis Junior, Clodoveu Augusto (2024). Computational complexity and linear formulations for optimizing the location of shared mobility stations. International Transactions in Operational Research ; 31 31 (5), 2903–2929. https://doi.org/10.1111/itor.13411.

Morim, André/Campuzano, Giovanni/Amorim, Pedro/Mes, Martijn/Lalla-Ruiz, Eduardo (2024). The drone-assisted vehicle routing problem with robot stations. Expert Systems with Applications 238, 121741. https://doi.org/10.1016/j.eswa.2023.121741.

Motwani, Rajeev/Raghavan, Prabhakar (op. 2007). Randomized algorithms. 9. Aufl. Cambridge, Cambridge Univ.

Murray, Chase C./Raj, Ritwik (2020). The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. Transportation Research Part C: Emerging Technologies 110, 368–398. https://doi.org/10.1016/j.trc.2019.11.003.

Nebl, Theodor (2007). Produktionswirtschaft, 6., vollständig überarbeitete und erweiterte Auflage. De Gruyter.

Ng, C. T./Barketau, M. S./Cheng, T. C. E./Kovalyov, Mikhail Y. (2010). 'Product Partition' and related problems of scheduling and systems reliability : computational complexity and approximation. European journal of operational research ; 207 207 (2), 601–605.

Papadimitriou, Christos H. (2003). Computational complexity. In: Encyclopedia of computer science, 260–265.

Park, Myoung-Ju/Min, Yun-Hong/Choi, Byung-Cheon (2018). Two-agent scheduling with generalized due dates. The Asian Journal of Shipping and Logistics 34 (4), 345–350. https://doi.org/10.1016/j.ajsl.2018.12.008.

Peiyou Chen/Xinming Xu (2008). A hybrid algorithm for multi-depot vehicle routing problem. In: 2008 IEEE International Conference on Service Operations and Logistics, and Informatics, 2008 IEEE International Conference on Service Operations and Logistics, and Informatics, 2031–2034.

Peng Wang/Stephan Bohacek (2008). On the practical complexity of solving the maximum weighted independent set problem for optimal scheduling in wireless networks. In: Proceedings of the 4th Annual International Conference on Wireless Internet. <conf-loc>, <city>Maui</city>, <state>Hawaii</state>, <country>USA</country>, </conf-loc>, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Article 15.

Pereira, L. M. (2012). Turing is among us. JOURNAL OF LOGIC AND COMPUTATION 22 (6), 1257–1277. https://doi.org/10.1093/logcom/exs035.

Pfohl, Hans-Christian (2004). Logistiksysteme. Betriebswirtschaftliche Grundlagen. 7. Aufl. Berlin/Heidelberg/New York/Hongkong/London/Mailand/Paris/Tokio, Springer.

Rabe. Algorithmen TU Dortmund, IT-Systeme in der industriellen Produktion.

Riley, David D./Hunt, Kenny A. (2014). Computational thinking for the modern problem solver. Boca Raton, Fla, CRC Press.

Rinaldi, Marco/Primatesta, Stefano/Bugaj, Martin/Rostáš, Ján/Guglieri, Giorgio (2023). Development of Heuristic Approaches for Last-Mile Delivery TSP with a Truck and Multiple Drones. Drones ; 7 7 (7), 407. https://doi.org/10.3390/drones7070407.

Rudek, Radosław (2012). Scheduling problems with position dependent job processing times: computational complexity results. Annals of Operations Research ; 196 196 (1), 491–516. https://doi.org/10.1007/s10479-012-1098-1.

S. A. Cook (2000). The P versus NP problem.

S. Prabu/A. Tripathi/K. Kaur/M. M. Krishna/A. Bora/M. F. Hasan (2021). A Comprehensive Study of Internet of Things and Digital Business on the Economic Growth and its Im-

pact on Human Resource Management. In: 2021 International Conference on Computing Sciences (ICCS), 2021 International Conference on Computing Sciences (ICCS), 91–94.

Sarid, Shahar/Shapiro, Amir (2008). Classifying the multi robot path finding problem into a quadratic competitive complexity class. Annals of Mathematics & Artificial Intelligence ; 62 52 (2-4), 169–203. https://doi.org/10.1007/s10472-009-9122-0.

Savitch, Walter J. (1970). Relationships between nondeterministic and deterministic tape complexities. Journal of Computer and System Sciences 4 (2), 177–192. https://doi.org/10.1016/S0022-0000(70)80006-X.

Sedgewick, Robert (2002). Algorithmen. 2. Aufl. München/Boston [u.a.], Addison-Wesley.

Sipser, Michael (2005). Introduction to the theory of competition. 2. Aufl. New York, Brooks Cole.

Stockmeyer, Larry J. (1992). Chapter 9 Computational complexity. In: Handbooks in Operations Research and Management Science : Computing. Elsevier, 455–517.

T. Kitamura/K. Okamoto (2012). Automated Route Planning for Milk-run Transport Logistics Using Model Checking. In: 2012 Third International Conference on Networking and Computing, 2012 Third International Conference on Networking and Computing, 240–246.

Tian, Qian/Yin, Qingwei/Meng, Yagang (2022). Swarm Intelligence Technique for Supply Chain Market in Logistic Analytics Management. International Journal of Information Systems and Supply Chain Management 15 (4), 1–20. https://doi.org/10.4018/IJISSCM.305845.

Trefethen, Lloyd N. (1997). Numerical linear algebra. Society for Industrial and Applied Mathematics.

Vanajakumari, Manoj/Sun, Haoying/Jones, Ashley/Sriskandarajah, Chelliah (2022). Supply chain planning: A case for Hybrid Cross-Docks. Omega 108, 102585. https://doi.org/10.1016/j.omega.2021.102585.

Villar-Medina, Israel/López-Ortega, Omar/Hernández-Gómez, Roberto (2009). Implementation of a supervised learning technique in a multi-agent system for building production orders. International Journal of Advanced Manufacturing Technology ; 40 40 (7-8), 808–818. https://doi.org/10.1007/s00170-008-1387-7.

vom Brocke, Jan/Simons, Alexander/Riemer, Kai/Niehaves, Björn/Plattfaut, Ralf/Cleven, Anne (2015). Standing on the Shoulders of Giants: Challenges and Recommendations of Literature Search in Information Systems Research. Communications of the Association for Information Systems 37. https://doi.org/10.17705/1CAIS.03709.

Walter, Rico/Boysen, Nils/Scholl, Armin (2013). The discrete forward–reserve problem – Allocating space, selecting products, and area sizing in forward order picking. European Journal of Operational Research ; 11 229 (3), 585–594. https://doi.org/10.1016/j.ejor.2013.02.047.

Webster, Jane/Watson, Richard T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly 26 (2), xiii–xxiii. Online verfügbar unter http://www.jstor.org/stable/4132319.

Wegener, Ingo (2003). Komplexitätstheorie. Grenzen der Effizienz von Algorithmen. Berlin, Heidelberg, Springer Berlin Heidelberg.

Weyland, D. (2014). On the computational complexity of the Probabilistic Traveling Salesman Problem with Deadlines. Theoretical Computer Science ; 540 540-541, 156–168. https://doi.org/10.1016/j.tcs.2013.11.028.

Xie, Lin/Thieme, Nils/Krenzler, Ruslan/Li, Hanyi (2021). Introducing split orders and optimizing operational policies in robotic mobile fulfillment systems. European Journal of Operational Research ; 11 288 (1), 80–97. https://doi.org/10.1016/j.ejor.2020.05.032.

Yamada, Hisao (1962). Real-Time Computation and Recursive Functions Not Real-Time Computable. IEEE Transactions on Electronic Computers EC-11 (6), 753–760. https://doi.org/10.1109/TEC.1962.5219459.

Ying, Kuo-Ching/Lu, Chung-Cheng/Chen, Jhao-Cheng (2016). Exact algorithms for single-machine scheduling problems with a variable maintenance. Computers & Industrial Engineering 98, 427–433. https://doi.org/10.1016/j.cie.2016.05.037.

Zachariadis, Emmanouil E./Kiranoudis, Chris T. (2010). A strategy for reducing the computational complexity of local search-based methods for the vehicle routing problem. Computers & Operations Research ; 37 37 (12), 2089–2105. https://doi.org/10.1016/j.cor.2010.02.009.

Zawadzki, E./Platzer, A./Gordon, G. J. (2013). A generalization of SAT and #SAT for robust policy evaluation. In: IJCAI International Joint Conference on Artificial Intelligence, 2583–2589.

Zelewski, Stephan (1988). Komplexitätstheorie.

Zhang, Lingye/Lu, Jing/Yang, Zaili (2021). Optimal scheduling of emergency resources for major maritime oil spills considering time-varying demand and transportation networks. European Journal of Operational Research ; 11 293 (2), 529–546. https://doi.org/10.1016/j.ejor.2020.12.040.