

Masterarbeit

Klassifizierung von logistischen Fragestellungen des Supply-Chain-Managements und deren Beantwortung unter Berücksichtigung einer Graphenrepräsentation

verfasst von

Estelle Nguyễn-Lan Duparc

Matrikel-Nr.: 166273

Studiengang: Wirtschaftsingenieurwesen

Estelle.Duparc@tu-dortmund.de

Ausgegeben am: 15.12.2019

Eingereicht am: 16.06.2020

Prüferin: Dr.-Ing. Dipl.-Inform. Anne Antonia Scheidler

Betreuer: M.Sc. Joachim Hunker

Inhaltsverzeichnis

Inhaltsverzeichnis	I
1 Einleitung	1
2 Grundlagen des Supply-Chain-Managements	4
2.1 Logistik und Supply Chains	4
2.1.1 Entwicklungsphasen der Logistik und Logistikbegriff	4
2.1.2 Supply Chains	6
2.2 Supply-Chain-Management	8
2.2.1 Begriffsbestimmung	8
2.2.2 Ausgewählte Modelle des Supply-Chain-Managements	10
2.3 Logistische Aufgaben und bestehende Lösungsansätze	13
2.3.1 Aufgaben der Gestaltungsebene	13
2.3.2 Aufgaben der Planungsebene	15
3 Graphdatenbanksysteme und graphbasierte Verfahren	19
3.1 Daten und Informationssysteme	19
3.2 Relationale Datenbanksysteme	22
3.3 Graphdatenbanksysteme	23
3.3.1 Graphentheoretische Grundlagen und Begriffsbestimmung	24
3.3.2 Graphmodellierung	27
3.3.3 Graphbasierte Abfragesprache	28
3.4 Graphbasierte Verfahren	29
3.4.1 Graphbasierte Modelle im Supply-Chain-Management	29
3.4.2 Graph-Management-Algorithmen	32
3.4.3 Graph-Mining-Algorithmen	35
4 Untersuchung graphbasierter Verfahren zur Beantwortung logistischer Fragestellungen des Supply-Chain-Managements	40
4.1 Vorbereitung zur Erstellung einer Bewertungsmatrix	40
4.1.1 Erstellung einer Taxonomie zur Klassifizierung von Fragestellungen	41
4.1.2 Kategorisierung graphbasierter Verfahren	45
4.2 Untersuchung und Bewertung graphbasierter Verfahren	51
4.2.1 Lösungsansätze in der Gestaltungsebene	51
4.2.2 Lösungsansätze in der Planungsebene	61
4.2.3 Gesamtbewertung und Zusammenfassung	67
5 Validierung der Taxonomie und Beantwortung logistischer Fragestellungen auf Basis einer Graphenrepräsentation	76
5.1 Fallbeispiel 1: Vorbereitung zur Gestaltung und Planung einer SC	76
5.2 Fallbeispiel 2: Beantwortung ausgewählter Fragestellungen der Transportplanung	79
5.3 Fazit	90

6 Zusammenfassung und Ausblick	92
Literaturverzeichnis	95
Abbildungsverzeichnis	102
Tabellenverzeichnis	103
Abkürzungsverzeichnis	104
Symbolverzeichnis	105
Algorithmenverzeichnis	105
Eidesstaatliche Versicherung	106

1 Einleitung

Die Logistik befindet sich seit Beginn der 1970er Jahre im stetigen Wandel (vgl. [Baumgarten 2008](#), S. 13). Die Globalisierung und steigende Kundenanforderungen sind zentrale Veränderungstreiber, die zur Erweiterung von logistischen Aufgaben führen (vgl. [Werner 2017](#), S. 50–52; vgl. [Schulte 2017](#), S. 26–27). Im Zuge des Wandels steigt die Komplexität der Beziehungen miteinander kooperierender Unternehmen an, um den heutigen Markt- und Kundenanforderungen an Lieferbereitschaft und Individualität der Produkte begegnen zu können (vgl. [Stich et al. 2013c](#), S. 35). Die entstehenden Wertschöpfungsketten werden Supply Chains (SC) genannt und umspannen alle beteiligten Akteure der Wertschöpfungsstufen (vgl. [Werner 2017](#), S. 29). Die umfassende Beschreibung, Analyse und Gestaltung der SC bilden einen zentralen Schlüsselfaktor zur Bereitstellung wettbewerbsfähiger Produkte und Dienstleistungen. Diese meist logistischen Aufgaben sind Teil des Supply-Chain-Managements (SCM) (vgl. [Stich et al. 2013c](#), S. 35). Hierbei entstehen entlang der SC und auf unterschiedlich hohen Abstraktionsebenen vielfältige Aufgaben, die eine Vielzahl an Fragestellungen mit sich bringen. So stellt die Bestimmung der Nachfrage nach einem Produkt in einer untersuchten Region eine Fragestellung der Bedarfsplanung dar (vgl. [Kilger und Wagner 2015](#), S. 127–128; vgl. [Hellingrath et al. 2004](#), S. 106). Insgesamt reichen die Aufgaben von der strategischen Netzwerkgestaltung über die Beschaffungs-, Produktions- und Distributionsplanung bis hin zur Auftragsabwicklung (vgl. [Hellingrath et al. 2004](#), S. 104).

Zur Beantwortung von Fragestellungen wird der Anwender durch rechnergestützte Informationssysteme, bestehend aus einer Software- und einer Speicherkomponente, unterstützt (vgl. [Meier und Kaufmann 2016](#), S. 2). Die Softwarekomponente enthält eine Abfrage- und Manipulationssprache, um auf gespeicherte Daten und Informationen zugreifen zu können (vgl. [Meier und Kaufmann 2016](#), S. 2–3). Die Speicherkomponente dient zur Speicherung und Beschreibung von Daten in einer organisierten Form, wobei in der Praxis meist relationale Datenbanken eingesetzt werden (vgl. [Meier und Kaufmann 2016](#), S. 2–3). Relationale Datenbanken speichern Daten in Tabellenform ein. Jedoch gehen durch die starr vorgegebenen Strukturen nicht selten Informationen, wie bspw. komplexe Beziehungsgeflechte, verloren (vgl. [Robak et al. 2013](#), S. 1216).

Für die Lösung von logistischen Aufgaben existieren verschiedene Softwaremodule, die je nach untergeordneter Fragestellung verwendet werden (vgl. [Hellingrath et al. 2004](#), S. 102–104). Dazu müssen die Daten zuerst an die gegebene Fragestellung angepasst und neu strukturiert werden (vgl. [Bousonville 2017](#), S. 10). So wird in der Tourenplanung die Darstellung der Daten als Graphen in Form von Knoten und Kanten vorausgesetzt, um die SC und Transportverbindungen abbilden und untersuchen zu können (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 221). Hierbei erfolgt der Import und Export der Daten anhand von Schnittstellen, die relationale Daten in Graphen umwandeln und speichern (Graphenrepräsentation) (vgl. [Fleischmann und Kopfer 2018](#), S. 93–94). Auch Standortprobleme werden auf Basis von Graphen gelöst, wobei die traditionelle Datenspeicherung in relationaler Form den Aufwand graphbasierter Lösungsansätze erhöht (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 101–103; vgl. [Aggarwal 2010](#), S. 16).

Zur Erfüllung steigender Anforderungen an die Datenhaltung wurde in den letzten Jahren die Entwicklung neuer Datenbanktechnologien, die nicht auf relationalen Datenbanken basieren, vorangetrieben (vgl. [Meier und Kaufmann 2016](#), S. 221–222). Insbesondere Graphdatenbanken bieten unentdeckte Einsatzpotenziale im SCM, da sie zur Analyse von Beziehungen und Netzwerken eingesetzt werden und Daten in einer Graphenrepräsentation abspeichern (vgl. [Fasel 2016](#), S. 124). Es existieren Modelle (graphbasierte Modelle) und Algorithmen (Graphalgorithmen), die speziell für die Anwendung auf Graphen bzw. einer Graphenrepräsentation entwickelt wurden (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 3–4; vgl. [Aggarwal und Wang 2010b](#), S. 1–3). Durch die Nutzung dieser graphbasierten Verfahren ergeben sich neue Lösungsansätze, bspw. das Graph-Mining, die zur Beantwortung unterschiedlicher Fragestellungen eingesetzt werden können (vgl. [Cook und Holder 2007](#), S. 1–3; vgl. [Aggarwal und Wang 2010b](#), S. 1–3).

Allerdings existiert keine systematische Untersuchung graphbasierter Verfahren im SCM. Lediglich graphbasierte Modelle werden für spezifische logistische Aufgaben, wie die Standortbestimmung oder Tourenplanung, eingesetzt (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 3–7). Darüber hinaus bieten Graphalgorithmen verschiedene Lösungsansätzen, die vorwiegend in stark verflochtenen Netzwerken, bspw. in sozialen Netzwerken, eingesetzt werden (vgl. [Aggarwal und Wang 2010b](#), S. 1–2). Eine SC setzt sich aus Akteuren und Verbindungen in Form von unterschiedlichen Objektflüssen zusammen, so dass sie in der Realität vielmehr ein Netzwerk als eine Kette darstellt (vgl. [Steven und Pollmeier 2008](#), S. 50; vgl. [Göpfert 2013](#), S. 17). Deshalb könnten graphbasierte Lösungsansätze auf logistische Fragestellungen übertragen werden. Aufgrund der Vielzahl an Fragestellungen entsteht die Überlegung, übergeordnete Aufgaben anstatt einzelner Fragestellungen zu betrachten. Eine Vorgehensweise zur Sortierung der Fragestellungen existiert allerdings nicht. Auch graphbasierte Verfahren bilden ein heterogenes Feld bestehend aus Modellen und Algorithmen ohne festgelegter Kategorisierung. Aus den bestehenden Forschungslücken resultieren die nachfolgend beschriebenen Ziele.

Das Hauptziel dieser Arbeit ist die Untersuchung graphbasierter Lösungsansätze zur Beantwortung logistischer Fragestellungen auf Basis übergeordneter Aufgaben. Als Teilziel wird eine Taxonomie zur Klassifizierung logistischer Fragestellungen erstellt. Hierbei sollen logistische Aufgaben in Aufgabenbereiche sowie Aufgabenstellungen unterteilt und durch bestehende Lösungsansätze charakterisiert werden. Ein weiteres Teilziel bildet die Kategorisierung graphbasierter Verfahren in Ober- und Unterkategorien. Auf Basis der Taxonomie und der Kategorisierung wird eine Bewertungsmatrix zur Untersuchung graphbasierter Lösungsansätze erstellt, wodurch Lösungsansätze identifiziert und analysiert werden sollen. Abschließend werden die Unterkategorien und ihre identifizierten Lösungsansätze hinsichtlich der betrachteten Aufgabenstellungen in einer tabellarischen Übersicht zusammengetragen. Validiert werden die Ergebnisse durch zwei Fallbeispiele, um die praktische Tauglichkeit der Taxonomie und ausgewählter Lösungsansätze im SCM zu prüfen.

Die methodische Vorgehensweise leitet sich aus der Zielformulierung ab und basiert auf einer Literaturrecherche zur Darstellung des aktuellen Forschungsstandes. Zuerst werden für die Themenstellung erforderliche Grundlagen zur Logistik, SC und SCM in [Kapitel 2](#) bearbeitet. Hierbei soll ein grundlegendes Begriffsverständnis des SCM, aufbauend auf verwandten logistischen Begriffen, geschaffen werden. Auch werden SCM-Modelle vorgestellt sowie logistische Fragestellungen und Aufgaben beschrieben. Diese werden als wissenstheoretische Grundlagen zur Erstellung der Taxonomie in [Abschnitt 4.1.1](#) benötigt.

Anschließend werden in [Kapitel 3](#) Grundlagen der Datenhaltung sowie graphbasierte Verfahren beschrieben. Einleitend stellen [Abschnitt 3.1](#) und [Abschnitt 3.2](#) begriffliche Grund-

lagen zu Daten, Informationssysteme und relationale Datenbanksysteme (RDBS) vor. Dann werden Graphdatenbanksysteme (GDBS) und ihre einzelne Komponenten in [Abschnitt 3.3](#) erörtert. Diese Grundlagen werden in [Abschnitt 5.2](#) zur Konzeption des zweiten Fallbeispiels benötigt. Abschließend behandelt [Kapitel 3](#) graphbasierte Modelle und Graphalgorithmen, die in [Kapitel 4](#) kategorisiert und untersucht werden.

Auf Basis der erörterten wissenstheoretischen Grundlagen aus [Abschnitt 2.3.1](#) und [Abschnitt 2.3.2](#) werden die zuvor vorgestellten Ziele in [Kapitel 4](#) behandelt. In [Abschnitt 4.1.1](#) wird die Taxonomie durch die Inhalte aus [Abschnitt 2.3](#) zur Klassifizierung logistischer Fragestellungen erstellt. Dann erfolgt die Kategorisierung graphbasierter Verfahren aus [Abschnitt 3.4](#). Aus den Ergebnissen aus [Abschnitt 4.1.1](#) und [Abschnitt 4.1.2](#) wird eine Bewertungsmatrix erstellt. Mithilfe der Bewertungsmatrix erfolgt in [Abschnitt 4.2](#) die Untersuchung der Unterkategorien aus [Abschnitt 4.1.2](#) hinsichtlich ihrer Lösungsansätze für die Aufgabenstellungen aus [Abschnitt 4.1.1](#). Abschließend werden die Ergebnisse der Bewertung ausgewertet und zusammengefasst.

Die Ergebnisse aus [Kapitel 4](#) werden in [Kapitel 5](#) durch zwei Fallbeispiele validiert. Im ersten Fallbeispiel werden logistische Fragestellungen des SCM mithilfe der Taxonomie aus [Abschnitt 4.1.1](#) sortiert. Im zweiten Fallbeispiel werden logistische Fragestellungen einer ausgewählten Aufgabenstellung mithilfe einer Graphdatenbank beantwortet. Das Kapitel schließt in [Abschnitt 5.2](#) mit einem Fazit. In [Kapitel 6](#) erfolgt eine Zusammenfassung und ein Ausblick auf offene Aspekte und Forschungsfragen, die in Zukunft untersucht werden können.

2 Grundlagen des Supply-Chain-Managements

In [Kapitel 2](#) werden wissenschaftliche Grundlagen zur Logistik, SC und SCM bearbeitet. Zuerst werden in [Abschnitt 2.1.1](#) die Entwicklungsphasen der Logistik sowie der Logistikbegriff vorgestellt und in Zusammenhang mit dem SCM gesetzt. Anschließend werden in [Abschnitt 2.1.2](#) SC und ihre Objektflüsse beschrieben. In [Abschnitt 2.2](#) werden Definitionen, Akteure, Ziele und ausgewählte Modelle des SCM vorgestellt. Das Kapitel schließt mit der Vorstellung logistischer Aufgaben und Fragestellungen unterschiedlicher Aufgabenebenen. Die wissenschaftlichen Grundlagen werden zur Erstellung der Taxonomie in [Abschnitt 4.1.1](#) benötigt.

2.1 Logistik und Supply Chains

Veränderungstreiber führen seit den letzten Jahrzehnten zu einer stetigen Weiterentwicklung der Logistik. Der seit den 1980er Jahren zunehmende Kosten- und Zeitdruck sowie die Globalisierung erhöhen die Komplexität logistischer Aufgaben, so dass klassische Logistikaufgaben sukzessiv erweitert und angepasst werden (vgl. [Werner 2017](#), S. 50). Durch die zunehmende Bedeutung logistischer Aktivitäten wurde die Logistik zu einer Managementdisziplin aufgewertet und es entstanden neue Definitionen zur Beschreibung der Aktivitäten (vgl. [Schuh et al. 2013a](#), S. 2). Allerdings haben sich in der Literatur keine einheitlichen Definitionen für die Logistik, SC oder das SCM durchgesetzt (vgl. [Schulte 2017](#), S. 21). Für ein grundlegendes Verständnis der Zusammenhänge logistischer Definitionen erfolgt in [Abschnitt 2.1.1](#) die Erklärung der historischen Entwicklung der Logistik. Anschließend werden in [Abschnitt 2.1.2](#) SC erläutert.

2.1.1 Entwicklungsphasen der Logistik und Logistikbegriff

Die Entwicklung der Logistik bis hin zur Entstehung des SCM lässt sich seit den 1970er Jahren in fünf Entwicklungsphasen einordnen. Ausgangsbasis bildet die klassische Logistik, in der die Optimierung der Kernfunktionen Transportieren, Umschlagen und Lagern ([TuL-Prozesse](#)) im Fokus standen (vgl. [Baumgarten 2004](#), S. 2).

Mit dem einsetzenden Kostendruck, ausgelöst durch die Sättigung vieler Märkte in den 1980er Jahren, entwickelten sich neue Anforderungen an die Qualität und Effizienz der Prozessabwicklung, so dass sich in der zweiten Entwicklungsphase eine prozessorientierte Sichtweise der Logistik etablierte (vgl. [Baumgarten 2008](#), S. 13–14). Die Logistik wurde hierbei als Querschnittsfunktion für die Steuerung und Planung der Beschaffungs-, Produktions- und Vertriebsfunktion verstanden (vgl. [Schulte 2017](#), S. 28).

Das in der zweiten Phase entwickelte Logistikkonzept wurde in der dritten Entwicklungsphase fortgesetzt und vertieft, indem die Bereiche Entwicklung sowie Entsorgung und die Koordination von Güter- und Informationsflüssen in die Planung einbezogen wurden.

Durch diese ganzheitliche Betrachtungsweise konnten Prozesse eines Unternehmens bereichsübergreifend optimiert werden (vgl. [Baumgarten 2004](#), S. 4).

Die in den 1990er Jahren einsetzende vierte Entwicklungsphase markiert den Übergang zu einer unternehmensübergreifenden Sichtweise der Logistik. Erstmals wurden Güter- und Informationsflüsse über die gesamten Unternehmen hinweg koordiniert und Kunden sowie Lieferanten stärker eingebunden (vgl. [Baumgarten 2008](#), S. 13–15). Durch unterschiedliche Informationstechnologien wurde die unternehmensübergreifende Integration der Teilnehmer realisiert, so dass Ineffizienzen einzelner Akteure beseitigt werden konnten. Im Fokus stand die ganzheitliche Zusammenarbeit der Unternehmen zur Leistungssteigerung der SC. Dieses logistische Managementkonzept wird in historischen Definitionen als SCM bezeichnet (vgl. [Göpfert 2013](#), S. 23–26).

Seit Beginn des 21. Jahrhunderts wächst die Komplexität von SC stetig. Einflüsse der Digitalisierung und Globalisierung brechen Branchengrenzen auf und lassen neue Geschäftsfelder entstehen. Durch das Verschwimmen traditioneller Grenzen der Unternehmen überlagern sich unterschiedliche SC, da ein Unternehmen in mehreren SC tätig sein kann (vgl. [Baumgarten 2004](#), S. 6). Um der Komplexität begegnen zu können, werden neue Ansätze der SC-Strukturierung entwickelt. Ziel ist, kleine und flexible SC zu schaffen, die sich dezentral steuern und miteinander kooperieren. Die Entwicklung dieser komplexen SC wird unter der fünften Entwicklungsphase, die bis heute andauert, zusammengefasst (vgl. [Baumgarten 2004](#), S. 6; vgl. [Werner 2017](#), S. 14).

Logistik Je nach Anwendungsfeld und Betrachtungsweise variiert das Verständnis der Logistik, so dass der Begriff und seine Abgrenzung zu verwandten Themengebieten abhängig von Autoren oder Verbänden variieren (vgl. [Schulte 2017](#), S. 3). Eine erste Vorstellung der Logistik und ihrer Ziele erfolgt anhand der 8Rs:

Definition 2.1 Logistik: „Unter Logistik wird die Aufgabe verstanden, das richtige Produkt, zur richtigen Zeit, in der richtigen Menge, am richtigen Ort, in der richtigen Qualität, dem richtigen Kunden und zu den richtigen Kosten zur Verfügung zu stellen. Im heutigen Informationszeitalter bedarf es [...] einer Ergänzung dieser klassischen 7Rs, um eine weitere entscheidende Komponente: „versehen mit den richtigen Informationen.““ ([Hausladen 2016](#), S. 4).

Die Definition verdeutlicht, dass in der Logistik primär der Material- und der dazugehörige Informationsfluss betrachtet werden, mit dem Ziel, maximalen Kundennutzen zu schaffen. Zusammengefasst beschäftigt sich die klassische Logistik mit der Planung, Implementierung und Steuerung von TUL-Prozessen von Produkte oder Dienstleistungen sowie den zugehörigen Informationen entlang ihres Lebenszyklus (vgl. [CSCMP 2013](#), S. 117). Das nachfolgend genutzte Begriffsverständnis orientiert sich an den ersten Entwicklungsphasen der Logistik, so dass eine klare Abgrenzung zwischen Logistik und SCM vorliegt. Durch die Abgrenzung sollen die Unterschiede der beiden Begriffe verdeutlicht werden. Die Einordnung der Logistik in das SCM geschieht durch die nachfolgende Definition des Logistikmanagements:

Definition 2.2 Logistikmanagement: “Logistics management is that part of supply chain management that plans, implements, and controls the efficient, effective forward and reverse flow and storage of goods, services, and related information between the

point of origin and the point of consumption in order to meet customers' requirements. [...] It is involved in all levels of planning and execution-strategic, operational, and tactical. Logistics management is an integrating function which coordinates and optimizes all logistics activities, as well as integrates logistics activities with other functions [...]" (CSCMP 2013, S. 117).

Die unternehmens- und funktionsübergreifende Koordination logistischer Aufgaben wird als Logistikmanagement verstanden. Im Gegensatz zur Logistik erweitern sich somit die betrachteten Aufgaben um Managementaktivitäten, die über die klassischen TUL-Prozesse hinausgehen (vgl. Schuh et al. 2013a, S. 10–11). Jedoch zeigt die Definition, dass das Logistikmanagement einen Teilbereich des SCM darstellt und nicht alle SCM-Aufgabenbereiche abdeckt. Diese Ansicht deckt sich mit der Position der „Unionisten“, die Logistik bzw. das übergeordnete Logistikmanagement als Teildisziplin des SCM betrachten (vgl. Larson und Halldorsson 2004, S. 18–20). Die Sicht der Unionisten wird in dieser Arbeit verfolgt.

2.1.2 Supply Chains

Die SC (dt. Versorgungs- oder Lieferkette) umfasst einen Unternehmensverbund, der über mehrere Wertschöpfungsstufen, angefangen mit der Beschaffung des Rohmaterials bis hin zur Auslieferung des Produktes an den Endkunden, miteinander verbunden ist und unternehmensübergreifend koordiniert wird (vgl. CSCMP 2013, S. 186; vgl. Busch und Dangelmaier 2004a, S. 4). In der deutschsprachigen Literatur wird der Begriff „Supply Chain“ synonym zur Logistikkette, Wertschöpfungskette oder zum Wertschöpfungsnetzwerk verwendet (vgl. Vahrenkamp 2003, S. 1). Die Akteure einer Logistikkette handeln auf Basis einzelwirtschaftlicher Unternehmensentscheidungen und betrachten Material- und Informationsflüsse, wohingegen unternehmensübergreifende Wertschöpfungsketten die ganzheitliche Betrachtung der Unternehmenskette und ihrer Objektflüsse fokussieren (vgl. Schulte 2017, S. 22–23). Eine synonyme Verwendung von Logistikkette und SC findet nachfolgend wegen der auf Unternehmensgrenzen beschränkten Sichtweise von Logistikketten nicht statt. Durch die steigende Komplexität von unternehmensübergreifenden Wertschöpfungsketten wird der Begriff „Wertschöpfungsnetzwerk“ zunehmend zur Veranschaulichung der Vielzahl an Teilnehmern und ihrer Beziehungen in einer SC verwendet. Häufig existieren mehrere Lieferanten (Lieferanten an Lieferanten) sowie Kunden (Kunden von Kunden), so dass die Wertschöpfungskette in der Praxis häufig einem Netzwerk gleicht (vgl. Christopher 2016, S. 3). Aufgrund der inhaltlichen Übereinstimmung von unternehmensübergreifenden Wertschöpfungsketten bzw -netzwerken und SC können diese Begriffe synonym zueinander verwendet werden. Nachfolgend werden zur Vereinheitlichung der Begriffe jedoch nur SC und Netzwerke verwendet. Für ein eindeutiges Begriffsverständnis werden SC wie folgt definiert:

Definition 2.3 Supply Chains: Eine SC bezeichnet den Zusammenschluss kooperierender Unternehmen verschiedener Wertschöpfungsstufen (Zulieferer, Hersteller, Händler und Dienstleister) zu einem organisierten Netzwerk. In der SC werden Unternehmen durch Objektflüsse über Beschaffungs-, Produktions- und Absatzaktivitäten mit vor- und nachgelagerten Akteuren der SC miteinander verbunden, um Wert in Form eines Produktes oder einer Dienstleistung für den Endkunden zu schaffen (vgl. Steven und Pollmeier 2008, S. 50; vgl. Göpfert 2013, S. 17; vgl. Busch und Dangelmaier 2004a, S. 4).

Neben der unternehmensübergreifenden Zusammenarbeit bilden Objektflüsse als Verbindungsglieder der Prozesse sowie Unternehmen einen zentralen Aspekt von SC. Die Flüsse

lassen sich in Material- und Waren-, Informations-, Finanzmittel-, Entwicklungs-, und Rechtfloss einteilen (vgl. [Stewens 2005](#), S. 87).

Objektflüsse Stewens (2005, S. 88) beschreibt den physischen Fluss als Material- und Warenfluss, dessen Komplexität aufgrund der vermehrten Auslagerung von Fertigungsaufgaben und des Wachstums lokaler Märkte stetig zunimmt. Im Rahmen des SCM wird die Gestaltung eines durchgängigen Flusses behandelt, in dem Objekte flussabwärts, d. h. von der Rohstoffgewinnung (Quelle) bis hin zum Endkunden (Senke), und flussaufwärts entlang der SC fließen. Für einen unterbrechungsfreien und reibungslosen Fluss ist die unternehmensübergreifende Koordination von TUL-Prozessen, Kapazitäten und Beständen notwendig (vgl. [Pfohl 2018](#), S. 8–9; vgl. [Stewens 2005](#), S. 88).

Der Informationsfluss ist eng mit dem physischen Fluss verbunden, da erst die richtigen Informationen die gewollten Güterflüsse auslösen (vgl. [Göpfert 2013](#), S. 17). Stromaufwärts werden vor allem Bestelldaten übermittelt, wohingegen stromabwärts die Übermittlung von Lieferdaten erfolgt. Zu den Gestaltungszielen des Informationsflusses zählen eine hohe Fließgeschwindigkeit, Datensicherheit und die Vermeidung von Medienbrüchen sowie Informationsverlusten (vgl. [Stewens 2005](#), S. 88–89). Zahlungsverbindlichkeiten zwischen Verkäufer und Käufer bzw. Lieferant und Abnehmer bilden die betrachteten Objekte des Finanzmittelflusses (vgl. [Göpfert 2013](#), S. 18). Die Fließgeschwindigkeit von Finanzflüssen wird ähnlich wie bei Informationsflüssen durch die verwendeten Übertragungsmedien sowie die Interoperabilität von Informationssystemen bestimmt (vgl. [Stewens 2005](#), S. 89).

Nach [Stewens](#) (vgl. [2005](#), S. 90–91) wurden bis in die 1990er Jahre die Planung, Koordination und Optimierung des physischen Flusses sowie des Informations- und Finanzmittelflusses fokussiert. Erst mit Zunahme der Zusammenarbeit der Akteure und der Auslagerung von Entwicklungsprozessen wuchs die Bedeutung des Entwicklungsflusses. Das Gestaltungsziel des Entwicklungsflusses ist die Bestimmung der optimalen Entwicklungstiefe und die Gewährleistung eines reibungslos funktionierenden und schnellen Flusses. Im Gegensatz zu den anderen Flüssen beschreibt Stewens den Entwicklungsfluss nicht als Verbindungsglied von Quellen und Senken, da vielmehr ein wechselseitiger Austausch zwischen den beteiligten Akteuren stattfindet.

[Stewens](#) (vgl. [2005](#), S. 90–92) beschreibt den Rechtfloss als Konsequenz der zunehmenden Verflechtung von Produktions- und Entwicklungsleistungen von Unternehmen einer SC. Der Rechtfloss befasst sich mit Fragestellungen rund um Rechte und Pflichten, die sich durch das gemeinsam erzeugte Produkt ergeben. So werden im Rechtfloss bspw. Verfügungsrechte an Patenten oder Pflichten bezüglich der Rücknahme von Produkten festgelegt. Nachfolgend werden eine beispielhafte SC und ihre Objektflüsse in [Abbildung 2.1](#) dargestellt. Hierbei werden die Akteure durch Vierecke, die Objektflüsse durch Kanten dargestellt.

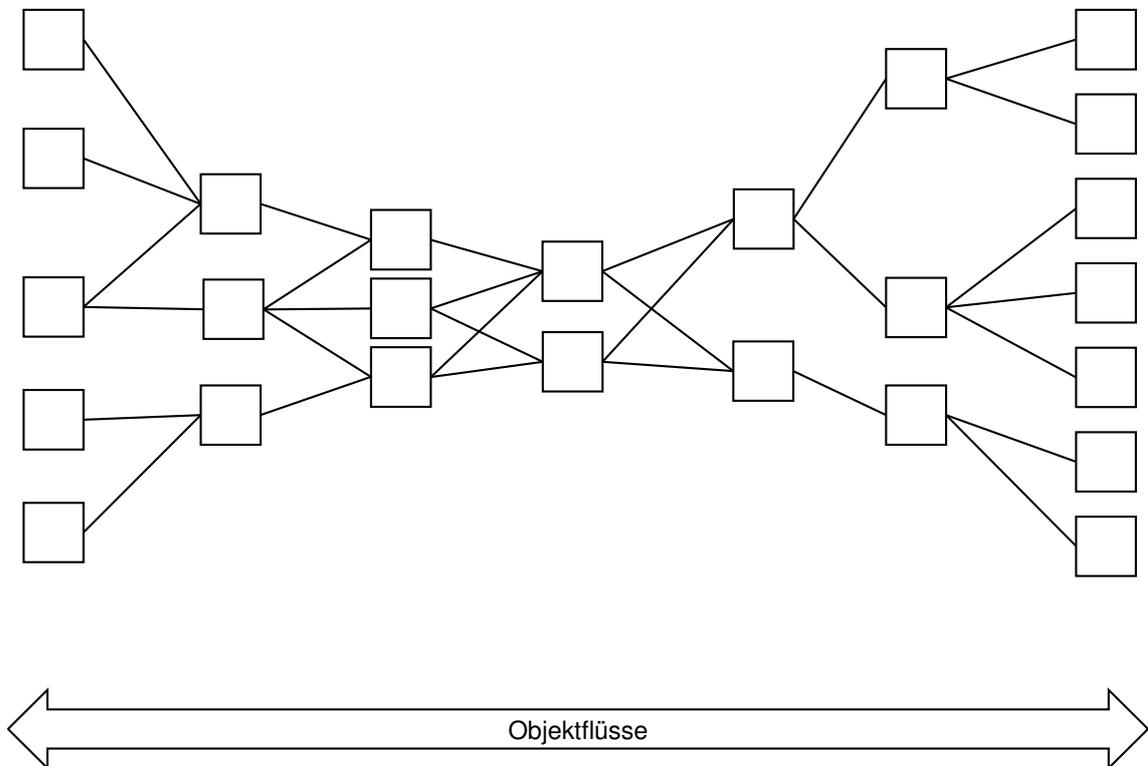


Abbildung 2.1: Beispielhafte Darstellung einer SC in Anlehnung an [Busch und Dangelmaier \(2004b, S. 5\)](#)

Laut (vgl. [Göpfert 2013, S. 29](#)) ist die Planung, Koordination, Gestaltung und Optimierung der Objektflüsse eine der Kernaufgaben des SCM. Viele unternehmensübergreifende Fragestellungen und Herausforderungen lassen sich auf einzelne Objektflüsse zurückführen, so dass die Flüsse als einer der unmittelbaren Auslöser für die Entstehung des SCM betrachtet werden.

2.2 Supply-Chain-Management

Nachdem ein grundlegendes Verständnis der Logistik und SC erörtert wurde, erfolgt die Definition des SCM. Dazu werden in [Abschnitt 2.2.1](#) unterschiedliche Begriffe des SCM vorgestellt, bevor ein gültiger SCM-Begriff definiert wird. Anschließend werden Akteure und Ziele sowie ausgewählte SCM-Modelle vorgestellt.

2.2.1 Begriffsbestimmung

In Theorie und Praxis hat sich keine einheitliche Definition des SCM durchgesetzt, jedoch lassen sich gemeinsame Erklärungsansätze erkennen. So besteht Einigkeit darüber, dass das SCM auf die Fluss- und Prozessorientierung der Logistikkonzeption basiert und die Logistik zum Verständnis des SCM beiträgt. Auch der integrative Koordinationsgedanke

von Lieferanten und Kunden spiegelt sich in den meisten Definitionen wider (vgl. [Schulte 2017](#), S. 21). Jedoch variieren die SCM-Begriffe in ihren Betrachtungsschwerpunkten und ihrer Abgrenzung zur Logistik.

[Stewens](#) (vgl. [2005](#), S. 5) führt die Diskrepanz der unterschiedlichen SCM-Begriffe auf mehrere Teilaspekte zurück. So lässt sich die Diskrepanz durch die Neuartigkeit des SCM im Vergleich zu anderen Forschungsgebieten begründen. Auch berühren jeweilige Teilaspekte unterschiedliche Fachgebiete, so dass Forschungsschwerpunkte je nach Fachgebiet variieren und das SCM somit einen vielschichtigen Untersuchungsgegenstand darstellt.

[Schuh et al.](#) (vgl. [2013a](#), S. 14) beschreibt das SCM als prozessorientierten Managementansatz, der logistische Aktivitäten entlang der SC anhand ihrer Material- und Informationsflüsse unternehmensübergreifend koordiniert. Dies entspricht einer Erweiterung des Logistikbegriffs um eine interorganisationale Betrachtungsweise, wobei die zu koordinierenden Objekte und Aktivitäten identisch sind. Breitere Auffassungen des Begriffes erweitern die behandelten Aktivitäten auf weitere Unternehmensbereiche wie bspw. Marketing, Produktion oder betriebliche Aktivitäten (vgl. [Mentzer et al. 2008](#), S. 38).

Andere Definitionen setzen die Betrachtungsschwerpunkte auf der Flussgestaltung oder der Koordination von Beziehungen eines SCM. So beschreibt [Göpfert](#) (vgl. [2013](#), S. 32) das SCM als eine Konzeption für Unternehmensnetzwerke zur Erschließung unternehmensübergreifender Erfolgspotenziale mittels effektiver und effizienter Objektflüsse. [Christopher](#) (vgl. [2005](#), S. 18) legt den Fokus auf die Beziehungen der Akteure und beschreibt SCM als Koordination von vor- und nachgelagerten Beziehungen von Lieferanten und Kunden, um einen hohen Kundennutzen kostengünstig zu erzeugen. Die zuvor aufgeführten Aspekte lassen sich in nachfolgender Definition zusammenfassen:

Definition 2.4 Supply-Chain-Management: „Supply Chain Management ist die unternehmensübergreifende Koordination und Optimierung der Material-, Informations- und Wertflüsse über den gesamten Wertschöpfungsprozess von der Rohstoffgewinnung über die einzelnen Veredelungsstufen bis hin zum Endkunden mit dem Ziel, den Gesamtprozess unter Berücksichtigung der Kundenbedürfnisse sowohl zeit- als auch kostenoptimal zu gestalten“ ([Arndt 2008](#), S. 47).

In der von Arndt aufgeführten Definition werden Objektflüsse, Akteure und ihre Beziehungen, Prozesse sowie Ziele zusammengeführt, so dass sie eine holistische Definition des SCM liefert und nachfolgend als Grundverständnis dienen soll. Nachdem die Objektflüsse in [Abschnitt 2.1.1](#) erläutert wurden, erfolgt eine Erläuterung der Akteure und Ziele des SCM, bevor zwei SCM-Modelle vorgestellt werden.

Akteure und Ziele Nach [Werner](#) (vgl. [2017](#), S. 29) bilden Lieferanten, Hersteller, Händler, Distributoren, Dienstleister und Kunden ein heterogenes Interessentenbündel der SC. Die SC bildet sich durch komplexe Beziehungsgeflechte aus und befindet sich in einem latenten Spannungsverhältnis der beteiligten Akteure. Einerseits erhoffen sich die einzelnen Unternehmen durch unternehmensübergreifende Koordination eine gesteigerte Wettbewerbsfähigkeit, andererseits streben die rechtlich selbstständigen Organisationen nach Autonomie. Insgesamt verfolgen die Teilnehmer einer SC u. a. folgende Ziele (vgl. [Schuh et al. 2013c](#), S. 212):

- Verbesserte Erfüllung der Kundenwünsche
- Synchronisation von Versorgung und Bedarf

- flexible und bedarfsgerechte Produktion
- effizientere Ressourceneinsätze
- Abbau von Beständen
- Realisierung von Zeit-, Kosten- und Qualitätsvorteilen entlang der gesamten Wertschöpfungskette

Zur Erreichung der Ziele ist die unternehmensübergreifende Synchronisation der SC, so dass ein gemeinsames Verständnis der Abläufe innerhalb einer SC geschaffen werden müssen (vgl. [Schönsleben 2016](#), S. 14). Dies wird durch die Nutzung der nachfolgend vorgestellten Modellen unterstützt.

2.2.2 Ausgewählte Modelle des Supply-Chain-Managements

Das SCOR-Modell Das Supply-Chain-Operations-Reference-Modell (SCOR-Modell) des Supply Chain Councils dient zur Standardisierung, Beschreibung und Bewertung von unternehmensinternen und -übergreifenden Prozessketten, um ein gemeinsames Verständnis der Abläufe innerhalb einer SC zu schaffen (vgl. [Schönsleben 2016](#), S. 14). Das SCOR-Modell wurde mit dem Ziel entworfen, ein einheitliches, vergleichbares und bewertbares Prozessmodell zum Informationsaustausch zwischen Unternehmen einer SC zu schaffen (vgl. [Schulte 2017](#), S. 783). Das Modell wurde in den vergangenen Jahren stetig weiterentwickelt und ergänzt, so dass mittlerweile Version 12.0 veröffentlicht wurde (vgl. [Schröder 2019](#), S. 16–17).

Das SCOR-Modell beinhaltet vier hierarchisch aufgebaute Ebenen: die Netzwerkebene (oberste Ebene), die Konfigurationsebene (Ebene 2), die Gestaltungsebene (Ebene 3) und die Implementierungsebene (Ebene 4). Auf oberster Ebene unterscheidet man sechs Kernprozesse, die sich in den untergeordneten Ebenen in Prozesskategorien sowie Prozesselemente aufteilen. Mit jeder Ebene erhöht sich der Detaillierungsgrad der Konkretisierung von Prozessen und Aktivitäten (vgl. [Eßig et al. 2013](#), S. 289–290). Für ein grundlegendes Verständnis ist jedoch die oberste Ebene ausreichend, so dass die weiteren Ebenen nachfolgend nicht näher erläutert werden und der Fokus auf den sechs Kernprozessen der obersten Ebene liegt. Die sechs Prozesse gliedern sich in planen (plan), beschaffen (source), herstellen (make), liefern (deliver), zurückführen (return) und unterstützen (enable). Sie beschäftigen sich auf Netzwerkebene (oberste Ebene) mit unternehmensübergreifenden Fragestellungen der SC. Die Prozesse sind wie folgt definiert (vgl. [Corsten und Gössinger 2008](#), S. 150; vgl. [Werner 2017](#), S. 71):

- **Das Planen** umfasst alle vorbereitenden Aktivitäten über die gesamte SC und die anderen Kernprozesse hinweg, so dass es einen übergeordneten Charakter erhält. Hierzu zählen neben der Planung der Infrastruktur, die Ressourcenplanung und -gestaltung (Aggregate, Personal, Materialien), das Ermitteln von Nachfrageanforderungen, die Planung von Beständen und das Produktmanagement (Make-or-Buy-Entscheidungen, Produkteinführungs- sowie Eliminationszeitpunkte).
- **Die Beschaffung** umfasst alle Aktivitäten, die mit dem Erwerb, dem Erhalt, der Prüfung und der Bereitstellung der eingehenden Materialien entstehen (Beschaffungsquelle, Liefervertragsgestaltung, etc.)
- **Das Herstellen** umfasst den gesamten Prozess zur Produkterstellung (Kapazitätssteuerung, Zwischenlagerung, Verpackung, Übergabe, etc.).

- **Das Liefern** umfasst das Kundenauftragsmanagement (Bestellerfassung, Fakturierung, Inkassowesen), die Bewirtschaftung des Fertigwarenlagers sowie die Distributivvorgänge.
- **Das Zurückführen** umfasst alle administrativen Tätigkeiten, die mit der Rückgabe oder dem Austausch von Produkten verbunden sind. Auch Instandhaltungs- oder Reparaturleistungen zählen zum Kernprozess der Rückführung.
- **Das Unterstützen** bildet seit der Version 11.0 den sechsten Kernprozess. Die Unterstützungsprozesse unterstützen die Planung und Durchführung der zuvor beschriebenen Kernprozesse durch die Überwachung von Informationen, Ressourcen und der Einhaltung von Verträgen entlang der SC (vgl. [Sürle und Reuter 2015](#), S. 34).

Durch die unternehmensübergreifende Prozessdarstellung werden die Akteure einer SC auf oberster Ebene miteinander verknüpft, so dass ein gemeinsames Verständnis der Prozesse entsteht und die Kommunikation der Unternehmen untereinander erleichtert wird (vgl. [Schulte 2017](#), S. 783). Kennzahlen und Best-Practice-Analysen unterstützen den objektiven Vergleich und die Prozessoptimierung einzelner Unternehmen der SC. Auch Einsatzgebiete und Funktionalitäten von Software in der SC lassen sich durch das SCOR-Modell identifizieren (vgl. [Eßig et al. 2013](#), S. 291; vgl. [Schulte 2017](#), S. 783).

[Eßig et al.](#) (vgl. [2013](#), S. 291) kritisieren jedoch, dass der Kern der Prozesshierarchie auf einzelne produzierende Unternehmen bezogen ist und unternehmensübergreifende Aspekte durch die Verknüpfung der unternehmensbezogenen Modellspezifikationen sowie mittels genereller Kennzahlenstandards erreicht werden. Auf Netzwerkebene sind die Kernprozesse je nach Unternehmensschwerpunkt unzureichend für die Beschreibung der Prozesse geeignet und auf Implementierungsebene existieren keine detaillierten Modellierungselemente (vgl. [Schulte 2017](#), S. 787). Auch werden die Aufgabenbereiche von Software-Funktionalitäten im SCOR-Modell nicht spezifisch genug abgebildet. Um dem entgegenzuwirken wurde deshalb das im nachfolgenden Abschnitt beschriebene Aufgabenmodell entwickelt (vgl. [Hellingrath et al. 2004](#), S. 103).

Das Aufgabenmodell Aufbauend auf dem SCOR-Modell entwickelten zwei Fraunhofer-Institute das Aufgabenmodell, um unterschiedliche Aufgabenspektren durch gezielten Softwareeinsatz zu unterstützen. Das Modell zerlegt die Planungselemente (Plan Supply Chain, Plan Source, Plan Make und Plan Deliver) des SCOR-Modells und beschreibt diese bezüglich der möglichen Funktionalitäten von Softwaresystemen (vgl. [Hellingrath et al. 2004](#), S. 102).. So kann das Modell zur Untersuchung, Analyse und Auswahl von SCM-Software zur Unterstützung von logistischen Aufgaben und ihren Fragestellungen herangezogen werden. Das Aufgabenmodell gliedert sich in die drei Hauptaufgabenebenen: Die Gestaltungsebene, die Planungsebene und die Ausführungsebene (vgl. [Hellingrath et al. 2004](#), S. 102–103). Nachfolgend werden die Aufgabenebenen anhand von [Abbildung 2.2](#) erläutert. Eine detaillierte Beschreibung der einzelnen Aufgaben erfolgt in diesem Abschnitt nicht, da ein Grundverständnis des Modellaufbaus im Vordergrund steht.

[Hellingrath et al.](#) (vgl. [2004](#), S. 104) beschreiben die Aufgabe der strategischen Netzwerkgestaltung als die kostengünstige Auslegung und Gestaltung der SC unter Einbeziehung der SCM-Strategie und -Ziele eines Unternehmens. Eine zentrale Fragestellung ist die Beurteilung von Investitionsvorhaben unter Berücksichtigung der verschiedenen Elemente einer SC, bspw. SC-Partner, Standorte sowie Größe und Anzahl von Werken, Lagern und Distributionszentren. Die kostenmäßige Beurteilung von Veränderungen innerhalb der SC, der Aufbau neuer Produktionswerke oder die Planung von Distributionszentren erfolgen auf

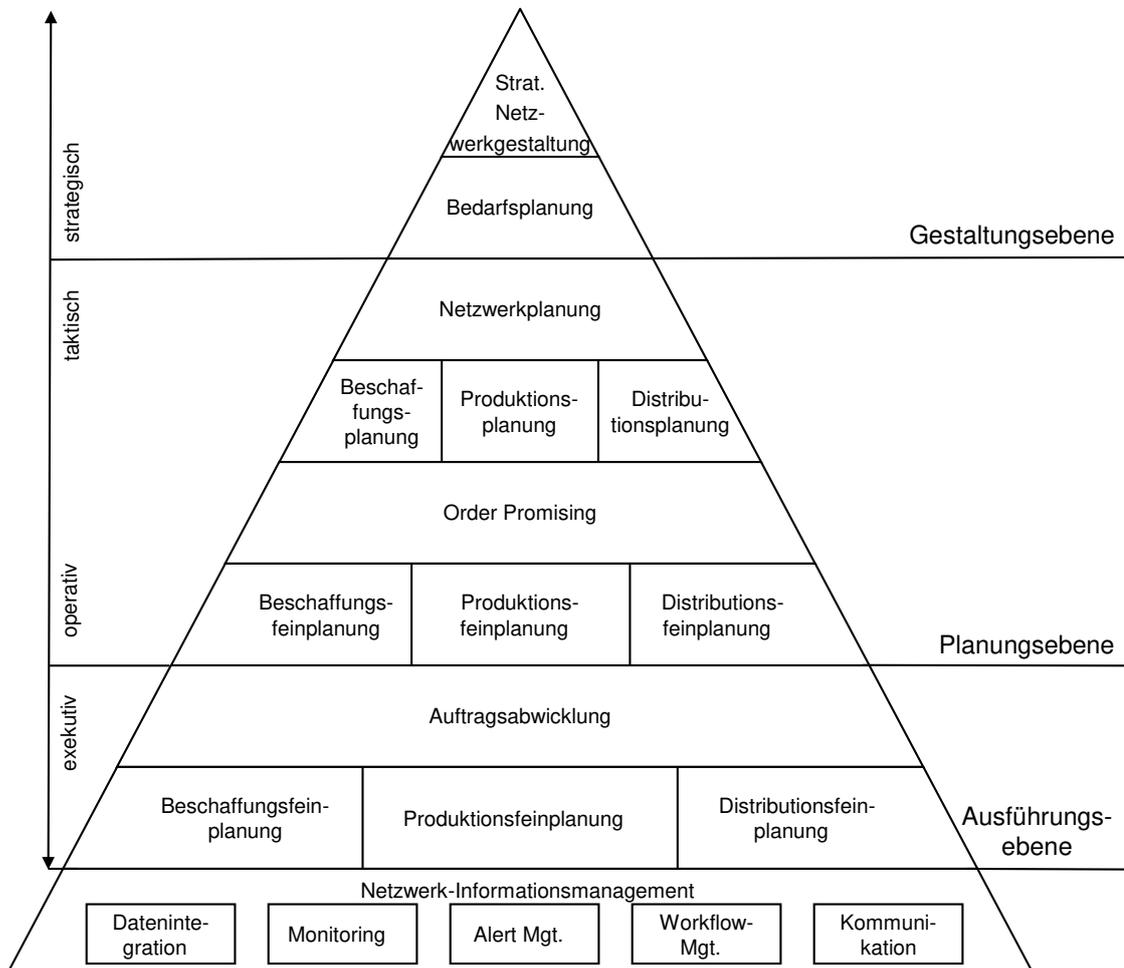


Abbildung 2.2: SCM-Aufgabenmodell in Anlehnung an Schulte (2017, S. 791) nach Hellingrath et al. (2004)

Basis von SCM-Informationssystemen. Der Planungshorizont kann über mehrere Monate bis hin zu mehreren Jahren reichen (vgl. Schulte 2017, S. 793).

Die Planungsebene beschreibt die zweite Hauptebene und ist für den taktischen und operativen Umsetzungsprozess der Gestaltungsebene verantwortlich. Es erfolgt eine kollaborative Entscheidungsfindung, in der Bedarfe, Bestände, Kapazitäten oder Kapazitätszuordnungen der Akteure abgeglichen werden (vgl. Werner 2017, S. 88). Wie in Abbildung 2.2 dargestellt, umfassen die Planungsaufgabe die Netzwerk-, Produktions-, Beschaffungs- und Distributionsplanung, Order Promising sowie die Beschaffungs-, Produktions- und Distributionsfineplanung. Die Fragestellungen der Planungsebene unterscheiden sich bezüglich lang-, mittel- und kurzfristigen Zeithorizonten, des (nicht) vorhandenen Bezugs zu Kundenaufträgen sowie des Umfangs der einbezogenen Teile der SC (vgl. Schulte 2017, S. 793; vgl. Hellingrath et al. 2004, S. 105).

Nach Abschluss der Planungstätigkeiten werden ausführende logistische Aufgaben in der Ausführungsebene initiiert (vgl. Werner 2017, S. 92). In der Ausführungsebene werden alle Aufgaben zusammengefasst, die die Auftragsabwicklung und Kontrolle der SC ermöglichen bzw. der operativen Prozessabwicklung dienen. Der Zeithorizont der Planung beträgt wenige Stunden bis Tage (vgl. Schulte 2017, S. 796). Die Aufgaben umfassen

das Lagermanagement, die Produktions- und die Transportabwicklung (vgl. [Hellingrath et al. 2004](#), S. 112). Die auf dieser Ebene eingesetzten Softwareprogramme unterstützen und überwachen zum einen die Ausführungsaktivitäten, zum anderen sind sie für das Netzwerk-Informationsmanagement verantwortlich (vgl. [Schulte 2017](#), S. 796). Insgesamt beschreibt und unterteilt das Aufgabenmodell die verschiedenen Aufgabenbereiche von SCM-Software zur vereinfachten Bestimmung benötigter Software-Funktionalitäten (vgl. [Hellingrath et al. 2004](#), S. 103). Die operative Feinplanung und die unterste Ebene beschreiben Ausführungsaktivitäten, die im SCM von untergeordneter Bedeutung sind und unternehmensintern gelöst werden. Eine nähere Betrachtung der Ausführungsebene findet deshalb in den nächsten Abschnitten nicht statt.

2.3 Logistische Aufgaben und bestehende Lösungsansätze

In [Abschnitt 2.1](#) und [Abschnitt 2.2](#) wurden Begriffe im Rahmen des SCM erläutert und definiert. So wurde festgelegt, dass logistische Aufgaben im Rahmen des Logistikmanagements bearbeitet werden und dass das Logistikmanagement Teil des SCM ist. Deshalb wird nachfolgend auf den Zusatz des SCM-Begriffs verzichtet.

[Abschnitt 2.3](#) behandelt logistische Aufgaben und ihre Fragestellungen. Hierbei werden die Aufgaben nach der Gestaltungs- und Planungsebene aus [Abbildung 2.2](#) vorgestellt. In [Kapitel 4](#) werden die Aufgaben zur Bildung der Taxonomie analysiert und in Aufgabenbereiche sowie -stellungen eingeteilt, so dass logistische Fragestellungen in die Aufgabenstellungen klassifiziert werden können.

2.3.1 Aufgaben der Gestaltungsebene

In der obersten Ebene des Aufgabenmodells aus [Abbildung 2.2](#) werden strategische und somit längerfristige Entscheidungen im Planungszeitraum von mehreren Jahren hinsichtlich der logistischen Struktur von SC getroffen (vgl. [Stich et al. 2013c](#), S. 47). Hierbei setzt sich eine SC aus miteinander verbundenen Netzwerkknoten zusammen. Je nach Betrachtungsebene stellen Knoten und ihre Kanten entweder Funktionen, Institutionen, Ausstattungen oder Prozesse dar (vgl. [Muchna et al. 2018](#), S. 121). In der strategischen Netzwerkgestaltung werden Planungsalternativen mithilfe von Software hinsichtlich einer kostenoptimalen Auslegung der Netzwerkelemente gebildet und bewertet (vgl. [Hellingrath et al. 2004](#), S. 104). Die Entscheidung für eine Gestaltungsalternative basiert auf dem Ziel, den Unternehmenswert zu steigern und die Gesamtkosten zu reduzieren. Durch die Festlegung der Netzwerkstruktur stellen die gesamte Netzwerkkapazität und ihre Standorte feste Planungsgrößen für die oberste Ebene dar (vgl. [Stich et al. 2013c](#), S. 41). Die Gestaltung von Netzwerkkapazitäten ergibt sich auf Basis langfristiger Bedarfsprognosen und des damit einhergehenden Netzwerkbedarfs, d. h. die Anzahl an benötigten Standorten und ihre Konfiguration zueinander (vgl. [Stich et al. 2013c](#), S. 60). Neben interner Maßnahmen können durch das Hinzufügen neuer SC-Partner größere Kapazitäten geschaffen werden (vgl. [Corsten und Gössinger 2008](#), S. 199). Auch die Optimierung der Potenzialeinheiten, bspw. die Leistungssteigerung von Distributionssystemen, führt zu einer Vergrößerung der Netzwerkkapazitäten (vgl. [Stich et al. 2013c](#), S. 60–63).

Neben der Planung des Netzwerkbedarfs befasst sich die strategische Netzwerkgestaltung mit der Zuordnung von Standortrollen. Dazu wird entschieden, welchen Zweck ein Werk erfüllen soll, so dass sich die Zuordnung von Standortrollen auf die gesamte SC auswirkt.

So ist je nach Standortrolle z. B. zwischen Produktvielfalt oder -fokus, produktbezogener oder funktionaler Spezialisierung und verschiedenen Flexibilitätsgraden zu wählen (vgl. [Chopra und Meindl 2014](#), S. 81). Ziel ist die Gestaltung eines ganzheitlich abgestimmten Netzwerkes mit optimal entwickelten Kernkompetenzen der Standorte. Es wird eine ausbalancierte SC angestrebt, die durch die einzelnen Kernkompetenzen der Standorte ihr Gesamtoptimum erreicht (vgl. [Friedli et al. 2013](#), S. 26–27). Innerhalb der Standortgestaltung werden bspw. folgende Fragestellungen beantwortet (vgl. [Stich et al. 2013c](#), S. 42; vgl. [Daskin und Maass 2019](#), S. 380):

- An welchen Orten sind neue Standorte einzurichten?
- Wie viele Standorte sollten in der SC genutzt werden?
- Zwischen welchen Standorten sollten welche Beziehungen für die Beschaffung, Produktion und Distribution von Rohstoffen, Zwischen- und Fertigprodukten verwirklicht werden?
- Inwiefern ist eine bestehende SC zu rekonfigurieren, um neue wettbewerbsfähige Produkte herzustellen?

Die strategische Netzwerkgestaltung unterteilt sich in die Erstellung von Planungsszenarien, die Entwicklung von Gestaltungsalternativen und die Bewertung sowie Auswahl der Alternativen (vgl. [Fleischmann und Koberstein 2015](#), S. 119). Die eingesetzten SCM-Softwares enthalten Funktionalitäten zur Modellierung der SC und der, monetären und logistischen Simulation sowie zur Bewertung unterschiedlicher Netzwerkkonfigurationen (vgl. [Fink 2018](#), S. 237; vgl. [Busch et al. 2003](#), S. 39).

Nach [Stich et al.](#) (vgl. [2013c](#), S. 58) erfolgt die Netzwerkgestaltung in mehreren iterativen Schritten. In einem ersten Schritt wird die SC modelliert, um die SC-Strukturen zu erfassen und zu verstehen. Anschließend werden unterschiedliche Szenarien zur möglichen Entwicklung der Unternehmensumwelt und ihres Einflusses auf die SC erstellt und mittels einer Szenarioanalyse simuliert. Hierbei werden neben dem zu erwartenden Trendszenario auch optimistische und pessimistische Szenarien gebildet. Zur Generierung der Szenarien werden ökonometrische und stochastische Methoden verwendet, wie bspw. Verteilungsfunktionen, stochastische Prozesse oder Zeitreihenanalysen. Durch Parametervariationen, bspw. der Änderung von Steuersätzen, entstehen unterschiedliche Planungsumgebungen (vgl. [Fleischmann und Koberstein 2015](#), S. 118–119).

Nach der Erstellung der Planungsalternativen werden die Gestaltungsalternativen gebildet (vgl. [Fleischmann und Koberstein 2015](#), S. 119). Diese Alternativen basieren auf der Modifizierung der ursprünglichen SC durch das Hinzufügen von Standorten (geografische Standortbestimmung) oder der Rekonfiguration der SC (vgl. [Stich et al. 2013c](#), S. 70). Aufgrund der Vielzahl an Modellen der Standorttheorie liegt der Fokus nachfolgend auf quantitative Modelle der Graphentheorie (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 101). Die optimale Netzwerkkonfiguration zwischen Lieferanten, Produktionswerken, Distributionszentren und Kunden wird durch die kostengünstige Auslegung des Materialflusses erreicht (vgl. [Häberle und Kilger 2015](#), S. 366).

Zur Bewertung der Gestaltungsalternativen werden unterschiedliche Bewertungsverfahren und Zielgrößen, bspw. geringe Kosten oder Lieferzeiten, herangezogen (vgl. [Stich et al. 2013c](#), S. 59). Dazu eignen sich heuristische Bewertungsverfahren, die qualitative, quantitative, zeitliche und örtliche Faktoren berücksichtigen. So können bspw. Checklisten oder Nutzwertanalysen zur Vorauswahl und Bewertung potenzieller Standorte oder SC-Partnerschaften verwendet werden (vgl. [Stich et al. 2013c](#), S. 71–72). Typische Standortfaktoren

sind bspw. Transportkosten vom Standort zu den Absatzmärkten, Grundstückskosten oder die Güte der Infrastruktur des Standorts (vgl. [Stich et al. 2013c](#), S. 62). SC-Partner werden anhand strategischer, struktureller und kultureller Merkmale wie bspw. Kooperationskultur oder Vertrauenswürdigkeit ausgewählt (vgl. [Corsten und Gössinger 2008](#), S. 199). Nach [Friedli et al.](#) (vgl. [2013](#), S. 26–27) basiert die Festlegung von Standortrollen ebenfalls auf Bewertungsverfahren, die sowohl qualitative als auch quantitative Faktoren berücksichtigen. So erfolgt die Zuordnung durch Rollenmodelle, die eine Einteilung auf Basis von zuvor festgelegten Kriterien, bspw. strategischen Standortvorteilen, technologischem Know-How oder standortbezogenen Kernkompetenzen, vornehmen. Nach dem Auswahlprozess können neue Zielgrößen und Szenarien entwickelt sowie Gestaltungsalternativen modifiziert werden, so dass der Prozess bei Bedarf wiederholt wird. Der Gestaltungsprozess einer SC ist somit iterativ (vgl. [Fleischmann und Koberstein 2015](#), S. 118–119).

2.3.2 Aufgaben der Planungsebene

Im Rahmen der Planungsebene werden die Kapazitäten entlang der SC zur Auftragserfüllung festgelegt (vgl. [Hellingrath et al. 2004](#), S. 105). Der Planungszeitraum der mittleren Ebene erstreckt sich zwischen einem Vierteljahr bis zu zwei Jahren. Auf Basis der zuvor festgelegten Netzwerkkonfiguration wird die Maximierung des Gewinns der SC fokussiert. Die Planungsphase beginnt mit der Bedarfsplanung, in der eine Prognose hinsichtlich der Nachfrage, Kosten, Preise in verschiedenen Märkten und weiterer relevanter Planungsgrößen erfolgt (vgl. [Chopra und Meindl 2014](#), S. 30). Anschließend werden Planungstätigkeiten zur Kapazitätszuordnung entlang der SC durchgeführt (vgl. [Hellingrath et al. 2004](#), S. 105). Weiterhin werden die Beschaffungs-, Produktions- und Distributionsplanung sowie das Order Promising auf der Planungsebene behandelt (vgl. [Schulte 2017](#), S. 791).

Bedarfsplanung Die Bedarfsplanung ermittelt und prognostiziert die nachgefragten Produktmengen durch statistische Prognoseverfahren (vgl. [Corsten und Gössinger 2008](#), S. 165). Kurzfristige Bedarfsprognosen erfolgen durch die Auswertung vorliegender Bestellungen, die mithilfe von Software über die verschiedenen Stufen der SC entnommen werden. Der Fokus der Bedarfsplanung liegt jedoch auf der Prognose von mittel- und langfristigen Bedarfen, die auf der Auswertung von Vergangenheitsdaten, Saisonalitäten, Promotionen oder Trends im Gesamtmarkt basiert (vgl. [Hellingrath et al. 2004](#), S. 105; vgl. [Busch et al. 2003](#), S. 39). Im Rahmen der Bedarfsplanung werden u. a. folgende Fragestellungen beantwortet (vgl. [Kilger und Wagner 2015](#), S. 127–128; vgl. [Hellingrath et al. 2004](#), S. 106):

- Wie hoch ist die Nachfrage nach einem bestimmten Produkt in einem bestimmten Zeitraum in einer bestimmten Region?
- Wie verändert sich die Nachfrage nach einem bestimmten Produkt bei einer Preisänderung?
- Lassen sich Nachfragemuster auf Basis der Vergangenheitsdaten erkennen?

Zur Beantwortung der Fragestellungen existieren vergangenheits- und zukunfts-basierte Prognoseverfahren. Vergangenheitsbasierte Verfahren prognostizieren den zukünftigen Bedarf mithilfe statistischer Verfahren und vergangenheitsbezogener Daten wie bspw. Verbraucherstatistiken (vgl. [Schönsleben 2016](#), S. 406–507). Zukunfts-basierte Vorhersageverfahren berücksichtigen aktuelle Daten, bspw. aus festen Bestellmengen oder Kundenbefragungen (vgl. [Schönsleben 2016](#), S. 406–507). Softwares für die Bedarfsplanung enthalten meist eine Kombination aus Methoden beider Prognosegruppen, die auf den gesammelten

Daten angewendet werden (vgl. [Wannenwetsch 2014](#), S. 494).

Bullwhip-Effekt Durch das Verhalten zahlreicher SC-Akteure lässt sich eine Verzerrung der Nachfrageinformationen beobachten, so dass die Bedarfsplanung verkompliziert wird. Vorherrschende Intransparenz und mangelnde Absprache zwischen den jeweiligen Partnern einer SC führen zu einem Aufschaukeln der Nachfrage entlang der Wertschöpfungskette (vgl. [Schuh et al. 2013a](#), S. 25). Dieses Phänomen wird als „Bullwhip-Effekt“ bezeichnet und fällt umso größer aus, je schlechter die Material-, Waren und Informationsflüsse zwischen den Unternehmen koordiniert werden (vgl. [Göpfert 2013](#), S. 30). Der Effekt resultiert aus Informationsdefiziten über zukünftige Bedarfe, problembehafteten Beschaffungsprozessen, Reduzierung bestellfixer Kosten durch Bedarfsbündelungen und Preisschwankungen (vgl. [Werner 2017](#), S. 48). Folgende Handlungsempfehlungen werden zur Beseitigung von Informationsverzögerungen und zur Minderung des Bullwhip-Effekts beschrieben (vgl. [Schuh et al. 2013a](#), S. 25–26):

- Direkter Austausch der Nachfrageinformationen
- Unternehmensübergreifende Erstellung von Bedarfsprognosen
- Direkte Weitergabe über Eingangsfaktoren (z. B. geplante Rabatte)
- Senkung von Losgrößenbündelungen durch reduzierte fixe Bestellkosten (automatisierte Bestellprozesse, optimierte Transportplanung)

Im Rahmen der kollaborativen Bedarfsplanung werden die zuvor genannten Ansätze durch web-basierte Lösungen realisiert. Durch eine unternehmensübergreifende Planung werden die Akteure in früher Phase miteinbezogen. Alle Markt-, Kunden- und Produktinformationen fließen in die Bedarfsprognose ein, um deren Genauigkeit zu erhöhen (vgl. [Hellingrath et al. 2004](#), S. 111). Hierbei wird eine Echtzeitverarbeitung von Kunden-, Lieferanten- und eigenen Unternehmensdaten angestrebt, um auf Basis einer gemeinsamen Datengrundlage eine Bedarfsprognose zu erstellen (vgl. [Werner 2017](#), S. 92).

Netzwerkplanung Die Netzwerkplanung ist für die unternehmensübergreifende Koordination der SC-Akteure zuständig, wobei sich der Planungshorizont zwischen einigen Monaten und mehreren Jahren erstreckt (vgl. [Hellingrath et al. 2004](#), S. 106–107). Auf Basis der zuvor ermittelten Bedarfsgrundlage werden die Material- und Kapazitätsressourcen aufeinander abgestimmt, indem die Zuordnung von Produktmengen zu einem Standort erfolgt (vgl. [Schulte 2017](#), S. 794–795). Ziel ist ein mittelfristig optimaler Abgleich zwischen Prognosen und Aufträgen sowie Kapazitäten und Beständen, um ein Gesamtoptimum für alle Partner der SC zu bestimmen (vgl. [Busch et al. 2003](#), S. 40). Meist ist der dominierende Akteur, das fokale Unternehmen, mit dem höchsten Wertschöpfungsanteil und der größten Nähe zum Endkunden für die Netzwerkplanung verantwortlich, da er die meisten Informationen und Einflussmöglichkeiten zur Planung des Netzwerkes besitzt. Ergebnis der Netzwerkplanung ist die Zuordnung von Produktionsvolumina zu Werken (vgl. [Schulte 2017](#), S. 794–795; vgl. [Hellingrath et al. 2004](#), S. 106–107). Folgende Fragestellungen können im Rahmen der Netzwerkplanung beantwortet werden (vgl. [Stich et al. 2013c](#), S. 42; vgl. [Günther 2005](#), S. 20):

- Wie sollen Produktionsmengen zwischen den Werken unter Berücksichtigung ihrer Produktionskapazitäten und Produktionsstückkosten aufgeteilt werden?
- Welche Produktmengen eines Werkes sollen an das einzelne Distributionszentrum geliefert werden?

Zur Beantwortung der Fragestellungen modelliert der Anwender zuerst die betrachteten Produktions- und Distributionsstandorte sowie Kunden und -lieferanten (Knoten) und verbindet sie miteinander (Kanten) (vgl. [Sucky und Hönscheidt 2008](#), S. 133–134). In einem zweiten Schritt wird das Modell durch ergänzende Informationen an Knoten und Kanten angereichert. So können Transportrouten durch Zeit- oder Kostenrestriktionen und Werke durch Kapazitätsrestriktionen ergänzt werden (vgl. [Albrecht et al. 2015](#), S. 164–168; vgl. [Günther 2005](#), S. 21–22). Anschließend werden die zu produzierenden Produkte und ihre Kapazitäten den Standorten der SC mittels Netzwerkplanungsmodellen zugeordnet. Hierbei werden die Bedarfe den potenziellen Standorten gegenübergestellt und unter Berücksichtigung regionaler oder kapazitiver Restriktionen auf die Standorte aufgeteilt, so dass die Produktionskosten insgesamt minimiert werden (vgl. [Chopra und Meindl 2014](#), S. 159–163). Die Ergebnisse der Netzwerkplanung bilden die Grundlage für Beschaffungs-, Produktions- und Bedarfsplanung (vgl. [Busch et al. 2003](#), S. 40).

Beschaffungs-, Produktions- und Distributionsplanung Die Beschaffungsplanung sichert die Teileversorgung unter Minimierung der Bestände (vgl. [Schulte 2017](#), S. 795). Sie basiert auf den Ergebnissen der Bedarfs- und Netzwerkplanung. Der Planungshorizont beträgt in der Regel Tage oder Wochen. Im Rahmen der Beschaffungsplanung werden Lieferanten, Anlieferhythmen, Wiederbeschaffungszeit und Bestände definiert (vgl. [Werner 2017](#), S. 89). Die Beschaffungsplanung beschäftigt sich u. a. mit folgenden Fragestellungen (vgl. [Corsten und Gössinger 2008](#), S. 167; vgl. [Stich et al. 2013a](#), S. 81):

- Welche Lieferart sollte unter Berücksichtigung von Transportkosten gewählt werden?
- Wie lautet die optimale Beschaffungsmenge?

Für die Beschaffungsplanung eingesetzte Software enthält Simulationsszenarien zur Auswahl einer optimalen Versorgungs- und Dispositionsstrategien (vgl. [Hellingrath et al. 2004](#), S. 107). In der Bestandsplanung werden Verfahren zur Bestimmung optimaler Bestellrhythmen oder bestandsorientierte Verfahren, bspw. die Auslösung einer Bestellung bei Unterschreitung eines Sicherheitsbestandes, eingesetzt (vgl. [Stich et al. 2013a](#), S. 97–100). Zur kollaborativen Bestandsplanung werden unternehmensübergreifende Planungssoftwares eingesetzt, so dass die volle Verantwortung und der Handlungsspielraum zur Nachschubmenge und Bestandsführung beim Lieferanten liegt. Für dieses Konzept wird die Übermittlung von Echtzeitdaten des Bestandes vorausgesetzt (vgl. [Schuh et al. 2013b](#), S. 252).

Aufgabe der Produktionsplanung ist die Erstellung eines optimierten Produktionsplans für alle Werke auf Basis der werksübergreifenden Netzwerkplanung. Der Planungshorizont erstreckt sich von einigen Tagen bis hin zu Wochen (vgl. [Werner 2017](#), S. 90). Ziel ist eine optimale Kapazitätsauslastung unter Minimierung der Bestandskosten und Maximierung der Lieferbereitschaft sowie Termintreue (vgl. [Schulte 2017](#), S. 795). Als Hilfsmittel der Produktionsplanung werden Schicht- oder Maschinenbelegungspläne verwendet (vgl. [Werner 2017](#), S. 90). Die Beantwortung von Fragestellungen der Produktionsplanung findet in [Kapitel 4](#) keine Berücksichtigung, da die Produktionsplanung keinen zentralen Bereich des Logistikmanagements darstellt (vgl. [CSCMP 2013](#), S. 117).

Im Rahmen der Distributionsplanung erfolgt die Planung des Lagerbestands sowie die Transportplanung zur kostenminimalen Belieferung des Kunden. Ausgangsbasis bilden die Netzwerkplanung sowie die Ergebnisse aus der Bedarfsplanung (vgl. [Hellingrath et al. 2004](#), S. 106). Hierbei werden die notwendigen Transporte und Bestände unter Berücksichtigung von Transport- und Lagerrestriktionen ermittelt (vgl. [Busch et al. 2003](#), S. 41; vgl. [Schulte](#)

2017, S. 795). Im Rahmen der Distributionsplanung werden bspw. folgende Fragestellungen beantwortet (vgl. [Chopra und Meindl 2014](#), S. 394, 497):

- Wie viel Sicherheitsbestand ist für den gewünschten Grad der Produktverfügbarkeit erforderlich?
- Sollte der Transport direkt oder über ein Zwischenlager erfolgen?

In der Distributionsplanung werden zur Optimierung unterschiedliche Versorgungsszenarien durchgespielt (vgl. [Werner 2017](#), S. 90). Auf Basis der Ist-Struktur und ihrer zusammenhängenden Kosten können Konfigurationen unter variierenden Bedarfsprognosen oder Restriktionen getestet werden. Die Bestimmung von Transportflüssen erfolgt durch Optimierungen unter Berücksichtigung bestimmter Zielkriterien. Als Grundlage werden Mengendaten, bspw. Sendungsdaten oder geografische Daten, benötigt (vgl. [Fleischmann und Kopfer 2018](#), S. 81).

Order Promising Unter Order Promising wird die Verfügbarkeits- oder Machbarkeitsprüfung von Kundenanfragen oder -aufträgen verstanden (vgl. [Werner 2017](#), S. 90). Hierbei wird geprüft, ob Liefertermin, Liefermenge und geforderte Produktkonfiguration des Kunden erfüllt werden können. Bei nicht realisierbaren Kundenanforderungen werden Alternativen zur Auftrags Erfüllung generiert (vgl. [Schulte 2017](#), S. 795). Der Planungshorizont beträgt Stunden bis Tage (vgl. [Werner 2017](#), S. 90). So kann z. B. geprüft werden, ob eine Leistung zu definierten Konditionen erbracht werden kann, die benötigten Kapazitäten und Materialien vorhanden sind oder die gewünschte Produktkonfiguration durchführbar ist (vgl. [Schulte 2017](#), S. 795; vgl. [Werner 2017](#), S. 90). Typische Fragestellungen des Order Promising sind (vgl. [Hellingrath et al. 2004](#), S. 109):

- Was ist der schnellstmögliche Liefertermin für das Produkt?
- Ist die Einbringung eines individuellen Kundenwunsches möglich?

Zur Beantwortung der Kundenanfragen werden die Auftragsmengen durch Datenbankabfragen mit den vorhandenen Kapazitäten abgeglichen und mögliche Liefertermine berechnet. Mittels einer Locking-Funktionalität werden Produkte für bestimmte Kunden in unterschiedlichen Produktions- bzw. Distributionsphasen reserviert (vgl. [Hellingrath et al. 2004](#), S. 109). Zur Prüfung der verfügbaren Kapazitäten muss der Zugriff auf aktuelle Produktions- und Bestandsdaten gewährleistet sein (vgl. [Kilger 2015](#), S. 438–439).

In [Kapitel 2](#) wurden die Grundlagen des SCM besprochen. Hierbei wurden verwandte logistische Begriffe des SCM erläutert und abgegrenzt. Anschließend wurden ausgewählte Modelle mit dem Schwerpunkt auf SCM-Aufgaben erläutert. Abschließend wurden logistische Aufgaben und ihre Lösungsansätze dargestellt. Die Aufgaben aus [Abschnitt 2.2.1](#) und [Abschnitt 2.2.2](#) stellen den Untersuchungsgegenstand von [Abschnitt 4.1.1](#) dar.

3 Graphdatenbanksysteme und graphbasierte Verfahren

In [Kapitel 3](#) werden Grundlagen der Datenhaltung mit dem Schwerpunkt auf GDBS sowie graphbasierte Verfahren erläutert. So sollen wissenstheoretische Grundlagen geschaffen werden, um in [Kapitel 4](#) graphbasierte Kategorien herzuleiten. Hierbei bezeichnet das Adjektiv „graphbasiert“ alle Ansätze und Verfahren, die auf Graphen angewendet werden (vgl. [Meier und Kaufmann 2016](#), S. 91). Zuerst werden in [Abschnitt 3.1](#) Daten und Informationssysteme im SCM erläutert. So wird ein Verständnis über Zusammenhänge von Daten und Wissen sowie die Rolle von Informationssystemen im SCM geschaffen. Anschließend werden in [Abschnitt 3.2](#) Grundlagen der Datenhaltung anhand RDBS erläutert. So werden einerseits der grundlegende Aufbau von Datenbanksystemen (DBS) erklärt, andererseits die Vorteile von GDBS in [Abschnitt 3.3](#) verdeutlicht. Anschließend werden graphbasierte Modelle in [Abschnitt 3.4](#) im SCM und Graphalgorithmen vorgestellt. Sie bilden die Grundlage der zu untersuchenden Kategorien in [Kapitel 4](#).

3.1 Daten und Informationssysteme

Daten Daten basieren auf Zeichen mit einer hinzugefügten Syntax. Durch die Zuordnung einer Bedeutung und die Einbettung der Daten in einen Kontext werden sie zu Informationen (vgl. [Engelmann und Großmann 2015](#), S. 6–8). Informationen ändern die Wahrnehmung des Empfängers in Bezug auf einen Sachverhalt und wirken sich auf seine Beurteilung aus (vgl. [Bodendorf 2006](#), S. 2).

[Bodendorf](#) (vgl. [2006](#), S. 2) erläutert, dass die Definitionen des Wissensbegriffs in der Literatur variieren. Er hebt jedoch hervor, dass nach verbreiteter Auffassung Wissen durch die Verknüpfung von Informationen entsteht. Hierfür müssen sinnvolle Verknüpfungen von Informationen erkannt werden. Zur Beantwortung bestimmter Fragestellungen benötigt der Anwender zum einen Informationen über einen bestimmten Zustand, zum anderen Wissen über Zusammenhänge und Ursache-Wirkungsbeziehungen, d. h. wie sich dieser Zustand ergibt und ändert. Damit erfolgt die Vernetzung von Informationen zweckgerichtet hinsichtlich der jeweiligen Fragestellung (Pragmatik). [Bodendorf](#) fasst zusammen, dass Wissen im Gegensatz zu Daten meist unstrukturiert, vernetzt sowie kontextabhängig vorliegt und sich unmittelbar auf das Verhalten des Anwenders auswirkt. Daten bilden jedoch die Grundlage von Informationen und Wissen, so dass sie als größter Einflussfaktor zur Steuerung von SC gesehen werden (vgl. [Chopra und Meindl 2014](#), S. 75–76).

Daten lassen sich anhand von unterschiedlichen Kriterien, bspw. der Zeichenart, der Erscheinungsform, der Formatierung, der Stellung im Verarbeitungsprozess oder dem Verwendungszweck klassifizieren (vgl. [Schulte 2017](#), S. 105). Nach [Schemm](#) (vgl. [2009](#), S. 20) lassen sich Daten im SCM anhand ihres Verwendungszwecks, ihrer Änderungshäufigkeit und ihres Volumens unterscheiden. Hieraus ergeben sich vier grundlegende Datenarten:

die Stamm-, Bestands-, Änderungs- und die nachfolgend beschriebenen Bewegungsdaten (vgl. [Schemm 2009](#), S. 19–20).

Bewegungsdaten bilden den größten Teil des Datenbestandes in einer SC, da sie im Rahmen von Geschäftstätigkeiten immer wieder neu entstehen und in ihrem Volumen stetig wachsen (vgl. [Schemm 2009](#), S. 20). Da Bewegungsdaten in der SC von essenzieller Bedeutung sind, bilden sie häufig die Grundlage zur Wissensentdeckung (vgl. [Anane et al. 4-5 Nov. 2002](#), S. 1956–1957; vgl. [Scheidler 2017](#), S. 17). Sie ergeben sich in der SC aus einer Transaktion, den Wechsel eines materiellen oder immateriellen Gutes von dem Wirkungskreis eines Akteurs in den eines anderen (vgl. [Corsten und Gössinger 2008](#), S. 2–3). Nach [Schulte](#) (vgl. [2017](#), S. 104) stellen (Bewegungs-)Daten die Grundlage für Planungs-, Steuerungs- und Kontrollprozesse im SCM dar. (Bewegungs-)Daten werden in Unternehmen als wertsteigernde Ressource betrachtet, so dass logische und physische Konzepte zur Datenorganisation, -integration und -analyse erforderlich sind. Die Konzepte und damit einhergehenden Prozesse werden laut Schulte im Rahmen des Datenmanagements organisiert. Das Datenmanagement stellt ein Teilgebiet des Informationsmanagements dar. Dieses sichert die Informationsbereitstellung im Unternehmen in inhaltlicher, zeitlicher und räumlicher Hinsicht (vgl. [Dippold et al. 2005](#), S. 257).

Informationssysteme Nach [Schönsleben](#) (vgl. [2016](#), S. 393) werden Informationssysteme zur Speicherung und Auswertung großer Datenmengen sowie zur Bereitstellung von Informationen eingesetzt. Sie werden zur Beantwortung logistischer Fragestellungen genutzt, wenn die Komplexität der Fragestellung die menschlichen Fähigkeiten übersteigt. Dies lässt sich u. a. durch die steigende Komplexität von Produkten, größere Datenmengen und Häufigkeit der Aufträge sowie höhere Anforderungen an die Schnelligkeit der Prozessadministration begründen. Ein Informationssystem wird wie folgt definiert:

Definition 3.1 Informationssystem: „Ein IT-gestütztes logistisches Informationssystem stellt die Gesamtheit der Hard- und Software, Daten, Netzwerke und Personen dar, die der Unterstützung aller logistischen Planungs-, Abwicklungs-, Kontroll- und Steuerungsaufgaben dienen“ ([Schulte 2017](#), S. 86).

Jedes Informationssystem setzt sich aus einer Speicherungs- sowie einer Softwarekomponente zusammen (vgl. [Meier 2018](#), S. 3). Offene Informationssysteme sind mit dem Internet verbunden, so dass webbasierte Rechercharbeiten und der Informationsaustausch weltweit möglich sind (vgl. [Meier und Kaufmann 2016](#), S. 2). Die Speicherungskomponente, auch Wissensbank genannt, besteht aus einer Daten- und Methodenbank. Sie umfasst Daten und Verfahren (Methoden, Algorithmen), die auf die Datensammlungen der Speicherungskomponente angewendet werden können. Die Softwarekomponente bildet die Schnittstelle zum Anwender, der die Daten mittels Abfrage- und Manipulationssprache auswerten und verändern kann. Darüber hinaus regelt die Softwarekomponente die Zugriffs- und Bearbeitungsrechte und sorgt für Datenschutz sowie -sicherheit (vgl. [Meier 2018](#), S. 2–3).

Informationssysteme sind im SCM grundlegende Voraussetzung, um den Informationsfluss zwischen den Prozessbeteiligten zu gestalten. Sie verknüpfen alle Unternehmen und ermöglichen eine Gestaltung und Planung der SC (vgl. [Stich et al. 2013b](#), S. 278). Zur Unterstützung unternehmensübergreifender Aufgabenbereiche werden SCM-Informationssysteme eingesetzt. Hierbei teilen sich die Softwaremodule in einzelne Aufgabenbereiche auf, die meist der Einteilung des Aufgabenmodells aus [Abbildung 2.2](#) ähneln (vgl. [Schulte 2017](#), S. 791–792).

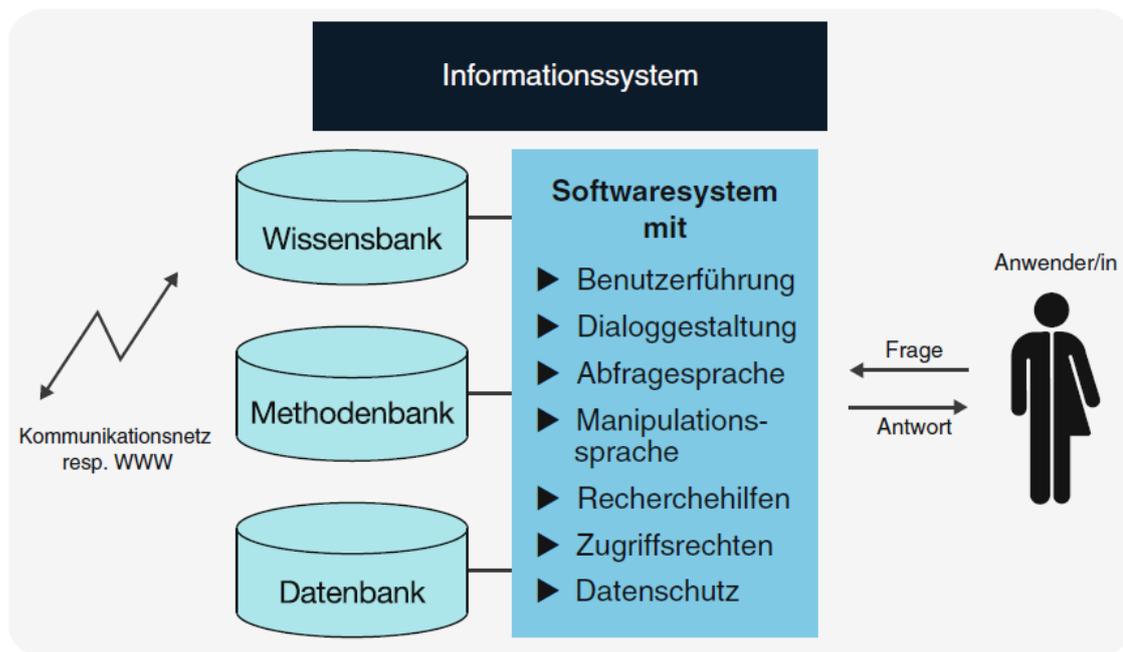


Abbildung 3.1: Architektur und Komponenten eines Informationssystems (Meier und Kaufmann 2016, S. 3)

Nach Stich et al. (vgl. 2013b, S. 278–280) erweitern SCM-Informationssysteme klassische ERP-Systeme, die auf eine unternehmensinterne Koordination der Ressourcen und Informationen abzielen. Die Module des SCM-Informationssystems fokussieren die überbetriebliche Koordination der SC unter Berücksichtigung von Restriktionen wie z. B. beschränkter Ressourcen oder Kapazitäten. Stich et al. erläutern, dass die bestehenden ERP-Systeme hierbei als Datengrundlage dienen, so dass alle Akteure auf eine gemeinsame Datengrundlage zurückgreifen können. Nach Stich et al. ergeben sich die Vorteile durch die Möglichkeit einer Zusammenarbeit an Planungsprojekten, der Wiederverwendbarkeit bestehender Projekte sowie der durchgängigen Dokumentation der Logistikprozesse und -strukturen. Auch wird die Planung durch einheitliche Methoden und Algorithmen standardisiert.

Schulte (vgl. 2017, S. 801) beschreibt die zentralen Grenzen der Implementierung gegenüber den Vorteilen von SCM-Informationssystemen. Oft existieren Zielkonflikte der beteiligten Unternehmen, bspw. werden sensible Informationen zu Lasten eines funktionierenden Informationsflusses nur zögerlich geteilt. Auch führen fehlende Kenntnisse über Planungspotenziale zu fehlender Akzeptanz der notwendigen IT-Unterstützung aufgrund angezweifelter Kosten-Nutzen-Verhältnisse. Unterschiedliche Systemleistungen der beteiligten Unternehmen und ein häufiger Wechsel der involvierten Akteure führen zu einer begrenzten Kompatibilität der Systeme.

Informationssysteme nutzen DBS zur Verwaltung und Speicherung von Daten, so werden in der Praxis häufig relationale Datenbanken eingesetzt (vgl. Meier und Kaufmann 2016, S. 2). Aufgrund wachsender heterogener Datenbestände, die in Echtzeit ausgewertet werden müssen, werden zunehmend nicht-relationale DBS zur Datenhaltung erforscht (vgl. Robak et al. 2013, S. 1216). Zum grundlegenden Verständnis der DBS werden zunächst RDBS vorgestellt, bevor nicht-relationale DBS eingeführt werden.

3.2 Relationale Datenbanksysteme

Ein DBS wird als eine Software definiert, die zur applikationsunabhängigen Beschreibung, Speicherung und Abfrage von Daten dient. Jedes DBS setzt sich aus einer Speicherungs- und Verwaltungskomponente zusammen. Die Datenbank (Speicherungskomponente) umfasst eine Sammlung mehrerer Dateien, die durch eine logische Abhängigkeit miteinander verbunden sind (vgl. [Meier und Kaufmann 2016](#), S. 2). Eine Datei besteht aus einer Zusammenfassung gleichartiger Datensätze. Hierbei umfasst ein Datensatz mehrere inhaltlich zusammenhängende Daten. Ein Datensatz besteht z. B. aus Materialnummer, -name, -gruppe und -preis (vgl. [Schulte 2017](#), S. 106). Das Datenbank-Management-System (DBMS) bildet die Verwaltungskomponente des DBS und ermöglicht die einheitliche Beschreibung und sichere Bearbeitung einer Datenbank. Ein DBMS garantiert u. a. die Korrektheit der Daten durch Einhaltung von Konsistenzregeln und Datensicherheit sowie den Schutz der Daten vor unberechtigten Zugriffen und Manipulationen (vgl. [Bodendorf 2006](#), S. 7). Darüber hinaus stellt das DBMS eine Datendefinitions- oder Beschreibungssprache zur Festlegung der Datenstruktur, eine Datenmanipulationssprache zum Datenbankzugriff und eine Speicherbeschreibungssprache zur physischen Datenorganisation zur Verfügung (vgl. [Schulte 2017](#), S. 107–108).

In Unternehmen werden meist relationale DBS verwendet, um Daten zentral, strukturiert und persistent zu speichern. Ein relationales DBS setzt sich analog zu DBS aus einer relationalen Datenbank und einem relationalen DBMS zusammen (vgl. [Meier und Kaufmann 2016](#), S. 10–11). Die relationale Datenbank umfasst das verwendete relationale Datenbank-Schema zusammen mit den gespeicherten Datensätzen (vgl. [Studer 2019](#), S. 14). Das relationale DBMS enthält die Datendefinitions-, Datenselektions- und Datenmanipulationssprache SQL. Darüber hinaus enthält die Sprache Spezialfunktionen zur Wiederherstellung von Datenbeständen, zum Datenschutz und zur Datensicherheit (vgl. [Meier und Kaufmann 2016](#), S. 9). Nachfolgend wird die Konzeption einer relationalen DBS beschrieben.

Zuerst werden auf abstrakter Ebene Datenstrukturen ohne Bezug zum konkreten DBS meist durch das Entity-Relationship-Modell (ERM) von Chen (1976) modelliert. Für die semantische Datenmodellierung hat sich die Modellierungssprache des ERM als Quasi-Standard etabliert (vgl. [Kudraß und Brinkhoff 2015](#), S. 49). Das ERM beschreibt Objekte, bspw. Gegenstände, Individuen, abstrakte Begriffe oder Ereignisse, durch Entitäten, die durch Beziehungen verbunden und mithilfe von Attributen beschrieben werden. Zusammen bilden Entitäts- sowie Beziehungstypen und Attribute die drei grundlegenden Elemente des ERM (vgl. [Meier und Kaufmann 2016](#), S. 28–33). Durch Kardinalitäten werden unterschiedliche Beziehungskonstellationen zwischen den Entitäten modelliert ([Schulte 2017](#), S. 11).

Im zweiten Schritt wird das ER-Diagramm in das formale Schema eines Datenbankmodells überführt. Bei relationalen Datenbanken erfolgt die Umwandlung durch das Relationenmodell (vgl. [Schulte 2017](#), S. 110). Das Modell basiert auf dem mathematischen Konzept der Relationen und ermöglicht eine formale Beschreibung der Struktur einer relationalen Datenbank (vgl. [Studer 2019](#), S. 11). Gleichartige Datensätze werden in einer Tabelle (Relation) innerhalb einer Datenbank gespeichert (vgl. [Meier 2018](#), S. 15–16). Zur Überführung des ER-Diagramms werden Entitäts- und Beziehungsmengen in Tabellen umgewandelt. Hierbei müssen Normalformen berücksichtigt werden, um Anomalien und Redundanzen zu vermeiden (vgl. [Meier und Kaufmann 2016](#), S. 26; vgl. [Unterstein und Matthiessen 2012](#), S. 245).

Zur Umsetzung des Relationenmodells in eine Datenbank wird als Quasi-Standard die Datenbanksprache SQL durchgesetzt. Durch SQL lassen sich Relationenschemata mittels Schemadefinitionssprache definieren, interne Dateiorganisationsformen sowie Zugriffspfade erzeugen, Anfragen an die Datenbank formulieren und Datenmanipulationen durchführen (vgl. [Bodendorf 2006](#), S. 32; vgl. [Meier und Kaufmann 2016](#), S. 107–113). Die Datenmanipulation erfolgt durch Operationen, die auf einer Menge von Relationen wirken und als Ergebnis stets eine Relation ausgeben. Operationen sind u. a. Selektionen (Zeilenauswahl), Projektionen (Spaltenauswahl) oder Verbunde bzw. Joins (Tabellenzusammenführung) (vgl. [Kudraß und Brinkhoff 2015](#), S. 92–94). Ein grundlegender Abfrageblock selektiert das gewünschte Merkmal (Output), definiert die Tabellen des Suchraums (Input) und die Selektionsprädikate (Processing) (vgl. [Meier und Kaufmann 2016](#), S. 105). Darüber hinaus existieren Transaktionsverwaltungen, damit mehrere Anwender gleichzeitig Zugriff auf die Datenbank haben (vgl. [Meier und Kaufmann 2016](#), S. 135). Eine Transaktion ist hierbei eine Folge von Datenbankanweisungen in Form von Operationen, die zusammen eine Einheit bilden (vgl. [Studer 2019](#), S. 145). In einem relationalen DBS sind Transaktionen an Integritätsregeln gebunden und müssen atomar, konsistent, isoliert und dauerhaft sein (ACID-Prinzip). Durch das ACID-Prinzip wird gewährleistet, dass eine Änderung in der Datenbank nach außen erst sichtbar wird, wenn die Integritätsbedingungen eingehalten wurden (vgl. [Meier und Kaufmann 2016](#), S. 135–136).

Durch die zunehmende Digitalisierung von SC nehmen die zu verarbeitenden Datenmengen und auch die geforderte Verarbeitungsgeschwindigkeit zu (vgl. [Schulte 2017](#), S. 115). Die zuvor beschriebenen Eigenschaften relationaler DBS können diesen Anforderungen nur begrenzt begegnen. Aufgrund der starren Tabellenstruktur werden heterogene Datenmengen meist nur unter Informationsverlusten umgewandelt (vgl. [Robak et al. 2013](#), S. 1216). Die Überprüfung der Datenintegrität und der einzuhaltende Transaktionsschutz sind mit viel Aufwand und Rechenleistung verbunden. Große Datenmengen (teilweise im Petabyte-Bereich und größer) können deshalb nur langsam und ineffizient verarbeitet werden (vgl. [Meier und Kaufmann 2016](#), S. 221–222). Auch die Verfügbarkeit und Skalierbarkeit relationaler Datenbanken werden aufgrund der Konsistenzregeln eingeschränkt (vgl. [Pokorny 2013](#), S. 72). Insbesondere komplexe Beziehungsgeflechte können in relationalen Datenbanken nur unter hohem Aufwand abgebildet und untersucht werden. So können in relationalen Datenbanken komplexe Datenstrukturen wie z. B. Bäume oder Graphen nur bedingt traversiert werden (vgl. [Aggarwal und Wang 2010c](#), S. 16). So erfolgt die Darstellung transitiver Relationen über Vereinigungen von Tabellen, so dass zunehmende Vernetzungen zu wachsenden Tabellen führen. Das Resultat sind große Verknüpfungstabellen mit dünn besetzten Zeilen (vgl. [Robinson et al. 2015](#), S. 11–13). Analytische Fragestellungen führen aufgrund der normalisierten Datenspeicherung zu komplexen und aufwendigen Operationen (vgl. [Fasel 2016](#), S. 110). Als Ergänzung relationaler DBS wurden deshalb in den letzten Jahren zunehmend NoSQL-Technologien („not only SQL“) erforscht. Insbesondere mit dem Aufkommen des Internets und der Vielzahl an webbasierten Anwendungen steigt die Bedeutung von NoSQL-Technologien (vgl. [Meier und Kaufmann 2016](#), S. 18). Im folgenden Abschnitt werden die NoSQL-DBS mit dem Fokus auf GDBS eingeführt und erläutert.

3.3 Graphdatenbanksysteme

Durch die Entwicklung des Internets und die zunehmende Popularität von Web-Diensten wurden neue DBS benötigt, die große Datenmengen effizient verarbeiten konnten. So wurde

die Entwicklung massiv verteilter Datenbanken, die nicht auf relationalen Datenmodellen sowie SQL basieren, vorangetrieben (vgl. [Meier und Kaufmann 2016](#), S. 221–222). Die neu entwickelten Datenbanktechnologien werden unter NoSQL-Datenbanken („Not only SQL“) zusammengefasst. Der Begriff schließt SQL-basierte Datenbanken nicht aus, sondern betrachtet diese als eine Ausprägung von NoSQL-Datenbanken (vgl. [Fasel 2016](#), S. 111).

Die Motivationen zur Entwicklung von NoSQL-DBS liegen in der leichteren Datenbereitstellung, der Verarbeitung großer Datenmengen und der horizontalen sowie vertikalen Skalierbarkeit in Form von verteilten Ressourcen oder erhöhter Leistung einzelner Ressourcen (vgl. [Ganesh Chandra 2015](#), S. 13). Im Gegensatz zu relationalen Datenbanken werden in NoSQL-Datenbanken Verfügbarkeit und Ausfalltoleranz fokussiert. NoSQL-Datenbanken basieren auf dem BASE-Konsistenzmodell (Basically Available, Soft State, Eventually Consistent) (vgl. [Meier und Kaufmann 2016](#), S. 148). Das Modell priorisiert Verfügbarkeit über Konsistenz und erlaubt temporär inkonsistente Kopien von Datenobjekten (vgl. [Ganesh Chandra 2015](#), S. 16). NoSQL-Datenbanken lassen sich in verschiedene Datenbankfamilien und werden wie folgt differenziert ([Meier und Kaufmann 2016](#), S. 222):

- Schlüssel-Wert-Datenbanken
- Spaltenfamilien-Datenbanken
- Dokument-Datenbanken
- Graphdatenbanken

Graphdatenbanken unterscheiden sich von den gelisteten Datenbankfamilien. Die Datenmodelle von Schlüssel-Wert-, Spaltenfamilien- und Dokument-Datenbanken verzichten auf Datenbankschemata, um eine einfache Fragmentierung der Datenbanken zu gewährleisten. Graphdatenbanken basieren auf Grundsätzen der Graphentheorie und besitzen ein darauf aufbauendes Datenbankschema (vgl. [Meier und Kaufmann 2016](#), S. 237). So werden die Daten in einer Graphenrepräsentation gespeichert, so dass neue Möglichkeiten zur Datenabfragen und -manipulation entstehen, die auf einen Graphen angewendet werden können (vgl. [Angles und Gutierrez 2018](#), S. 3). Dies ermöglicht die Beantwortung logistischer Fragestellungen auf Basis von Graphen. So lassen sich u. a. hierarchische Strukturen, Transportprobleme oder kostenoptimale Rundreisen durch Graphen einfacher als durch Tabellen beschreiben (vgl. [Struckmann und Wätjen 2016](#), S. 157). Im nächsten Abschnitt werden graphentheoretische Grundlagen erläutert, um ein Verständnis der unterschiedlichen Begrifflichkeiten zu Graphen zu schaffen.

3.3.1 Graphentheoretische Grundlagen und Begriffsbestimmung

Zur Schaffung der wissenstheoretischen Grundlagen werden im folgenden Abschnitt grundlegende Begriffe und Algorithmen der Graphentheorie beschrieben. Anschließend erfolgt die Erläuterung der einzelnen Bestandteile eines GDBS. Einleitend werden unterschiedliche Ausprägungen von Graphen beschrieben. Graphen stellen die Grundlage der Graphentheorie dar (vgl. [Aigner 2015](#), S. 4–9). Sie werden wie folgt definiert:

Definition 3.2 Graph: Ein *Graph* G ist ein Paar von Mengen

$$G = (V, E). \quad (3.1)$$

Dabei ist V eine Menge mit beliebig vielen Elementen, den sogenannten *Knoten* von G . Mit E wird die Menge aller *Kanten* bezeichnet. Eine Kante verbindet zwei (im Allgemeinen unterschiedliche) Knoten miteinander (Krischke und Röpcke 2015, S. 17).

Die aufgeführte Definition wird durch Ausführungen von Aigner (vgl. 2015, S. 4–5) ergänzt. Sind Start- und Endknoten einer Kante identisch, wird die Kante als Schlinge bezeichnet. Kanten, die über mehrere Knoten hinweg denselben Start- und Endknoten haben, werden Kreise bzw. Zyklen genannt. Verbinden zwei oder mehr Kanten das betrachtete Knotenpaar, werden diese als Mehrfachkanten bezeichnet. Ein Graph ohne Schlingen und Mehrfachkanten wird als einfacher Graph bezeichnet.

Neben den einfachen Graphen existieren weitere Ausprägungen von Graphen. So wird ein Graph, dessen Kanten von einem Knoten in die Richtung eines nächsten Knotens verlaufen, als gerichteter Graph bezeichnet. Die Richtungen der Kanten sind fest definiert und durch einen Pfeil an dem Linienende der Kante dargestellt (vgl. Struckmann und Wätjen 2016, S. 158).

Eine weitere Form von Graphen stellen Eigenschaftsgraphen dar. Eigenschaftsgraphen sind gerichtete Graphen, deren Knoten und Kanten durch Eigenschaften bzw. Gewichte charakterisiert werden (vgl. Angles und Gutierrez 2018, S. 11). Das Gewicht eines knoten- oder kantengewichteten Graphen wird durch die Summe sämtlicher Knoten- bzw. Kantengewichte berechnet (vgl. Struckmann und Wätjen 2016, S. 191)

Des Weiteren lassen sich Teilgraphen aus einem Graphen bilden. Hierbei wird ein Graph als Teilgraph bezeichnet, wenn seine Knoten und Kanten eine Teilmenge der Knoten und Kanten des übergeordneten Graphen bilden (vgl. Struckmann und Wätjen 2016, S. 160).

Bäume Ein zusammenhängender Graph ohne Zyklen mit einer minimalen Kantenanzahl wird als Baum bezeichnet (vgl. Aigner 2015, S. 9). Ein *Graph* G heißt zusammenhängend, falls je zwei verschiedene Knoten einen Weg in G besitzen (vgl. Krischke und Röpcke 2015, S. 22). Für einen Baum $N = (V, E)$ gelten folgende Eigenschaften (vgl. Mattfeld und Vahrenkamp 2014, S. 27):

- Ein Baum ist zusammenhängend und zyklusfrei.
- Je zwei Knoten sind in einem Baum über genau einen Weg verbunden
- Ein Baum besitzt bei n Knoten genau $n - 1$ Kanten
- Ein Baum besitzt bei n Knoten genau $n - 1$ Kanten

Ein Baum enthält wegen seines Zusammenhangs keine isolierten Knoten. Auch Zyklen können in einem Baum nicht enthalten sein, da in einem Zyklus stets eine Kante entfernt werden kann, ohne dass der Zusammenhang unterbrochen wird (vgl. Vahrenkamp 2003, S. 48). Existiert innerhalb eines ungerichteten Graphen G ein Teilgraph, der die Definition eines Baumes erfüllt und alle Knoten von G enthält, wird der Teilgraph aufspannender Baum genannt (vgl. Krumke und Noltemeier 2009, S. 99). Wird ein nicht zusammenhängender Graph aus Bäumen zusammengesetzt, wird der Graph als Wald bezeichnet (vgl. Aigner 2015, S. 115).

Graphrepräsentationen Die Gestaltung effizienter Datenstrukturen zur Speicherung und schnellen Suche nach Daten und Informationen ist eine zentrale Aufgabe der Informatik (vgl. Struckmann und Wätjen 2016, S. 157). Nach Cook und Holder (vgl. 2007, S. 1–2) ist das Data-Mining zum Auffinden von Wissen zu einer der größten Herausforderungen in

der Wissenschaft und der Industrie aufgestiegen. In vielen Bereichen liegt das Wissen laut Cook und Holder in den Beziehungen zwischen Entitäten. In relationalen Datenbanken werden Daten in Tabellenform gespeichert, so dass Beziehungen nur in begrenztem Maße untersucht werden können (vgl. [Robinson et al. 2015](#), S. 11; vgl. [Holder und Cook 2007](#), S. 1).

Eine Alternative zur relationalen Datenhaltung in Datenbanken bildet die Graphenrepräsentation [Cook und Holder](#) (vgl. [2007](#), S. 1–2). In einer Graphenrepräsentation werden Daten in Form von Graphen abgespeichert. So werden Entitäten und Beziehungen gemeinsam in Form von Knoten und Kanten dargestellt (vgl. [Shao und Li 2018](#), S. 145).

[Cook und Holder](#) (vgl. [2007](#), S. 1–2) heben hervor, dass die Graphenrepräsentation einerseits die notwendige Datenstruktur zur Anwendung von Graph-Mining-Algorithmen bietet, andererseits erleichtert sie das Verständnis des Anwenders durch eine direkte Visualisierung der Daten in Form von Graphen. Das Data-Mining, das Graph-Mining und Graph-Mining-Algorithmen werden in [Abschnitt 3.4.3](#) erläutert.

Die Graphenrepräsentation nutzt zur Datenspeicherung Adjazenzlisten oder Adjazenzmatrizen (vgl. [Shao und Li 2018](#), S. 145). Knoten, die durch eine Kante verbunden sind, werden adjazent bzw. benachbart genannt. Zu einem Knoten benachbarte Kanten werden als inzident bezeichnet (vgl. [Dörn 2017](#), S. 185). Adjazenzlisten speichern für jeden Knoten eine linear verkettete Liste aller inzidenter Kanten ab, d. h. für jeden Knoten wird eine Liste von Nachbarknoten oder angrenzenden Kanten bestimmt (gl. [Dörn 2017](#), S. 192–193). In der Adjazenzmatrix werden die Knoten in den Zeilen i und Spalten j in der Adjazenzmatrix aufgetragen. Werden zwei Knoten durch eine Kante verbunden, wird an der jeweiligen Stelle (i, j) eine 1, ansonsten eine 0 eingetragen (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 55).

Zusätzlich lässt sich auf Grundlage der Adjazenzmatrix eine Kostenmatrix aufstellen. In den Zeilen werden anstelle der binären Darstellung die Kosten bzw. Gewichte zwischen den betrachteten Knoten erfasst (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 56). Die Art und Weise, wie Daten in Form von Graphen in einer Graphenrepräsentation eingespeichert werden, wirkt sich auf die Verarbeitung des Graphen aus. So basieren die meisten Graphabfragen und Graph-Algorithmen auf Adjazenzlisten, da in Adjazenzmatrizen die gesamte Zeile nach den gesuchten benachbarten Kanten gescannt werden müsste (vgl. [Shao und Li 2018](#), S. 145).

Graphdatenbanksysteme Datenbanksysteme, die Daten in einer Graphenrepräsentation abspeichern, werden als GDBS bezeichnet (vgl. [Angles und Gutierrez 2018](#), S. 22–23; vgl. [Holder und Cook 2007](#), S. 1–2). GDBS wurden seit den 1980er Jahren erforscht, jedoch fokussierten Forschungsaktivitäten rund um DBS die Untersuchung von semistrukturierten Daten. Auch die technologischen Voraussetzungen zur Umsetzung von GDBS sind erst seit einigen Jahren gegeben, so dass die Forschung an GDBS intensiviert wird (vgl. [Angles und Gutierrez 2008](#), S. 9; vgl. [Angles und Gutierrez 2018](#), S. 22). Eine GDBS wird wie folgt definiert:

Definition 3.3 Graphdatenbanksystem: “A graph database system (GDBS)—or just graph database—is a system specifically designed for managing graph-like data following the basic principles of database systems, that is, persistent data storage, physical/logical data independence, data integrity and consistency“ ([Angles und Gutierrez 2018](#), S. 22).

GDBS eignen sich zur Speicherung stark vernetzter Daten und traversieren die gespeicherten Graphen effizienter als RDBS (vgl. [Pokorny 2015](#), S. 59). [Fasel](#) (vg. [2016](#), S. 122–224) beschreibt den Vorteil wie folgt: Durch die Speicherung von Graphen in einer relationalen Datenbank würden Datenbankabfragen schnell zu großen Tabellen führen. So würde ein soziales Netzwerk bestehend aus 100 Personen zu einer Beziehungstabelle von maximal 9900 Einträgen führen und exponentiell zu der Personenzahl wachsen. Fasel beschreibt, dass durch die Graphenrepräsentation Personen und ihre Beziehungen direkt miteinander verbunden werden. Abfragen fangen bei einem Knoten an und wandern durch den Graphen, so dass keine Relationentabelle zur Abfrage von Beziehungen notwendig ist. Graphdatenbanken werden zur Analyse komplexer Netzwerke, bspw. sozialer Netzwerke, des Internets, Proteininteraktionen oder Empfehlungssystemen eingesetzt (vgl. [Dominguez-Sal et al. 2011](#), S. 28–29)

[Meier und Kaufmann](#) (vgl. [2016](#), S. 239) beschreiben folgende Nachteile des GDBS. So ist die Fragmentierung (Sharding) der Graphdatenbank aufwändiger als bei anderen Arten von Datenbanken. Dies begründet sich durch die zentrale Bedeutung von Beziehungen zwischen den Datensätzen, die bei einer Aufteilung der Datensätze berücksichtigt werden müssen. Meier und Kaufmann empfehlen Cluster-Algorithmen zur Ermittlung stark vernetzter Teilgraphen als Partition (vgl. [Meier und Kaufmann 2016](#), S. 239). Eine Erläuterung von Cluster-Algorithmen erfolgt in [Abschnitt 3.4.3](#).

3.3.2 Graphmodellierung

Analog zu dem Relationenmodell dienen Graphenmodelle zur strukturellen Beschreibung der Daten in einer Graphdatenbank und geben ein bestimmtes Datenbank-Schema vor. Im Gegensatz zum Relationenmodell werden Daten und ihre Beziehungen untereinander nicht als Tabelle, sondern als Graph modelliert (vgl. [Angles und Gutierrez 2018](#), S. 3). Graphenmodelle werden formal durch die Graphentheorie gestützt (vgl. [Meier und Kaufmann 2016](#), S. 12). Grundlegende Graphenmodelle lassen sich wie folgt einteilen (vgl. [Angles und Gutierrez 2018](#), S. 7):

- Labeled Graphs (Beschriftete Graphen)
- Hypergraphs (Hypergraphen)
- Hypernodes (Hyperknoten)
- Resource Description Framework (RDF)
- Property Graphs (Eigenschaftsgraphen)

Beschriftete Graphen bilden die simpelste Form von Graphenmodellen und speichern Daten in beschriftete Knoten und Kanten ein (vgl. [Angles und Gutierrez 2018](#), S. 7). Hypergraphen stellen eine Generalisierung von Graphen dar. Die Kanten von Hypergraphen werden als Hyperkanten bezeichnet und verbinden anstatt einzelner Knoten Knotenmengen miteinander (vgl. [Pokorny 2015](#), S. 59). Hyperknoten sind gerichtete Graphen, deren Knoten wiederum Graphen darstellen können. Im Gegensatz zu den beschrifteten Graphen erlauben Hypergraphen und -knoten komplexe Datenstrukturen (vgl. [Angles und Gutierrez 2018](#), S. 7–10). Das RDF wurde ursprünglich zur Beschreibung von Metadaten entwickelt und ist heute grundlegend für das semantische Web, um Informationen des Internets zu semantischen Zusammenhängen bzw. Wissen zu verknüpfen (vgl. [Junghanns et al. 2017](#), S. 465). Das RDF bietet einen Mechanismus zur Beschreibung von Ressourcen. Hierbei besteht eine Beschreibung (Tripel) aus Subjekt (Ressource), Prädikat (Eigenschaft) und

Objekt (Eigenschaftswert). Jedes Tripel stellt eine Aussage zwischen Subjekt und Objekt, die durch das Prädikat miteinander verbunden werden, dar. Die Tripel werden in einer Graphenrepräsentation abgespeichert (vgl. [Ganzha 2013](#), S. 1219; vgl. [Angles und Gutierrez 2008](#), S. 33).

Wie in [Abschnitt 3.3.1](#) beschrieben, ordnet der Eigenschaftsgraph seinen Knoten und Kanten eine beliebige Anzahl an Eigenschaften bzw. Gewichten zu. Eigenschaftsgraphen werden häufig zur Konzeption von Datenbanken verwendet, da sie aussagekräftig sind und in der Praxis einfach umgesetzt werden können (vgl. [Angles und Gutierrez 2018](#), S. 11). In der Industrie zählen Eigenschaftsgraphen zu den populärsten Graphenmodellen und auch im akademischen Umfeld wächst die Relevanz von Eigenschaftsgraphen (vgl. [Francis et al. 2018](#), S. 1433).

[Meier und Kaufmann](#) (vgl. [2016](#), S. 74–80) beschreiben das Vorgehen zur Überführung des ER-Diagramms in den Eigenschaftsgraphen wie folgt: Zuerst werden die Entitätsmengen und ihre Attribute in Knoten mit zugehörigen Eigenschaften transformiert. Die Beziehungsmengen werden in Kanten mitsamt den zugeordneten Attributen umgewandelt und mit den entsprechenden Knoten verknüpft. Laut Meier und Kaufmann existieren analog zu den relationalen Datenbanken auch für Graphdatenbanken einzuhaltende Konsistenzbedingungen.

3.3.3 Graphbasierte Abfragesprache

Graphbasierte Datenbanksprachen manipulieren die in Form einer Graphenrepräsentation gespeicherten Daten durch Graph-Transformationen (vgl. [Meier und Kaufmann 2016](#), S. 111). Graph-Transformationen wie CREATE, DELETE und TRANSFORM sind Graphoperationen, die zur Manipulation von Graphdatenbanken dienen. Zusammen mit Operationen zur Abfrage von Kanten, Knoten oder Attributen bilden Graph-Transformationen atomare Graphoperationen. Diese Graphoperationen sind elementar zur Manipulation von Graphdatenbanken und bilden die Grundlage für jede komplexe Abfrage (vgl. [Dominguez-Sal et al. 2011](#), S. 31–34).

[Meier und Kaufmann](#) (vgl. [2016](#), S. 111) beschreiben Ähnlichkeiten von relationalen und graphbasierten Sprachen. Wie relationale Sprachen sind auch graphbasierte Sprachen mengenorientiert, da sie auf der Ebene von Graphen, einer Menge von Knoten und Kanten, arbeiten. Auch lassen graphbasierte Sprachen die Filterung von Daten anhand von Prädikaten (konjunktive Abfragen) zu, so dass Untermengen in Form von Teilgraphen zurückgegeben werden. Ebenfalls können Aggregationen in Form von Knotenanzahl, -summe oder -minimum durchgeführt werden.

Nach [Miller](#) (vgl. [2013](#), S. 144) unterscheiden sich relationale und graphbasierte Sprachen insbesondere in der Art, wie sie die Datenbank traversieren können. Durch die Graphenrepräsentation erfolgt die Abfrage oder Manipulation direkt auf einem Graphen. Pfad- oder Musterabfragen lassen sich im Gegensatz zu relationalen Abfragen unkompliziert realisieren, da sie ohne Joins auskommen und die Pfade direkt traversieren können. Insbesondere rekursive Abfragen, wie die Bestimmung von Pfadlängen und indirekten Beziehungen, sind in graphbasierten Abfragesprachen unkomplizierter als in relationalen Sprachen zu deklarieren (vgl. [Meier und Kaufmann 2016](#), S. 111–113).

Im Gegensatz zu relationalen Datenbanken, in der SQL als Quasi-Standard eingesetzt wird, existiert für Graphdatenbanken noch kein einheitlicher Standard zur Abfrage und

Manipulation von Daten (vgl. [Francis et al. 2018](#), S. 1434). Als populärste Graphdatenbank gilt Neo4j, die Cypher als Abfragesprache nutzt (vgl. [DB-Engines 2020](#), S. 1).

Durch die Abfragesprache Cypher wird auf die Datenbank zurückgegriffen, um Muster zu extrahieren und Daten zu manipulieren. Cypher ist eine deklarative Abfragesprache und orientiert sich an der Syntax von SQL (vgl. [Miller 2013](#), S. 144; vgl. [Meier 2016](#), S. 111). Mittlerweile wird Cypher in kommerziellen Datenbankprodukten, z. B. SAP HANA Graph, und in der Forschung eingesetzt (vgl. [Francis et al. 2018](#), S. 1433–1434). Auch lassen sich mathematische Formeln oder Algorithmen auf Basis der Programmiersprache Java implementieren (vgl. [Neo4j 2020c](#), S. 1). Ein grundlegender Abfrageblock in Cypher ist wie folgt aufgebaut ([Meier und Kaufmann 2016](#), S. 113):

- MATCH zur Beschreibung von Suchmustern
- WHERE für Bedingungen zur Filterung der Ereignisse
- RETURN zur Rückgabe von Eigenschaften, Knoten, Beziehungen oder Pfaden

Nach [Meier und Kaufmann](#) (vgl. [2016](#), S. 113–116) geschieht die Schemadefinition in Cypher implizit, so dass Befehle zur Beschreibung von Datenklassen, bspw. Knoten- und Kanten typen, nicht benötigt werden. Infolgedessen ist die Datendefinitionssprache in Cypher kompakter als in SQL. Jedoch existieren in Cypher im Gegensatz zu SQL keine direkten Sprachelemente für Sicherheitsmechanismen. Aufgrund der Popularität von Neo4j zur praktischen Anwendung und der ähnlichen Syntax von Cypher und SQL erfolgt die Umsetzung des Fallbeispiels in [Abschnitt 5.2](#) in Neo4j.

[Abschnitt 3.1](#) bis [Abschnitt 3.3](#) haben die wissens-theoretischen Grundlagen zur graphbasierten Datenhaltung aufgearbeitet. Den Untersuchungsgegenstand in [Kapitel 4](#) bilden jedoch graphbasierte Verfahren, um Lösungsansätze zur Beantwortung von logistischen Fragestellungen zu entdecken. Eine Beschreibung graphbasierter Modelle und Graphalgorithmen erfolgt im nächsten Abschnitt.

3.4 Graphbasierte Verfahren

Durch die Speicherung von Daten in eine Graphenrepräsentation ergeben sich eine Vielzahl an Lösungsansätzen zur Beantwortung logistischer Fragestellungen, da graphbasierte Verfahren eingesetzt werden können (vgl. [Nebel und Wild 2018](#), S. 293). In dieser Arbeit werden graphbasierte Verfahren als Vorgehensweisen verstanden, die auf Graphen angewendet werden. Zuerst werden in [Abschnitt 3.4.1](#) graphbasierte Modelle im SCM beschrieben. Anschließend werden in [Abschnitt 3.4.2](#) und [Abschnitt 3.4.3](#) Graph-Management- und Graph-Mining-Algorithmen vorgestellt.

3.4.1 Graphbasierte Modelle im Supply-Chain-Management

Nach [Töllner et al.](#) (vgl. [2010](#), S. 7–9) unterscheidet sich die Definition eines Modells abhängig der fachspezifischer Verwendung. Allgemein zeichnen sich Modelle durch die Abbildung, die Verkürzung und den Pragmatismus aus. Modelle sind die Repräsentation eines realen Systems, wobei nur die für den Modellierer relevanten Attribute des Originals erfasst werden. Der Pragmatismus drückt die zweckgebundene Verbindung zwischen Modell und Original aus. Töllner et al. beschreiben unterschiedliche Zielsetzungen von Model-

len, die abhängig von ihrem Einsatzzweck sind. So werden Modelle u. a. zur Beschreibung, Prognose, mathematischen Berechnung oder zur experimentellen Untersuchung eingesetzt.

Zur Beantwortung logistischer Fragestellungen existieren graphbasierte Modelle, die durch die Nutzung graphentheoretischer Grundlagen charakterisiert werden (vgl. [Vahrenkamp 2003](#), S. 3–5). Jedes Modell basiert auf Knoten und Kanten sowie optionalen Gewichten, die je nach Modell unterschiedliche Bedeutungen aufweisen. So können Knoten entweder Orte, Partner, Objekte oder Zustände ausdrücken. Kanten charakterisieren z. B. Verkehrswege, Güterflüsse oder Übergänge. Die Relationen können physisch, logisch, hierarchisch oder zeitlich sein. Ergänzt werden die Knoten und Kanten durch Gewichte in Form von Distanzen, Mengen oder Zeit (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 5). Eine Auswahl graphbasierter Modelle zur Lösung ausgewählter Fragestellungen wird nachfolgend gegeben.

Entscheidungsbäume (Szenariobäume) In Entscheidungsbäumen lassen sich sukzessive Entscheidungssituationen abbilden oder ein Gesamtproblem in Teilprobleme unterteilen. Hierbei werden jedem Knoten eine zeitlich auseinanderfallende Entscheidung bzw. ein Teilproblem zugeordnet (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 30). [Schade](#) (vgl. [2012](#), S. 48–49) verwendet Entscheidungsbäume in der stochastischen Optimierung zur Modellierung komplexer Szenarien und ihrer Wahrscheinlichkeiten. Schade bezeichnet die verwendeten Entscheidungsbäume als Szenariobäume, da mögliche Szenarien modelliert werden. Jedes Szenario stellt nach Schade ein Ereignis in Form eines Knotens dar und erfordert eine Entscheidung, bis das Endereignisse erreicht wird. Auch im Data-Mining werden Entscheidungsbäume zur autonomen Klassifizierung von Datensätzen genutzt (vgl. [Cleve und Lämmel 2014](#), S. 92).

Modelle der Standortbestimmung Die Standortbestimmung zur Ansiedlung von Produktionswerken und Distributionslagern ist von einer Vielzahl von Standortfaktoren abhängig. Standorte können weltweit, landesweit oder regional gesucht werden, wobei unterschiedliche Standortfaktoren beachtet werden müssen (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 99–101). Auf Basis quantitativer Standortfaktoren (z. B. Infrastruktur) und qualitativer Standortfaktoren (z. B. Verfügbarkeit von Fachkräften) werden zuerst potenzielle Standorte gewählt. Nach der Vorauswahl erfolgt die eigentliche Standortbestimmung, mit dem Ziel, den monetären Gesamtgewinn unter Berücksichtigung von Logistik- und Standortkosten zu maximieren (vgl. [Chopra und Meindl 2014](#), S. 159).

Im Fokus der Standortbestimmung steht die Suche nach optimalen Standorten für Produktionswerke, Zentrallager und Regionallager, wobei die Modelle unterschiedliche Zielsetzungen hinsichtlich optimaler Standorte verfolgen (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 102–103). Das Center-Problem bestimmt einen Standort (Zentrum), dessen maximale Entfernung zu den übrigen Knoten minimal ist. Die Erweiterung des Center-Problems wird als p-Center-Problem bezeichnet und wählt anstatt eines optimalen Standortes mehrere aus. Die Center-Probleme werden bei Dienstleistungsunternehmen angewendet (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 103–106). Im Rahmen des Covering-Problems wird eine minimale Anzahl an Standorten gewählt, um alle Kunden innerhalb eines gewissen Radius zu versorgen (vgl. [Drexel und Domschke 1996](#), S. 153–156). Das Covering-Problem wird u. a. für die Planung von Regionallagern angewandt (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 108). Das Median-Problem (oder p-Median-Problem) berechnet die optimale Ansiedlung eines Standortes (mehrerer Standorte) mit dem Ziel, die kleinste Summe der gewichteten Entfernungen des Lagerstandortes (der Lagerstandorte) zu Kunden zu be-

rechnen. Hierbei werden die gewichteten Entfernungen durch entstehende Transport- und Betriebskosten dargestellt (vgl. [Vahrenkamp 2003](#), S. 139–141). Das P-Median-Problem wird im Rahmen des Warehouse-Location-Problems z. B. zur Konsumgüter-Distribution eingesetzt (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 123).

Darüber hinaus lassen sich weitere Probleme und Heuristiken ableiten, die zur Herleitung möglicher Standorte unter Berücksichtigung spezifischer Kosten verwendet werden (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 125–127). Nach Chopra und Meindl (160, 2014) lassen sich insbesondere die Bestimmung der Anzahl an Lagern, Distributionszentren und Servicezentren sowie ihre optimalen Standorte durch Standortmodelle errechnen. Jedoch werden Modelle zur Standortbestimmung auch auf Produktionswerke oder zur Kapazitätsaufteilung angewendet, indem Produktionskosten und Kapazitäten in die Optimierung aufgenommen werden.

Transport- und Zuordnungsmodelle Nachdem die Standorte festgelegt wurden, erfolgt die Konfiguration der SC durch Transport- und Zuordnungsmodelle (**TuZ-Modelle**) (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 147). Transportmodelle modellieren und legen Produktflüsse, bspw. Liefergebiete, zwischen verschiedenen Anbietern und Kunden fest. Ziel ist die Minimierung der Gesamtkosten des Transportes durch Optimierung der transportierten Mengeneinheit und der betrachteten Fixkosten. Hierfür wird festgelegt, welcher Standort für welches Liefergebiet verantwortlich ist. Die Transportmodelle können auch mehrere Lagerstufen in Form von Zentral- oder Zwischenlager modellieren (vgl. [Vahrenkamp 2003](#), S. 90). Die Zuordnung erfolgt unter Berücksichtigung von Transportkosten, Angebot der Anbieter und Nachfrage der Kunden auf Basis einer Kostenmatrix. Durch sukzessive Sperrung der Anbieterzeilen und Kundenspalten nach aufsteigenden Kosten erfolgt so die Zuordnung der Anbieter zu den Kunden bis die gesamte Nachfrage gedeckt ist (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 151–152).

Durch das Zuordnungsproblem lassen sich kostenminimale Zuordnungen in einem bipartiten Netzwerk, d. h. einem Graph bestehend aus zwei disjunkten Teilmengen, auffinden (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 16; vgl. [Vahrenkamp 2003](#), S. 106). Zuordnungsmodelle werden zur Einsatzplanung von Mitarbeitern in Verkehrssystemen, z. B. Bus- und Bahnsystemen eingesetzt (vgl. [Vahrenkamp 2003](#), S. 106). Ähnlich zu Transportmodellen erfolgt die Zuordnung der Mitarbeiter zu ihren Arbeitsplätzen auf Basis einer Kostenmatrix, die sich aus den Zuordnungskosten der Mitarbeiter zu den Arbeitsplätzen ergibt (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 171).

Tourenplanung Eine Tour wird durch die Angabe der Mengen aller Kunden (Knoten) und Wege (Kanten) beschrieben, die auf ein und derselben Fahrt ausgehend von einem Depot bedient werden. Die Reihenfolge, in der die Kunden beliefert werden, wird als Route bezeichnet (vgl. [Scholl und Domschke 2007](#), S. 198).

Die Gestaltung der Touren erfolgt unter Einhaltung gegebener Restriktionen, wie Anzahl der Fahrzeuge, Kapazitäten und Zeitrestriktionen innerhalb der Tourenplanung. Das Ziel ist die Minimierung der Gesamttransportdistanz (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 276). Typische Problemstellungen der Tourenplanung betrachten eine Anzahl von Kunden innerhalb einer Region, deren Bedarfe und Standorte bekannt sind und die mit einer Anzahl von Fahrzeugen mit vorgegebenen Kapazitäten von einem Depot aus mit einem bestimmten Gut beliefert werden sollen (vgl. [Vahrenkamp 2003](#), S. 233). Probleme der Tourenplanung unterscheiden sich hinsichtlich Depot- und Kundencharakteristika, Fahr-

zeugcharakteristika, Problem- oder Zusatzcharakteristika sowie vorgegebener Zielsetzungen (vgl. [Scholl und Domschke 2007](#), S. 200).

Klassische Tourenprobleme stellen das Travelling-Salesman-Problem und das Briefträgerproblem dar (vgl. [Scholl und Domschke 2007](#), S. 203). Das knotenorientierte Travelling-Salesman-Problem behandelt die Planung einer Reise durch eine vorgegebene Anzahl an Orten, deren Entfernung paarweise bekannt ist. Das Ziel ist die Suche nach der kürzesten Route durch alle Orte (vgl. [Krischke und Röpcke 2015](#), S. 98). Das Travelling-Salesman-Problem findet Anwendung in der Paketauslieferung oder der Kommissionierung von Aufträgen in einem Lager (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 229). Innerhalb des kantenorientierten Briefträgerproblems soll von einem bestimmten Standort eine möglichst kurze Tour, die alle Kanten eines Graphen mindestens einmal durchläuft, geplant werden (vgl. [Scholl und Domschke 2007](#), S. 205). Das Briefträgerproblem tritt u. a. bei der Verteilung von Zeitungen und Briefen oder der Müllentsorgung auf (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 221).

Nachfolgend werden einige Probleme nach [Vahrenkamp](#) (vgl. [2003](#), S. 233–235), durch Ausführungen anderer Autoren ergänzt, vorgestellt. So kann anstatt eines Auslieferungsproblems ein Sammelproblem vorliegen, bei dem die Abholung von Gütern oder Recyclingprodukten erforderlich ist (Sammelprobleme). In den Mehrdepot-Auslieferungsproblemen werden mehrere Lagerstandorte zur Belieferung von Kunden betrachtet, wobei keine feste Zuordnung der Standorte zu Kunden existiert und jede Tour neu geplant wird. Weitere Tourenprobleme erfolgen unter Kapazitäts- oder Zeitrestriktionen, z. B. Tourenprobleme mit Kundenzeitfenstern (vgl. [Scholl und Domschke 2007](#), S. 204). Nachdem graphbasierte Modelle im logistischen Kontext erläutert wurden, behandelt [Abschnitt 3.4.2](#) Graphalgorithmen, die im Rahmen von Graphdatenbanken erforscht und implementiert werden.

3.4.2 Graph-Management-Algorithmen

Um Abfragen und Manipulationen auf Daten einer Graphdatenbank anzuwenden, müssen graphbasierte Abfragesprachen sowie -algorithmen entworfen werden (vgl. [Aggarwal und Wang 2010b](#), S. 2). In [Abschnitt 3.3.3](#) wurde Cypher als Beispiel einer graphbasierten Abfragesprache vorgestellt. In diesem Abschnitt werden Graph-Management-Algorithmen beschrieben. Ein Algorithmus umfasst eine endliche Menge von Instruktionen, die Operationen auf Daten ausführen (vgl. [Krumke und Noltemeier 2009](#), S. 16). In der Literatur lassen sich folgende Algorithmus-Eigenschaften ableiten ([Nahrstedt 2018](#), S. 3):

- Alle verwendeten Größen müssen bekannt sein
- Die Umarbeitung geschieht in Arbeitstakten
- Die Beschreibung des Algorithmus ist vollständig
- Die Beschreibung des Algorithmus ist endlich
- Alle angegebenen Operationen sind zulässig
- Angabe einer Sprache für die Regeln

[Nebel und Wild](#) (vgl. [2018](#), S. 293) beschreiben einen Graphalgorithmus als einen Algorithmus, der auf einem Graphen operiert. Es existiert eine Vielzahl an Graphalgorithmen, da Graphen für die Modellierung unterschiedlicher Sachverhalte und Zusammenhänge eingesetzt werden. Viele Anwendungen beruhen auf gemeinsamen Kernproblemen, wie die Bestimmung kürzester Wege oder die Berechnung eines minimalen spannenden Baumes.

Der Dijkstra-Algorithmus berechnet den kürzesten Weg eines Wurzelknotens zu einem beliebigen Knoten auf Basis eines gewichteten Graphen (vgl. [Meier und Kaufmann 2016](#), S. 63–66). Hierbei werden nicht negative Kantenbewertungen vorausgesetzt (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 80). In einem iterativen Verfahren werden Knoten, ausgehend von dem geringsten Kantengewicht, sukzessiv zu einem Kürzeste-Wege-Baum hinzugefügt (vgl. [Meier und Kaufmann 2016](#), S. 63–66). Im SCM werden durch den Dijkstra-Algorithmus bspw. die Länge des schnellsten Weges oder die Zeitdauer des kürzesten Weges berechnet (vgl. [Mattfeld und Vahrenkamp 2014](#), S. 83–86).

Im Gegensatz zu dem Dijkstra-Algorithmus, der die Suche nach einem Kürzeste-Wege-Baum behandelt, ist das Ziel des Kruskal-Algorithmus die Suche nach einem minimal aufspannenden Baum. Ein aufspannender Baum ist minimal, wenn die Summe seiner gesamten Kantengewichte minimal ist (vgl. [Krischke und Röpcke 2015](#), S. 66). Zur Berechnung eines minimal aufspannenden Baumes wird eine aufsteigend sortierte Kantenliste abgearbeitet. Kanten werden sukzessiv hinzugefügt, wenn das Netzwerk durch die Aufnahme zyklusfrei bleibt (vgl. [Vahrenkamp 2003](#), S. 58). In der Praxis werden minimal aufspannende Bäume zur kostengünstigen Verbindung von Knoten, wie z. B. beim Straßen- oder Schienennetz und bei der Sanierung des Energienetzes, verwendet (vgl. [Krischke und Röpcke 2015](#), S. 66).

Graphalgorithmen, die zur Abfrage von Graphdatenbanken oder zum Data-Mining erforscht werden, sind Gegenstand von [Abschnitt 3.4.2](#) und [Abschnitt 3.4.3](#). [Aggarwal und Wang](#) (vgl. [2010c](#), S. 15–40) differenzieren zwischen Graph-Management- und Graph-Mining-Algorithmen. Graph-Management-Algorithmen dienen zur Verwaltung und Manipulation von in einer Graphenrepräsentation gespeicherten Daten, wohingegen Graph-Mining-Algorithmen unbekannte Muster, Trends, Klassen und Cluster extrahieren.

Allerdings existiert keine einheitliche Klassifikation von Graphalgorithmen. So unterscheiden [Angles und Gutierrez](#) (vgl. [2018](#), S. 13–17) nicht zwischen Graph-Management- und Graph-Mining-Algorithmen, sondern fassen alle Algorithmen unter Graph-Abfragen zusammen. Jedoch heben die Autoren hervor, dass die Abfragen im Rahmen von Analytical-Queries die Basis für das Graph-Mining bilden und verweisen in diesem Zusammenhang auf [Aggarwal \(2010\)](#). Deshalb können Analytical-Queries eher dem Graph-Mining zugeordnet werden, so dass sich eine erste Einteilung von Graphalgorithmen abzeichnet. Da sich Graph-Mining-Algorithmen aufgrund ihrer Vorgehensweise von den Graph-Management-Algorithmen unterscheiden und dies auch in der Ausführung von [Angles und Gutierrez](#) (vgl. [2018](#), S. 13–17) ersichtlich wird, erfolgt die Einteilung der Graphalgorithmen in Anlehnung an [Aggarwal und Wang](#) (vgl. [2010c](#), S. 15). Zum einheitlichen Sprachgebrauch der nachfolgenden Graphalgorithmen werden die im Englischen verwendeten Begriffe übernommen und erläutert.

Adjacency-Queries Adjacency-Queries fassen einfache und komplexe Nachbarschaftsabfragen zusammen. Einfache Nachbarschaftsabfragen prüfen, ob Knoten oder Kanten benachbart sind. So werden bspw. Abfragen zum nächsten Nachbarn oder zur Existenz einer Kantenverbindung zwischen zwei Knoten formuliert (vgl. [Angles und Gutierrez 2018](#), S. 14). Komplexe Nachbarschaftsabfragen können z. B. alle Knoten innerhalb einer Weglänge k (k -Neighborhood) oder mit einer bestimmten Knoten-Distanz k (k -hops) zu dem Wurzelknoten berechnen (vgl. [Dominguez-Sal et al. 2011](#), S. 33; vgl. [Angles und Gutierrez 2018](#), S. 14). Typische Einsatzfelder sind Empfehlungssysteme oder soziale Netzwerke, um neue Freundschaften vorzuschlagen (vgl. [Dominguez-Sal et al. 2011](#), S. 33).

Reachability-Queries Nach Dominguez-Sal et al. (vgl. 2011, S. 31–33) werden Reachability-Queries zur Prüfung der Erreichbarkeit von Knoten eingesetzt. Durch Reachability-Queries können spezielle Kanten (Brücken), deren Entfernung die Trennung der Gruppe bedeuten würde, identifiziert werden. Auch können nach Dominguez-Sal der Zusammenhalt eines Graphen durch die Berechnung von minimal aufspannenden Bäumen abgeleitet werden.

Reachability-Queries werden auch für die Lösung von Pfadproblemen unter Berücksichtigung von Knoten- oder Kantenrestriktionen eingesetzt. So kann abgefragt werden, ob zwei Knoten durch eine Kante, die ein bestimmtes Attribut (z. B. „befreundet“) enthält, miteinander verbunden sind (vgl. Angles und Gutierrez 2018, S. 16; vgl. Aggarwal und Wang 2010c, S. 21). Reachability-Queries finden ihren Einsatz in sozialen Netzwerken, bspw. zur Suche nach gleichen Interessen, oder in biologischen Netzwerken, um spezifische biologische Verbindungen zu identifizieren. In Graphdatenbanken werden Reachability-Queries für diverse Pfadabfragen, z. B. kürzeste Wege, eingesetzt (vgl. Angles und Gutierrez 2018, S. 16).

Graph-Matching-Queries Graph-Matching-Queries identifizieren eine Menge von Teilgraphen, die mit einem gegebenen Muster übereinstimmen, unter Berücksichtigung von Charakteristika wie gleiche Knoten oder Kantenattribute oder gleiche Kantenverbindungen (vgl. Aggarwal und Wang 2010c, S. 21–22). Die gesuchten Muster werden in Form von Teilgraphen mit vorgegebenen und teilweise unbekanntem Attributeigenschaften dargestellt. Die Algorithmen ermöglichen u. a. die Suche nach Freundesfreunden, die ebenfalls miteinander befreundet sind (vgl. Angles und Gutierrez 2018, S. 14). Ein weiteres Problem ist das Auffinden eines maximal gemeinsamen Teilgraphen zwischen zwei Graphen (vgl. Aggarwal und Wang 2010c, S. 22). Auch können Muster zusammengeführt werden, indem sie durch Operatoren, z. B. Vereinigung oder Differenzen, miteinander kombiniert werden (vgl. Angles und Gutierrez 2018, S. 15).

Das Graph-Matching wird in exakte und inexacte Methoden unterteilt. So wird im Exact-Graph-Matching ein exakter Abgleich der Muster durchgeführt. Es werden nur Ergebnisse zurückgegeben, die exakt mit dem gesuchten Muster übereinstimmen (vgl. Bunke und Neuhaus 2007, S. 17). Auch das Auffinden von isomorphen Graphen ist Teil des Exact-Graph-Matching (vgl. Aggarwal und Wang 2010c, S. 22). Zwei Graphen sind isomorph, wenn sie die gleiche Anzahl von Knoten besitzen und bijektiv zueinander sind, d. h. trotz unterschiedlicher Darstellung ist die grundsätzliche Struktur der beiden Graphen identisch (vgl. Krischke und Röpcke 2015, S. 18). Da ein exakter Abgleich nicht immer durchführbar ist, werden inexacte Musterabgleiche durchgeführt. Hierbei werden mehrere Muster zurückgegeben, die dem gesuchten Muster am meisten ähneln und innerhalb einer definierten Toleranz liegen (vgl. Aggarwal und Wang 2010c, S. 22–23).

Nach Gallagher (vgl. 2006, S. 46) wird darüber hinaus zwischen Mustererkennung anhand der Struktur oder der Semantik unterschieden. Die herkömmlichen Vorgehensweisen des Graph-Matching fokussieren einen Musterabgleich auf Basis von Graph-Strukturen. Das Semantic-Matching zielt auf einen Musterabgleich ab, basierend auf der Bedeutung der Muster. Hierbei werden auch die Attribute sowie Typen der Knoten und Kanten berücksichtigt, um gleiche oder ähnliche Musterinterpretationen aufzufinden. Typische Einsatzfelder des Graph-Matching sind soziale Netzwerke, um Beziehungsmuster zu finden. Auch im semantischen Web wird das Graph Matching zur Suche nach semantischen Zusammenhängen eingesetzt (vgl. Dominguez-Sal et al. 2011, S. 28).

Das Graph-Matching bildet nach [Gallagher](#) (vgl. [2006](#), S. 46) die Ausgangsbasis des Graph-Pattern-Minings (GPM). Im Gegensatz zum Graph-Matching, das zum Auffinden bestimmter Muster eingesetzt wird, zielt das GPM auf die Suche nach geläufigen oder interessanten Mustern ab. [Scheidler](#) (vgl. [2017](#), S. 53) unterscheidet, nach [Klößen und Zytkow \(1996\)](#), zwischen Mustern als Ergebnis eines komplexen Data-Mining-Verfahrensschritts und einfachen Mustern, die bereits vor dem Data-Mining identifizierbar sind. Auf das Graph-Matching und das GPM übertragen, hat das Graph-Matching einfache Muster als Ergebnis, wohingegen das GPM Muster durch Graph-Mining extrahiert. Eine Beschreibung von Mustern, Data-Mining, Graph-Mining und Graph-Mining-Algorithmen erfolgt im nächsten Abschnitt.

3.4.3 Graph-Mining-Algorithmen

Nach [Bissantz und Hagedorn](#) (vgl. [2009](#), S. 139–141) kann der Begriff Data-Mining mit Datenmustererkennung übersetzt werden. Das Data-Mining wird zur Entdeckung von Wissen eingesetzt, das durch Datenmustererkennungsmethoden auf Basis von Aussagen, Regeln und Informationen extrahiert wird. [Bissantz und Hagedorn](#) (vgl. [2009](#), S. 139–141) charakterisieren die gesuchten Muster als Beziehungen zwischen Datensätzen bzw. zwischen den Daten eines Satzes oder bestimmte Regelmäßigkeiten. Jedoch existieren unterschiedliche Musterbegriffe, die je nach Betrachtungsweise variieren. So beschreibt [Petersohn](#) (vgl. [2005](#), S. 10) Muster als Schema, Vorbild oder Modell. Auch stellen interessante Muster nach [Runkler](#) (vgl. [2010](#), S. 2) Wissen dar. Nachfolgend wird der Musterbegriff von Bissantz und Hagedorn verwendet und durch Runkler ergänzt:

Definition 3.4 Muster: Muster bezeichnen Beziehungen zwischen Datensätzen bzw. zwischen Daten eines Satzes oder bestimmte Regelmäßigkeiten. Weiterhin stellen interessante Muster Wissen für den Anwender dar (vgl. [Bissantz und Hagedorn 2009](#), S. 139–141; vgl. [Runkler 2010](#), S. 2).

Je nach Analysekontext sind unterschiedliche Muster für den Anwender von Interesse (vgl. [Petersohn 2005](#), S. 10). Im Marketing sind bspw. Muster über das Kaufverhalten des Kunden interessant, wohingegen in der Betrugserkennung Transaktionsmuster, die auf Geldwäsche deuten, im Fokus stehen (vgl. [Fayyad 1996](#), S. 38). Weiterhin können bspw. Muster wie Sprach-, Waren-, Gebrauchs-, Geschmacks-, Bild-, Text-, Verhaltens-, Prozess- oder Problemmuster betrachtet werden (vgl. [Petersohn 2005](#), S. 10–11).

Die verborgenen Muster werden durch die Anwendung spezieller Algorithmen extrahiert (vgl. [Fayyad 1996](#), S. 38–39). Algorithmen des Data-Minings zeichnen sich durch folgende Charakteristika aus (vgl. [Bissantz und Hagedorn 2009](#), S. 139):

- Das Wissen wird in einer verständlichen Form präsentiert oder von einem Programm weiterverarbeitet (Datenvisualisierung).
- Die Aussagen sind mit einer bestimmten Sicherheit versehen, d. h. existent.
- Das Wissen ist gemessen am im System vorhandenen Wissen oder Anwenderwissen nicht trivial.
- Die Laufzeit der Algorithmen bewegt sich in einem vertretbaren Rahmen.

Das Data-Mining ist nicht mit dem Prozess der Wissensentdeckung in Datenbanken (Knowledge Discovery in Databases) (KDD) gleichzusetzen. So stellt das Data Mining lediglich

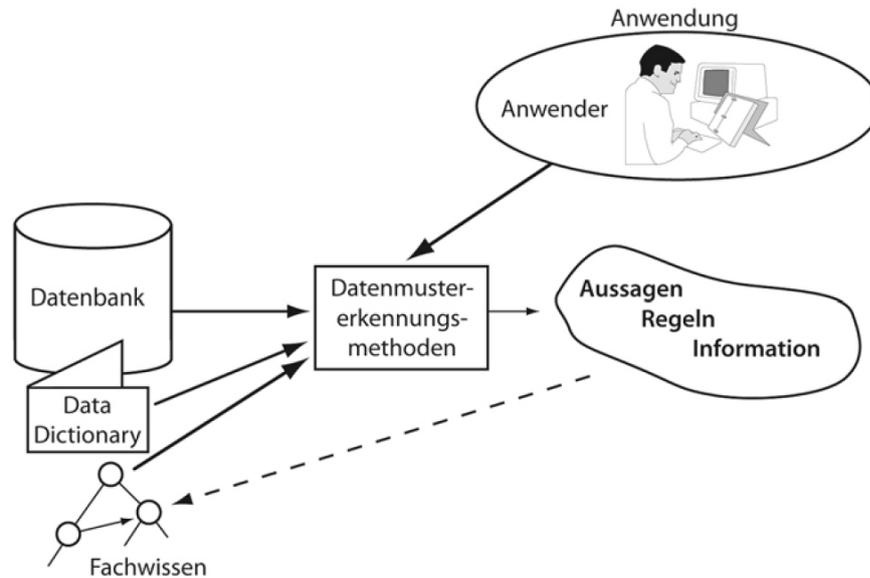


Abbildung 3.2: Rahmenkonzept des Data-Minings (Bissantz und Hagedorn 2009, S. 141)

eine zentrale Phase des KDD dar (vgl. Fayyad 1996, S. 39–41). Im Gegensatz zu dem Data-Mining bezieht sich das KDD nicht nur auf die Ermittlung und Abbildung von Mustern in Datenbeständen, sondern auch auf die Auswahl und Aufbereitung der Daten, die Festlegung der Analyseverfahren sowie die Interpretation und Evaluierung des gewonnenen Wissens (vgl. Düsing 2006, S. 252–253).

Analog zum Data-Mining extrahiert das Graph-Mining Muster aus einer Graphenrepräsentation, um zweckgerichtetes Wissen zu generieren. (vgl. Holder und Cook 2007, S. 3). Chakrabarti (vgl. 2010, S. 469) definiert das Graph-Mining als eine Reihe von Werkzeugen und Techniken, um Eigenschaften von Graphen zu analysieren. Zusätzlich wird es genutzt, um vorherzusagen, wie sich die Struktur und die Eigenschaften eines gegebenen Graphen auf eine bestimmte Anwendung auswirken könnte.

Die Charakteristiken von Data-Mining-Algorithmen werden nur teilweise von den vorgestellten Graph-Mining-Algorithmen erfüllt (vgl. Aggarwal und Wang 2010c, S. 29–40). Ein Abgleich aller Charakteristika der beiden Algorithmen-Klassen wurde in der Literatur nicht gefunden. Aggarwal und Wang (vgl. 2010c, S. 29) untersuchen Algorithmen, die zur Extraktion von Mustern, Trends, Klassen und Clustern auf Graphen eingesetzt werden. Im Gegensatz zu den Graph-Management-Algorithmen werden keine Informationen auf Basis vorgegebener Muster gesucht, vielmehr zielen Graph-Mining-Algorithmen auf das Auffinden von neuem und für den Anwender interessantem Wissen ab (vgl. Gallagher 2006, S. 46; vgl. Aggarwal und Wang 2010c, S. 15). Das Charakteristikum der Datenvisualisierung erfüllt sich durch die Graphenvisualisierung auf Basis von Graph-Mining-Algorithmen, die die Ergebnisse für den Anwender visuell aufbereiten (vgl. Eades und Klein 2018, S. 33). Auch die Berücksichtigung der Sicherheit von Aussagen, bspw. durch Nutzung von Schwellenwerten, oder Laufzeitoptimierungen werden in einigen Graph-Mining-Algorithmen berücksichtigt (vgl. Yan und Han 2007, S. 101; vgl. Aggarwal und Wang 2010c, S. 29–40). Nachfolgend werden ausgewählte Graph-Mining-Algorithmen vorgestellt.

Graph-Pattern-Mining-Algorithmen Das GPM zielt auf das Entdecken von verborgenen und für den Anwender interessanten Mustern, z. B. charakteristische oder regelmäßig auftretende Muster, in Graphen ab (vgl. [Chakrabarti 2010](#), S. 69–71; vgl. [Aggarwal und Wang 2010c](#), S. 29). Auch lassen sich durch GPM-Algorithmen Graphen, die in mindestens einem Knoten identisch sind, zusammenfügen (vgl. [Rao und Mitra 2014](#), S. 197–198). Je nach Einsatzfeld erfolgen durch das GPM die Suche nach chemischen Strukturen (vgl. [Ramraj und Prabhakar 2015](#), S. 198), die Entdeckung von Anomalien in der IT-Sicherheit (vgl. [Abdelhamid et al. 2017](#), S. 2711) oder die Suche nach zusammenhängenden Objekten (vgl. [Tiwari et al. 2010](#), S. 309). Aktuelle Forschungen zum Einsatz des GPM in der Logistik beschreiben einen Ansatz zur Optimierung der Kommissionierung (vgl. [Ito und Kato 2016](#), S. 114–115). Im Gegensatz zum Graph-Matching werden die Muster durch GPM-Algorithmen ohne vorgegebene Muster des Anwenders identifiziert (vgl. [Aggarwal und Wang 2010c](#), S. 29–32).

Zur Identifikation häufiger Muster werden die betrachteten Datensätze, eine Signifikanzfunktion zur Bewertung der Muster in Form von Teilgraphen und ein Schwellenwert benötigt (vgl. [Khan und Ranu 2017](#), S. 556). Die identifizierten Teilgraphen werden als Kandidaten bezeichnet und im Rahmen der Kandidatengenerierung extrahiert (vgl. [Yan und Han 2007](#), S. 100–101). Im nächsten Schritt werden die Teilgraphen mit dem Schwellenwert verglichen. Teilgraphen, deren Signifikanz höher als der gegebene Schwellenwert ist, werden in die Lösungsmenge mit aufgenommen und als häufig vorkommende Muster bezeichnet (vgl. [Khan und Ranu 2017](#), S. 556). Die Suche nach häufig vorkommenden Mustern kann entweder auf einer Gruppe von Graphen, z. B. Transaktionen, oder auf einen einzelnen Graphen angewendet werden (vgl. [Aggarwal und Wang 2010c](#), S. 29). Neben der Suche nach häufig vorkommenden Mustern ist auch die Suche nach maximal vorkommenden oder signifikanten Mustern Teil des GPM (vgl. [Aggarwal und Wang 2010c](#), S. 29–30). Die extrahierten Muster können für weitere Graph-Mining-Prozesse, wie die Klassifikation oder das Clustering von Graphen, genutzt werden (vgl. [Han et al. 2012](#), S. 591–592; vgl. [Abdelhamid et al. 2017](#), S. 2721). Die Erläuterung der beiden Verfahren erfolgt in den nachfolgenden Abschnitten.

Graph-Clustering-Algorithmen Das Ziel des Graph-Clustering ist die Suche nach möglichst homogenen, d. h. ähnlichen oder stark vernetzten, Entitäten und ihre Aufteilung in disjunkte Gruppen (Clusters) (vgl. [Flake et al. 2004](#), S. 385). Ein weiteres Ziel ist es, möglichst viele Kanten innerhalb eines Clusters zu erhalten und die Anzahl der Kanten in anderen Clustern zu minimieren (vgl. [Schaeffer 2007](#), S. 27). Graph-Clustering-Algorithmen werden auf Graphen angewendet, um ein Clustering als Output auszugeben (vgl. [Schaeffer 2007](#), S. 31). Die Graphalgorithmen teilen sich wie folgt auf (vgl. [Aggarwal 2010](#), S. 32–33):

- **Node-Clustering:** Die Algorithmen des Node-Clustering werden auf einen Graphen angewendet. Durch Distanz- und Ähnlichkeitsfunktionen werden die Knoten in Cluster unterteilt. Hierbei erhalten die Kanten numerische Distanzwerte, auf deren Basis Knoten-Gruppen gebildet werden können. Zu den Aufgaben des Node-Clustering zählen bspw. die Suche nach dem minimalen Schnitt und die Graphpartitionierung.
- **Graph-Clustering:** In diesem Fall wird eine Anzahl an Graphen betrachtet, die auf Basis von Strukturierungsregeln in Cluster eingeteilt werden. Dieses Problem stellt jedoch eine Herausforderung dar, da die zugrunde liegenden Datenstrukturen erkannt und zur Clusterbildung genutzt werden müssen. Die Algorithmen werden sowohl auf klassischen Graphdaten als auch auf semistrukturierte Daten angewendet.

Die Suche nach dem minimalen Schnitt eines Graphen zielt auf die Unterteilung des Graphen in zwei Cluster ab. Dabei wird berücksichtigt, dass die Summe der Kantengewichte zwischen den zwei Gruppen minimal ist (vgl. [Aggarwal und Wang 2010a](#), S. 277). Hierbei stellt ein Kantengewicht eine Kapazität zwischen einem Knoten (Quelle) und einem anderen Knoten (Senke) dar (vgl. [Schaeffer 2007](#), S. 42). Im Zusammenhang mit den minimalen Schnitten stehen die maximalen Flüsse (vgl. [Flake et al. 2004](#), S. 386). Nach [Aggarwal und Wang](#) (vgl. [2010a](#), S. 277–279) beschreibt ein maximaler Fluss die maximale Menge, z. B. Materialien, die von der Quelle über verschiedene Knoten bis hin zur Senke unter Einhaltung der Kapazitätsbeschränkungen in einen gerichteten Graphen fließen kann. Innerhalb dieses Graphen hat der Wert eines maximalen Flusses den Wert eines minimalen Schnittes.

Die Unterteilung eines Graphen durch Schnitte in mehrere Cluster wird als Graphpartitionierung bezeichnet (vgl. [Aggarwal und Wang 2010c](#), S. 32). Die Graphpartitionierung identifiziert und unterteilt dichte Bereiche durch Schnitte, so dass die Anzahl der Kanten zwischen den Bereichen minimal wird (vgl. [Schaeffer 2007](#), S. 31–33; vgl. [Aggarwal und Wang 2010a](#), S. 281–282).

[Aggarwal und Wang](#) (vgl. [2010a](#), S. 295–296) beschreiben eine Vielzahl von Einsatzgebieten für das Graph-Clustering. So lassen sich Graphen durch Clustering-Algorithmen komprimieren und zusammenfassen, so dass die Gesamtstruktur des Netzwerkes problemlos erfasst werden kann. Auch zur Bestimmung von Standorten, dichten Regionen oder zur Analyse von molekularen Strukturen wird das Graph-Clustering eingesetzt.

Im SCM stellen [Irfan et al.](#) (vgl. [2007](#), S. 422–424) einen K-Algorithmus vor, der die betrachteten SCM-Daten in 1 bis k Cluster einteilt und die Cluster-Zentren iterativ bestimmt. Ein Clusterzentrum ist ein Knoten, der im Durchschnitt die kürzesten Wegstrecken zu den anderen Knoten innerhalb des Knotens aufweist. Irfan et al. wenden den Algorithmus auf unterschiedliche Datensätze an (Material, Lieferanten, Produzenten, Distribution, Einzelhandel, Endkunden), so dass unterschiedliche Cluster entlang der SC entstehen. So dient die Analyse der Einzelhandels- und Kundendaten der Beobachtung des Konsumverhaltens und der Bildung von Kundengruppen. Auf Basis der Cluster lassen sich Muster, Prognosen oder das Kaufverhalten der Kunden extrahieren, so dass die einzelnen Kundengruppen zielgerecht beworben werden können. Auch werden auf Ebene des Einzelhandels Produktcluster gebildet, so dass Bedarfsprognosen vereinfacht werden. Auch betrachten Irfan et. al die Beziehungen zwischen den SC-Partnern, um Flüsse zwischen Hersteller und Lieferanten zu optimieren.

Graph-Classification-Algorithmen Graph-Classification-Algorithmen dienen im Graph-Mining zur Klassifizierung. Sie werden zur Beschreibung und Einordnung von Gruppen eingesetzt. Durch das wachsende Interesse von Forschung und Industrie an Graphen wird der Fokus verstärkt auf den Einsatz von Graph-Classification-Algorithmen gelegt (vgl. [Aggarwal und Wang 2010c](#), S. 37). Hierbei werden entweder einzelne Knoten oder ganze Graphen betrachtet, so dass sich die Graph-Classification-Algorithmen wie folgt unterscheiden (vgl. [Aggarwal und Wang 2010c](#), S. 37):

- **Label-Propagation:** Ein Teil einzelner Knoten in einem Graphen ist mit einem Label versehen. Auf Basis dieser Knoten soll ein Modell erlernt werden, das die restlichen Knoten ohne Label jeweils in vorhandene Klassen zuordnet, so dass am Ende alle Knoten beschriftet sind.
- **Graph-Classification:** Eine Teilmenge von Graphen aus einer Graphenmenge ist mit Labeln versehen. Auch hier ist es das Ziel, ein Modell zur Klassifizierung zu erler-

nen, wobei keine einzelnen Knoten, sondern unbeschriftete Graphen auf Grundlage der bereits beschrifteten Graphen in vorhandene Klassen klassifiziert werden.

In sozialen Netzwerken erfolgt die Klassifizierung von Personengruppen durch Zuordnung von Labels, um Gruppen, die positiv auf eine bestimmte Werbung reagieren, zu identifizieren. So sollen, basierend auf den beschrifteten Knoten, die Reaktionen anderer Menschen auf diese Werbung hergeleitet werden können (vgl. [Tsuda und Saigo 2010](#), S. 358).

[Riesen und Bunke](#) (vgl. [2009](#), S. 1067–1071) stellen unterschiedliche Experimente zum Einsatz von Graph-Classification-Algorithmen vor, indem sie die betrachteten Datensätze in einer Graphenrepräsentation einspeichern. So werden z. B. Bilder in unterschiedliche Klassen (Stadt, Land, Schnee, Menschen) eingeteilt oder Fingerabdrücke anhand ihrer Merkmale charakterisiert. Auch die Untersuchung molekularer Strukturen auf aktive und inaktive Moleküle kann mittels Graph-Classification durchgeführt werden.

Link-Prediction Die Entdeckung fehlender bzw. Vorhersage zukünftiger Kanten zwischen zwei Knoten wird für die Analyse von Netzwerken, z. B. zur Vorhersage von entstehenden Freundschaften in einem sozialen Netzwerk, verwendet (vgl. [Namata und Getoor 2010](#), S. 609). In wissenschaftlichen Netzwerken kann die Mitautorenschaft zwischen zwei Wissenschaftlern bestimmt werden, die noch nicht gemeinsam veröffentlicht haben. So erhöht sich die Wahrscheinlichkeit zur Zusammenarbeit, wenn sie viele gemeinsame Kollegen aufweisen, ähnliche Orte aufsuchen oder sich ihre geografische Distanz aufgrund eines Arbeitsortwechsel verringert hat (vgl. [Liben-Nowell und Kleinberg 2003](#), S. 556). Weitere Einsatzfelder sind biologische Netzwerke, bspw. um Proteininteraktionen zu untersuchen (vgl. [Lü und Zhou 2011](#), S. 1151).

Zur Suche möglicher Kanten werden ausgehend von einem bestimmten Zeitpunkt t in einem Netzwerk bis zu einem zukünftigen Zeitpunkt t' fehlende Kanten vorhergesagt, die dem Netzwerk innerhalb dieses Zeitintervalls hinzugefügt werden könnten (vgl. [Liben-Nowell und Kleinberg 2003](#), S. 556). Nach [Namata und Getoor](#) (vgl. [2010](#), S. 609–610) werden die möglichen zukünftigen Kanten als potenzielle Kanten bezeichnet. Die Anzahl der potenziellen Kanten ist abhängig vom Graphentypen und von der Definition der Kanten des Graphen. Die Realisierung der möglichen Kanten erfolgt durch einen Vergleich mit einem Wahrscheinlichkeitswert als Schwellenwert. Übersteigt die Wahrscheinlichkeit der Existenz einer Kante diesen Schwellenwert, wird die Kante realisiert. Abschließend beschreiben [Namata und Geetor](#), neben der Prüfung möglicher Kanten, die Prüfung bereits vorhandener Kanten, um überflüssige Kanten zu identifizieren.

In [Kapitel 3](#) wurden die wissenstheoretischen Grundlagen von Daten und Informationssystemen und ihre Rolle zur Beantwortung logistischer Fragestellungen erarbeitet. Auch wurden RDBS sowie GDBS vorgestellt und Vorteile einer Graphenrepräsentation dargestellt. Zum Schluss wurden graphbasierte Modelle sowie Graphalgorithmen erläutert, die in [Kapitel 4](#) als Basis zur Untersuchung graphbasierter Lösungsansätze zur Beantwortung logistischer Fragestellungen dienen.

4 Untersuchung graphbasierter Verfahren zur Beantwortung logistischer Fragestellungen des Supply-Chain-Managements

In [Kapitel 2](#) und [Kapitel 3](#) wurden die wissens-theoretischen Grundlagen zu SC, SCM, Informationssystemen und graphbasierter Verfahren erörtert. In [Kapitel 4](#) erfolgt die Untersuchung graphbasierter Lösungsansätze zur Beantwortung logistischer Fragestellungen, die nachfolgend als Fragestellungen bezeichnet werden.

Einleitend werden in [Abschnitt 4.1](#) wissens-theoretische Grundlagen zur Erstellung einer Taxonomie erläutert, um eine Taxonomie zur Klassifizierung von Fragestellungen zu entwickeln. Mithilfe der Taxonomie sollen die Fragestellungen systematisch beantwortet werden können. In [Abschnitt 4.1.2](#) wird eine Kategorisierung und Beschreibung graphbasierter Verfahren durchgeführt. Auf Basis der Ergebnisse aus [Abschnitt 4.1.1](#) und [Abschnitt 4.1.2](#) erfolgt die Untersuchung graphbasierter Verfahren in [Abschnitt 4.2](#) und der Herleitung graphbasierter Lösungsansätze. [Kapitel 4](#) schließt mit der Gesamtbewertung und Zusammenfassung der Lösungsansätze.

4.1 Vorbereitung zur Erstellung einer Bewertungsmatrix

Nach [Schmaltz](#) (vgl. [2004](#), S. 5) werden in einer Taxonomie hierarchisch gegliederte Klasseneinteilungen eines bestimmten Gegenstandsbereiches der Realität vorgenommen. Sie enthalten Über- und Unterordnungsbeziehungen, so dass Vererbungsrelationen und Zusammenhänge abgebildet werden können. Taxonomien sind ein grundlegendes Konzept zur Wissensorganisation eines betrachteten Gebietes und dienen bspw. zur Strukturierung von Inhalten in Intranets.

[Nickerson et al.](#) (vgl. [2013](#), S. 344–347) geben eine Methode zur Entwicklung von Taxonomien, die eine Kombination aus deduktiver und induktiver Herangehensweise vorsieht. Zur Entwicklung der Taxonomie werden zuerst Metacharakteristika und Abbruchkriterien bestimmt. Anschließend können sowohl eine deduktive als auch eine induktive Herangehensweise rekursiv durchlaufen werden. So schließt die induktive Herangehensweise von Objekten auf Charakteristika, die zu Klassen der Taxonomie führen, wohingegen in der deduktiven Herangehensweise zuerst Klassen gebildet werden. Die Herangehensweisen werden so lange durchlaufen, bis ein Abbruchkriterium erfüllt wird. Die nachfolgende Taxonomie orientiert sich an der Vorgehensweise von [Nickerson et al.](#), wobei die Klassen an das Aufgabenmodell aus [Abbildung 2.2](#) angelehnt sind.

4.1.1 Erstellung einer Taxonomie zur Klassifizierung von Fragestellungen

In [Abschnitt 4.1.1](#) erfolgt die Unterteilung der Aufgaben aus [Abschnitt 2.3](#) in Klassen, bestehend aus Aufgabenebenen, Aufgabenbereiche und untergeordneten Aufgabenstellungen, die in eine Taxonomie überführt werden sollen. Durch die Taxonomie wird die Einordnung von Fragestellungen zu übergeordneten Aufgabenstellungen erleichtert. Hierbei bilden die Gestaltungs- und Planungsebene aus dem SCM-Aufgabenmodell aus [Abbildung 2.2](#) die obersten Klasse (Aufgabenebene) der Taxonomie. Die Unterklassen der Aufgabenebene setzen sich aus den einzelnen Aufgaben des SCM-Aufgabenmodells, die als Aufgabenbereiche bezeichnet werden, zusammen. Aus [Abschnitt 2.3.1](#) und [Abschnitt 2.3.2](#) werden Aufgabenstellungen hergeleitet, die den jeweiligen Aufgabenbereichen als Unterklassen zugeordnet werden. Die Aufgabenstellungen werden anhand der in der Literatur gängigen Lösungsansätze charakterisiert. Die Taxonomie bietet eine grafische Darstellung und Strukturierung logistischer Aufgaben und ermöglicht die Klassifizierung von Fragestellungen in Aufgabenstellungen. So können die Fragestellungen mithilfe der gegebenen Lösungsansätze im Rahmen von Aufgabenstellungen beantwortet werden. Eine Validierung der Taxonomie erfolgt in [Abschnitt 5.1](#) durch die Einordnung ausgewählter Fragestellungen in die Taxonomie.

Die Gestaltungsebene Die Gestaltungsebene bildet die oberste Aufgabenebene. Im Rahmen der obersten Ebene findet die Gestaltung der SC statt (vgl. [Abschnitt 2.3.1](#)). In der Literatur werden die logistischen Aufgabenbereiche auf der Gestaltungsebene unter „strategische Netzwerkgestaltung“ zusammengefasst und nicht näher unterteilt (vgl. [Abschnitt 2.2.2](#)). Jedoch teilt sich die strategische Netzwerkgestaltung in [Abschnitt 2.3.1](#) in drei Phasen auf: die Erstellung von Planungsszenarien, die Entwicklung von Gestaltungsalternativen und die Bewertung sowie Auswahl der Alternativen. Aus den Phasen ergeben sich die Modellierung der SC, die Simulation der Unternehmensumwelt, die geografische Standortbestimmung, die Netzwerkkonfiguration, die Auswahl von Gestaltungsalternativen, die Auswahl von SC-Partnern und die Bestimmung von Standortrollen. Hierbei sind die Phasen als Aufgabenbereiche, die untergeordneten Aufgaben als Aufgabenstellungen zu verstehen. Der strategischen Netzwerkgestaltung werden keine Aufgabenstellungen zugeordnet, da sie im Rahmen der untergeordneten Aufgabenbereiche behandelt werden.

Es ist zu berücksichtigen, dass die drei Phasen aufeinander aufbauen und eng miteinander verknüpft sind. Deshalb ist eine integrative Betrachtung der drei Aufgabenbereiche notwendig, da sich die Ergebnisse der Aufgabenstellungen gegenseitig beeinflussen können (vgl. [Abschnitt 2.3.1](#)).

Erstellung von Planungsszenarien Die Erstellung von Planungsszenarien bildet den ersten Aufgabenbereich der strategischen Netzwerkgestaltung (vgl. [Abschnitt 2.3.1](#)). Aus [Abschnitt 2.3.1](#) ergeben sich als Aufgabenstellungen die Modellierung der SC und die Simulation der Unternehmensumwelt. Im Rahmen der strategischen Netzwerkgestaltung wird die SC als Ausgangslage für nachfolgende Aufgabenstellungen modelliert. Hierbei soll die Modellierung eine allgemeine Übersicht der Standorte und ihrer Beziehungen geben, wobei nur relevante Eigenschaften modelliert werden. Im Fokus steht das Verständnis der gesamten SC (vgl. [Abschnitt 2.3.1](#)). Mittels ökonomischer und stochastischer Methoden werden unterschiedliche Entwicklungen der Unternehmensumwelt innerhalb einer Szenarioanalyse simuliert (vgl. [Abschnitt 2.3.1](#)).

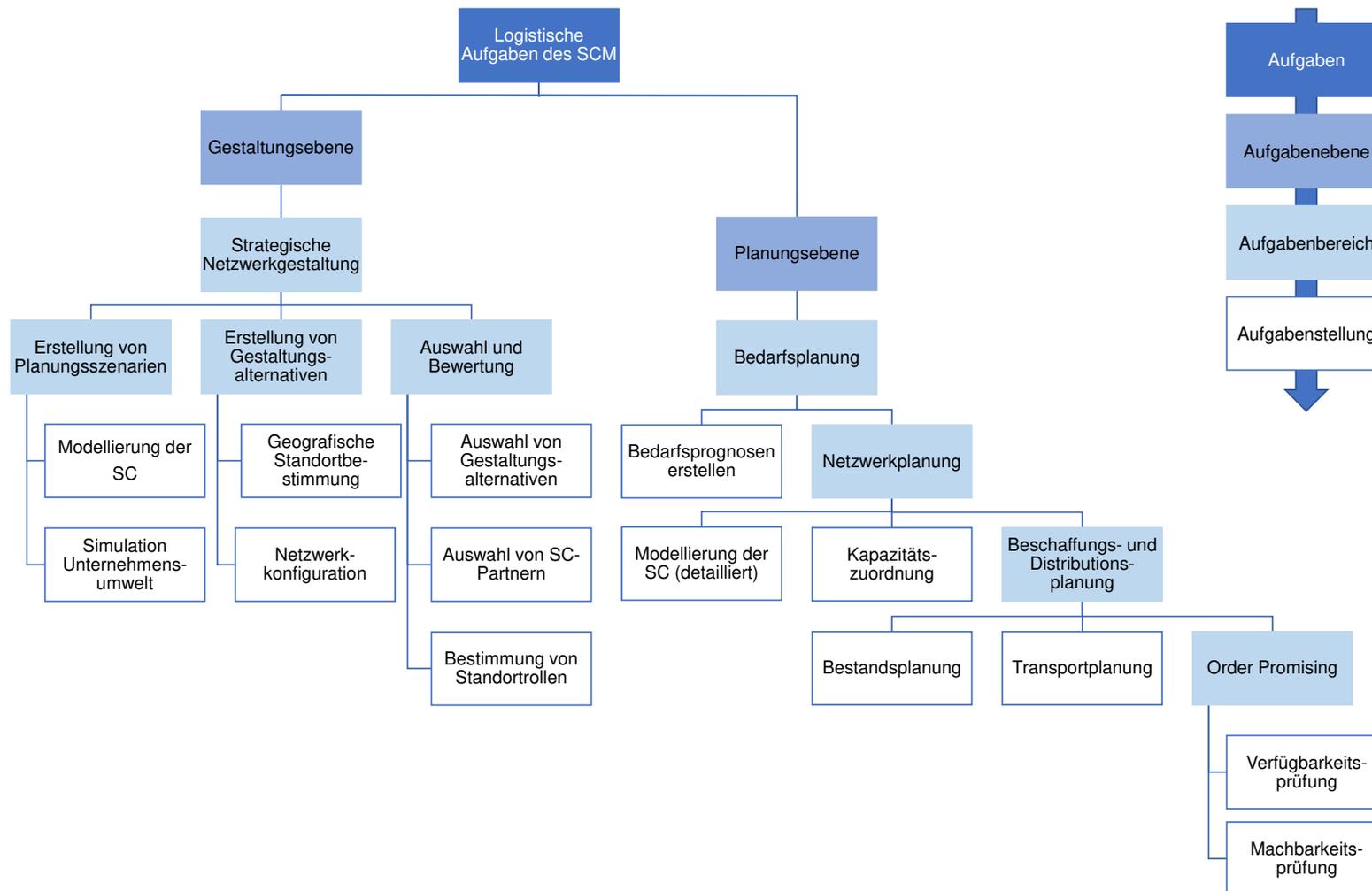


Abbildung 4.1: Taxonomie zur Klassifizierung logistischer Fragestellungen

Durch Parametervariationen entstehen unterschiedliche Planungsszenarien, die als Planungsumgebungen der Gestaltungsalternativen genutzt werden (vgl. [Abschnitt 2.3.1](#)). Folgende Lösungsansätze ergeben sich aus der Erstellung von Planungsszenarien (vgl. [Abschnitt 2.3.1](#)):

- SC-Modellierung
- Szenarioanalysen
- Ökonometrische oder stochastische Methoden

Erstellung von Gestaltungsalternativen Ein weiterer Aufgabenbereich der strategischen Netzwerkgestaltung ist die Erstellung verschiedener Planungsalternativen. Hieraus ergeben sich die geografische Standortbestimmung und die Netzwerkkonfiguration als untergeordnete Aufgabenstellungen (vgl. [Abschnitt 2.3.1](#)). So können zur Modifikation des vorhandenen Netzwerkes neue Standorte eröffnet oder das bestehende Netzwerk auf Optimierungspotenziale untersucht werden. Die geografische Standortbestimmung erfolgt mithilfe quantitativer Standortmodelle (vgl. [Abschnitt 3.4.1](#)). Ein vorhandenes Netzwerk kann durch Simulation unter Berücksichtigung von Zielgrößen (z. B. Kosten) optimal konfiguriert werden, so dass sich folgende Lösungsansätze ergeben (vgl. [Abschnitt 2.3.1](#)):

- Quantitative Standortmodelle
- Simulationsbasierte Optimierung

Auswahl und Bewertung Der dritte Aufgabenbereich umfasst die Auswahl der Gestaltungsalternativen und SC-Partner sowie die Bestimmung von Standortrollen (vgl. [Abschnitt 2.3.1](#)). In der Literatur werden die drei Aufgabenstellungen meist getrennt voneinander betrachtet. Jedoch ähneln sich ihre Lösungsansätze, so dass sie einem Aufgabenbereich zugeordnet werden. Alle drei Aufgabenstellungen benötigen Bewertungsverfahren, die sowohl quantitative als auch qualitative Merkmale berücksichtigen. Als Beispiele werden in [Abschnitt 2.3.1](#) Checklisten, Nutzwertanalysen und Rollenmodelle genannt (vgl. [Abschnitt 2.3.1](#)). Aus [Abschnitt 2.3.1](#) ergeben sich folgende Lösungsansätze:

- Multikriterielle Bewertungsverfahren
- Sammlung und Auswertung quantitativer sowie qualitativer Faktoren

Aufgrund der Aufgabenkomplexität auf der Gestaltungsebene werden die Aufgabenbereiche meist iterativ durchlaufen, bis eine endgültige Lösung definiert wird. Die festgelegte Netzwerkkonfiguration bildet den Planungsrahmen für die nachfolgende Aufgabenebene (vgl. [Abschnitt 2.3.1](#)).

Die Planungsebene ist für die Umsetzung der vorgegebenen Netzwerkkonfiguration verantwortlich und fokussiert die monetäre Gewinnmaximierung der SC (vgl. [Abschnitt 2.3.2](#)). Im Gegensatz zur Gestaltungsebene wird die Planungsebene in der Literatur detailliert in Aufgabenbereiche unterteilt, weshalb sich die Taxonomie an dieser Aufteilung orientiert (vgl. [Abbildung 2.2](#)). Ein weiterer Unterschied ist die hierarchische Abhängigkeit zwischen den Aufgabenbereichen. Zwar bauen die Aufgabenbereiche der strategischen Netzwerkplanung aufeinander auf, jedoch werden die Bereiche iterativ durchlaufen, so dass die Bereiche wechselseitige Rahmenbedingungen und Restriktionen setzen (vgl. [Abschnitt 2.3.1](#)). In der Planungsebene ist das iterative Durchlaufen der Aufgabenbereiche weniger ausgeprägt, so dass die Vorgaben hierarchisch an die untergeordneten Bereiche weitergegeben werden. So

gibt die Bedarfsplanung die notwendigen Vorgaben für die Netzwerkplanung vor, die wiederum zusammen mit der Bedarfsplanung den Planungsrahmen für die Beschaffungs-, Produktions- und Distributionsplanung setzen. Das Order Promising benötigt Zugriff auf die Daten aus den zuvor beschriebenen Planungsaufgaben, um die Verfügbarkeit und Machbarkeit der Aufträge zu prüfen (vgl. [Abschnitt 2.3.2](#)).

Nachfolgend sind Aufgabenstellungen der Produktionsplanung nicht aufgenommen worden, da sie meist nicht unternehmensübergreifend innerhalb des Logistikmanagements geplant werden (vgl. [Abschnitt 2.3.2](#)). Auch wird die Einteilung der Aufgabenbereiche des SCM-Aufgabenmodells aus [Abbildung 2.2](#) in taktische, operative und exekutive Aufgabenstellungen nicht in die Taxonomie aufgenommen. Im Fokus steht die Beantwortung logistischer Fragestellungen, so dass eine Einteilung aus betriebswirtschaftlicher Sicht zweitrangig ist.

Bedarfsplanung Die Bedarfsplanung ist der erste Planungsbereich in der Planungsebene. Dazu werden Prognosen hinsichtlich der Nachfrage, Kosten, Preise und weiterer relevanter Planungsgrößen in verschiedenen Märkten erstellt (vgl. [Abschnitt 2.3.2](#)). Die Erstellung von Bedarfsprognosen basiert auf vergangenen und aktuellen Daten, die durch entsprechende Prognoseverfahren ausgewertet werden. Durch die Auswertung der Daten lassen sich bspw. Nachfragemuster oder Ursache-Wirkungsbeziehungen erkennen, die sich auf zukünftige Bedarfe auswirken könnten (vgl. [Abschnitt 2.3.2](#)). Hieraus ergeben sich folgende Lösungsansätze (vgl. [Abschnitt 2.3.2](#)):

- Vergangenheits- und zukunftsbasierte Prognoseverfahren
- Analyse von Markt-, Kunden- und Produktdaten

Im Rahmen der Bedarfsplanung lässt sich der Bullwhip-Effekt ausgelöst durch einen langsamen Informationsfluss beobachten. Wie in [Abschnitt 2.3.2](#) beschrieben, beeinflusst der verzögerte Informationsaustausch die anderen Objektflüsse negativ, so dass ein Aufschaukeln der Nachfrage entlang der SC einsetzt. Der Bullwhip-Effekt ist keine direkte Aufgabenstellung, jedoch wirkt er sich negativ auf die gesamte SC aus. So beeinflusst eine fehlerhafte Bedarfsplanung die ihr untergeordneten Aufgabenbereiche (vgl. [Abschnitt 2.3.2](#)). Deshalb werden im Rahmen der Bedarfsplanung Faktoren zur Reduzierung des Bullwhip-Effektes betrachtet, so dass der Aufgabenbereich der Bedarfsplanung um folgende Lösungsansätze ergänzt wird (vgl. [Abschnitt 2.3.2](#)):

- Unternehmensübergreifender Datenaustausch
- Kollaborative Bedarfsplanung

Netzwerkplanung Nach der Bedarfsplanung erfolgt die Netzwerkplanung auf Basis von Bedarfsprognosen der Bedarfsplanung (vgl. [Abschnitt 2.3.2](#)). Aus [Abschnitt 2.3.2](#) ergeben sich die Modellierung der SC und die Kapazitätszuordnung. Im Gegensatz zur Modellierung der SC auf der Gestaltungsebene ist der Detaillierungsgrad auf der Planungsebene höher. So gilt es, Standorte und ihre Beziehungen durch weitere Informationen wie Zeit-, Kosten und Kapazitätsrestriktionen anzureichern (vgl. [Abschnitt 2.3.2](#)). Auf der Gestaltungsebene hingegen wird das Verständnis der Gesamtstruktur fokussiert (vgl. [Abschnitt 2.3.1](#)). Nach der SC-Modellierung werden die Produktionsvolumina kostenoptimal auf das Netzwerk verteilt. Die optimale Aufteilung der Produktionsvolumina auf die betrachteten Standorte erfolgt durch Netzwerkplanungsmodelle. Hieraus ergeben sich folgende Lösungsansätze (vgl. [Abschnitt 2.3.2](#)):

- Detaillierte SC-Modellierung
- Lineare Netzwerkplanungsmodelle

Beschaffungs- und Distributionsplanung Innerhalb der Beschaffungs- und Distributionsplanung erfolgt die Optimierung von Bestell- und Transportmengen sowie die Planung von Transportrouten (vgl. [Abschnitt 2.3.2](#)). Hierbei sollen die Kosten durch niedrige Bestände und optimierte Transportrouten und -rhythmen minimiert werden. Die Aufgabenstellungen der Beschaffungs- und Distributionsplanung unterscheiden sich hinsichtlich ihrer Zielgrößen (vgl. [Abschnitt 2.3.2](#)). Jedoch lassen sich Fragestellungen der Beschaffungs- und Distributionsplanung in die Bestands- oder Transportplanung einordnen, weshalb keine Aufteilung der Aufgabenbereiche vorgenommen wird.

Im Rahmen der Bestandsplanung werden mathematische Berechnungsverfahren zur Berechnung von Bestell-, Bestands-, und Liefermengen eingesetzt (vgl. [Abschnitt 2.3.2](#)). Die Transportplanung erfolgt durch die Simulation unterschiedlicher Liefergebiete und Transportrouten im bestehenden Netzwerk. Dazu wird die Optimierung der Transportflüsse hinsichtlich vordefinierter Zielkriterien wie Kosten oder Zeit bestimmt (vgl. [Abschnitt 2.3.2](#)). Folgende Lösungsansätze lassen sich aus [Abschnitt 2.3.2](#) herleiten:

- Mengen- und bestandsorientierte Berechnungsverfahren
- Simulationsverfahren zur Transportroutenbestimmung

Order Promising Der letzte Aufgabenbereich der Planungsebene ist das Order Promising. Hierbei werden Verfügbarkeits- oder Machbarkeitsprüfungen von Kundenanfragen oder -aufträgen durchgeführt (vgl. [Abschnitt 2.3.2](#)). Die Prüfungen erfolgen durch den Zugriff auf Produktions- und Bestandsdaten, um bspw. Kapazitäten oder Konfigurationen abzufragen und mögliche Liefertermine zu berechnen (vgl. [Abschnitt 2.3.2](#)). Hieraus ergeben sich folgende Lösungsansätze:

- Datenbankabfragen zur Verfügbarkeitsprüfung
- Datenbankabfragen zur Machbarkeitsprüfung

Die Taxonomie soll eine Einteilung logistischer Fragestellungen in Aufgabenstellungen und ihre -bereiche ermöglichen. Die Lösungsansätze dienen der Charakterisierung von Aufgabenstellungen, wobei neue Lösungsansätze im Rahmen einer Erweiterung der Taxonomie hinzugefügt werden können. So sollen die bestehenden Lösungsansätze als Ausgangsbasis zur Entwicklung neuer Lösungsansätze dienen. Durch den hierarchischen Aufbau der Taxonomie gelten die erarbeiteten Lösungsansätze der Aufgabenstellungen ebenfalls für die untergeordneten Fragestellungen.

4.1.2 Kategorisierung graphbasierter Verfahren

In der Literatur existiert keine Kategorisierung von graphbasierten Verfahren. Aufgrund der Heterogenität der Verfahren ist jedoch eine Kategorisierung zu verfolgen, um die Verfahren systematisch untersuchen zu können. Nachfolgend orientiert sich die Bildung von Kategorien nach [Aggarwal \(2010\)](#). Ergänzt werden die zwei Kategorien mit den graphbasierten Modellen aus [Abschnitt 3.4.1](#), so dass sich insgesamt drei Kategorien bilden. Anschließend werden die in [Abschnitt 3.4](#) beschriebenen Modelle und Algorithmen den entsprechenden Kategorien zugeordnet, wobei sich die Einordnung an den Ausführun-

gen der Literatur aus [Abschnitt 3.4.1](#), [Abschnitt 3.4.2](#) und [Abschnitt 3.4.3](#) orientiert. Zusätzlich werden Algorithmen der Graphenvisualisierung in die Kategorie der Graph-Mining-Algorithmen aufgenommen, da sie dem Graph-Mining zugeordnet werden (vgl. [Abschnitt 3.4.3](#)). Entscheidungsbäume, die auch im Data-Mining eingesetzt werden, verbleiben in der Kategorie der graphbasierten Modelle. Dies erweitert die potenziellen Lösungsansätze von Entscheidungsbäumen in [Abschnitt 4.2](#), da sich ihr Einsatz im Data-Mining meist auf Klassifizierungsaufgaben beschränkt (vgl. [Abschnitt 3.4.1](#)). Die restlichen Unterkategorien der graphbasierten Verfahren in [Abbildung 4.2](#) entsprechen der Einteilung aus [Abschnitt 3.4](#).

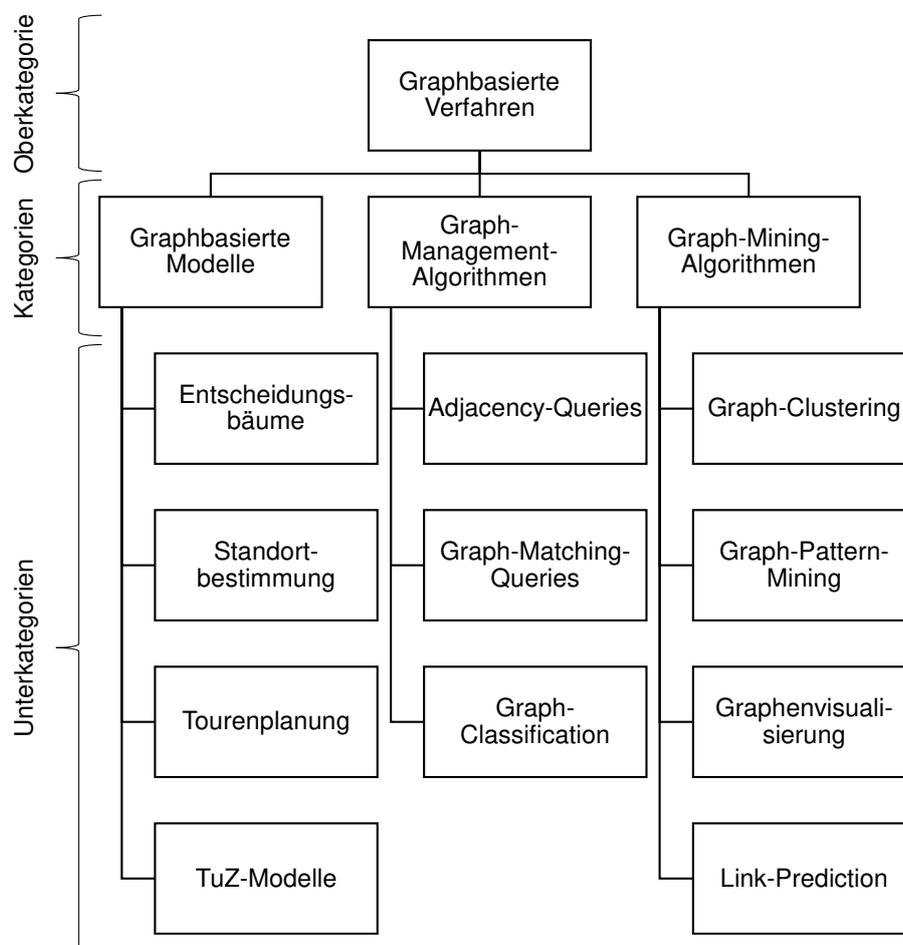


Abbildung 4.2: Kategorisierung graphbasierter Verfahren

Zur übersichtlichen Aufbereitung wird pro Kategorie eine Tabelle erstellt. Die Beschreibung der zu untersuchenden Unterkategorien erfolgt in [Tabelle 4.1](#), [Tabelle 4.14](#) und [Tabelle 4.3](#). Dies schafft eine Grundlage, um den Einsatz der Unterkategorien aus [Abbildung 4.2](#) in Aufgabenstellungen aus [Abschnitt 4.1.1](#) strukturiert aufzubereiten. So sollen auf Basis der Unterkategorien graphbasierte Lösungsansätze herausgearbeitet werden. Auch wird die Nachvollziehbarkeit der in [Abschnitt 4.2.1](#) und [Abschnitt 4.2.2](#) vergebenen Punkte durch die Beschreibung der Unterkategorien unterstützt. Die Tabellen basieren auf den Inhalten von [Abschnitt 3.4.1](#), [Abschnitt 3.4.2](#) und [Abschnitt 3.4.3](#). In der Beschreibung der Tabellen

werden die Funktionen der jeweiligen Unterkategorie zusammengefasst und durch Beispiele und veranschaulicht. [Tabelle 4.1](#) beschreibt graphbasierte Modelle im SCM. Die Modelle werden bereits im SCM eingesetzt, so dass die Beispiele und Einsatzfelder bereits Lösungsansätze auflisten, die in [Abschnitt 4.2.1](#) und [Abschnitt 4.2.2](#) hinsichtlich der untersuchten Aufgabenstellung übernommen werden können (vgl. [Abschnitt 3.4.1](#)). Die Graphalgorithmen aus [Tabelle 4.1](#) und [Tabelle 4.14](#) wurden noch nicht systematisch hinsichtlich ihrer Einsatzmöglichkeiten im SCM untersucht, so dass die Beispiele aus anderen Anwendungsfeldern stammen (vgl. [Abschnitt 3.4.2](#) und [Abschnitt 3.4.3](#)). In [Abschnitt 4.2.1](#) und [Abschnitt 4.2.2](#) müssen die Lösungsansätze deshalb erst aus der vorhandenen Beschreibung und den Beispiele hergeleitet werden.

Tabelle 4.1: Beschreibung graphbasierter Modelle des SCM

Unterkategorien	Beschreibung	Beispiele und Einsatzfelder
Entscheidungsbäume	<ul style="list-style-type: none"> • Darstellung und Vereinfachung sukzessiver Entscheidungssituationen • Unterteilung eines Gesamtproblems • Klassifizierung • Stochastische Optimierung 	<ul style="list-style-type: none"> • Planung von Lager- oder Fertigungsstandorten • Stochastische Optimierung • Modellierung komplexer Szenarien und ihrer Wahrscheinlichkeiten (Szenariobäume)
Standortbestimmung	<ul style="list-style-type: none"> • Bestimmung von Standorten unter Berücksichtigung von Restriktionen • Unterteilung der Modelle nach Fragestellungen (z. B. Center-Problem, Covering-Problem, Median-Problem, Warehouse-Location-Problem) 	<ul style="list-style-type: none"> • Standortbestimmung von Lagern, Distributionszentren, Servicezentren, Produktionswerken • Standortbestimmung bei kundenorientierten Dienstleistungen • Standortbestimmung bei Konsumgüter-Distribution • Planung von Regionallagern • Kapazitätsaufteilung
Tourenplanung	<ul style="list-style-type: none"> • Planung von optimalen Touren (z. B. Minimierung der Gesamttransportdistanz) • Planung von Reihenfolgen 	<ul style="list-style-type: none"> • Auslieferung von Gütern unter Kapazitäts- oder Zeitrestriktionen • Kommissionierung von Aufträgen • Optimierung von Rüstreihenfolgen • Briefverteilung • Mehrdepot-Auslieferungen
TuZ-Modelle	<ul style="list-style-type: none"> • Lösung von Transport- und Zuordnungsproblemen • Modellierung von Flüssen • Optimierung von Zuordnungen 	<ul style="list-style-type: none"> • Konfiguration der SC • Festlegung von Liefergebieten • Festlegung von Transportrouten • Mitarbeiterplanung

Tabelle 4.2: Beschreibung von Graph-Management-Algorithmen

Unterkategorien	Beschreibung	Beispiele und Einsatzfelder
Adjacency-Queries	<p>Nachbarschaftsabfragen in Graphdatenbanken:</p> <ul style="list-style-type: none"> • Einfache Nachbarschaftsabfragen • Komplexe Nachbarschaftsabfragen 	<ul style="list-style-type: none"> • Bestimmung von Nachbarn (Knoten) • Prüfung direkter Verbindungen (Kanten) • Bestimmung von Nachbarschaften (k-Neighborhood) • Suche nach Knoten mit bestimmter Distanz zum Wurzelknoten (k-hops) • Empfehlungssysteme, soziale Netzwerke (Vorschlag von Produkten, Freunden etc.)
Graph-Matching-Queries	<p>Musterabfragen in Graphdatenbanken:</p> <ul style="list-style-type: none"> • Mustererkennung (exakt und inexakt) • Semantische Mustererkennung • Kombination von Mustern 	<ul style="list-style-type: none"> • Prüfen der Isomorphie von Graphen • Auffinden maximaler Teilgraphen • Soziale Netzwerke (Erkennen von Beziehungsmustern) • Semantic Web (Erkennen semantischer Zusammenhänge)
Reachability-Queries	<p>Pfadabfragen in Graphdatenbanken:</p> <ul style="list-style-type: none"> • Abfrage der Erreichbarkeit • Prüfung spezieller Verbindungen • Identifizierung von Brücken • Berechnung minimal aufspannender Bäume • Berechnung kürzester Wege 	<ul style="list-style-type: none"> • Soziale Netzwerke (Suche nach gleichen Interessen) • Biologische Netzwerke (Identifikation spezieller biologischer Verbindungen) • Herleitung des Zusammenhaltes eines Netzwerks

Tabelle 4.3: Beschreibung von Graph-Mining-Algorithmen

Unterkategorien	Beschreibung	Beispiele und Einsatzfelder
Graph-Classification	<ul style="list-style-type: none"> • Beschreibung und Einordnung von Gruppen • Klassifizierung einzelner Knoten in vorhandene Gruppen (Label-Propagation) • Klassifizierung von Graphen in vorhandene Gruppen (Graph-Classification) 	<ul style="list-style-type: none"> • Soziale Netzwerke (Bildung von Zielgruppen für Werbung) • Klassifizierung von Bildern • Charakterisierung von Fingerabdrücken • Unterteilung und Untersuchung molekularer Strukturen
Graph-Clustering	<ul style="list-style-type: none"> • Gruppenbildung auf Basis einzelner Knoten (Node-Clustering) • Gruppenbildung auf Basis einzelner Graphen (Graph-Clustering) • Suche nach dem minimalen Schnitt und maximalen Flüssen • Graphpartitionierung • Komprimierung von Graphen • Analyse von Flüssen 	<ul style="list-style-type: none"> • Bestimmung dichter Regionen • Vereinfachung von Netzwerkstrukturen • Standortbestimmung • Kundenanalysen • Produktclusterbildung • Optimierungen in der Distribution (Bestand, Durchsatz)
Graph-Pattern-Mining	<ul style="list-style-type: none"> • Entdeckung unbekannter Muster (interessante, charakteristische oder regelmäßige Muster) • Zusammenführung von Graphen 	<ul style="list-style-type: none"> • Suche nach chemischen Strukturen • Identifizierung von Anomalien in IT-Sicherheitsnetzwerken • Suche nach zusammenhängenden Objekten • Optimierung der Kommissionierung • Vorbereitung für weitere Graph-Mining-Algorithmen
Graphvisualisierung	<ul style="list-style-type: none"> • Visualisierung eines Graphen (Knoten, Kanten, Eigenschaften) 	<ul style="list-style-type: none"> • Übersichtliche Darstellung von Netzwerken
Link-Prediction	<ul style="list-style-type: none"> • Identifizierung fehlender oder überflüssiger Kanten • Vorhersage zukünftiger Kanten 	<ul style="list-style-type: none"> • Soziale Netzwerke (Vorhersage von Freundschaften) • Wissenschaftliche Netzwerke (Vorhersage von Mitautoren-schaften)

4.2 Untersuchung und Bewertung graphbasierter Verfahren

Die zur Herleitung von Lösungsansätzen erstellte Bewertungsmatrix setzt sich aus den Aufgabenbereichen und -stellungen aus [Abschnitt 4.1.1](#) und den Ober- und Unterkategorien aus [Abschnitt 4.1.2](#) zusammen. Die Lösungsansätze ergeben sich durch die Untersuchung der Unterkategorien aus [Abschnitt 4.1.2](#) hinsichtlich der Aufgabenstellungen aus [Abschnitt 4.1.1](#). Hierbei werden die Unterkategorien beschrieben und durch Beispiele erläutert, um graphbasierte Lösungsansätze im SCM herzuleiten (vgl. [Abschnitt 4.1.2](#)). Lassen sich Aufgabenstellungen durch die Lösungsansätze beantworten, wird angenommen, dass untergeordnete Fragestellungen durch dieselben Lösungsansätze beantwortet werden können. Deshalb werden Fragestellungen erst in der praktischen Umsetzung in [Kapitel 5](#) untersucht.

Die in [Abschnitt 4.1.1](#) aufgelisteten Lösungsansätze dienen als Orientierungshilfe, um auf Basis dieser Ansätze neue Lösungsansätze abzuleiten. Dazu wird zwischen graphbasierten Lösungsansätzen, die den ursprünglichen Lösungsansätzen in ihrer Vorgehensweise gleichen und neuartigen bzw. nicht-validierten Vorgehensweise, unterschieden. Dies wirkt sich auf die Punktevergabe innerhalb der Bewertungsmatrix aus. Werden Lösungsansätze entdeckt, die nicht an bestehende Lösungsansätze im SCM anknüpfen, erhält der betrachtete Lösungsansatz einen Punkt. Dies verdeutlicht, dass zwar Lösungsansätze bestehen, diese aber noch validiert werden müssen. Ein Lösungsansatz wird mit zwei Punkten bewertet, wenn der Lösungsansatz bereits in der Literatur validiert wurde oder der Lösungsansatz eine vollständige Beantwortung der Aufgabenstellung, gemessen an den Lösungsansätzen aus [Abschnitt 4.1.1](#), verspricht. Keine entdeckten Lösungsansätze werden durch eine Null gekennzeichnet. Jedoch ist zu beachten, dass sich durch die Betrachtung anwendungsspezifischer Fragestellungen zusätzliche Lösungsansätze ergeben können, so dass eine Veränderung der vergebenen Punkte denkbar ist. Insgesamt ergibt sich eine Bewertungsskala von Null bis Zwei.

Die Identifikation der Lösungsansätze erfolgt zuerst abschnittsweise in [Abschnitt 4.2.1](#) sowie [Abschnitt 4.2.2](#) und werden in [Abschnitt 4.2.3](#) zusammengetragen. Die jeweiligen Aufgabenstellungen aus [Abschnitt 4.1.1](#) werden pro Zeile eingetragen. In den Spalten werden die Unterkategorien aus [Abschnitt 4.1.2](#) aufgelistet. Pro Aufgabenstellung werden alle Unterkategorien hinsichtlich ihrer potenziellen Lösungsansätze untersucht und bewertet.

Nachfolgend werden die Lösungsansätze auf Gestaltungsebene untersucht. Da die strategische Netzwerkgestaltung in [Abschnitt 4.1.1](#) in weitere Aufgabenbereiche unterteilt wurde und somit keine direkt zugeordneten Aufgabenstellungen enthält, wird sie als Aufgabenbereich nicht untersucht. Stattdessen werden die Aufgabenstellungen zur Erstellung von Planungsszenarien, zur Erstellung von Gestaltungsalternativen sowie zur Auswahl und Bewertung auf der Gestaltungsebene betrachtet (vgl. [Abschnitt 4.1.1](#)).

4.2.1 Lösungsansätze in der Gestaltungsebene

[Abschnitt 4.2.1](#) untersucht graphbasierte Lösungsansätze der Unterkategorien aus [Abschnitt 4.1.2](#) hinsichtlich der Aufgabenstellungen aus [Abschnitt 4.1.1](#) auf der Gestaltungsebene. Die Aufgabenbereiche der Gestaltungsebene teilen sich nach [Abschnitt 4.1.1](#) in die Erstellung von Planungsszenarien, die Erstellung von Gestaltungsalternativen sowie die Auswahl und Bewertung auf.

4.2.1.1 Erstellung von Planungsszenarien

Nachfolgend werden die Aufgabenstellungen des ersten Aufgabenbereichs, die Erstellung von Planungsszenarien untersucht. Die Aufgabenstellung gliedern sich in die Modellierung der SC und die Simulation der Unternehmensumwelt (vgl. [Abschnitt 4.1.1](#)). Die vergebenen Punkte werden in [Tabelle 4.4](#) dargestellt.

Modellierung der Supply Chain Zur Modellierung der SC müssen alle Logistikelemente und ihre entscheidungsrelevanten Merkmale erfasst werden. Die Logistikelemente setzen sich aus den SC-Partnern, SC-Standorten und Standortmerkmalen sowie ihren Beziehungen zueinander zusammen (vgl. [Abschnitt 2.2.2](#)). Die SC-Partner bilden Lieferanten, Hersteller, Händler, Distributoren, Dienstleister und Kunden. Produktionswerke, Lager und Distributionszentren sind Beispiele für Standorte (vgl. [Abschnitt 2.2.1](#)). Beziehungen zwischen den einzelnen Standorten der SC ergeben sich durch die Betrachtung der Objektflüsse (vgl. [Abschnitt 2.2](#)). Die Verbindung der Standorte führt zu einer netzwerkartigen Struktur des Verbundes, so dass SC in der Literatur häufig auf Basis graphbasierter Ansätze der Netzwerktheorie erforscht werden (vgl. [Abschnitt 3.4.1](#)). Hierbei werden die Standorte und ihre Akteure als Knoten und ihre Beziehungen als Kanten erfasst (vgl. [Abschnitt 3.4.1](#)).

In Graphdatenbanken werden Daten in einer Graphenrepräsentation gespeichert und durch Visualisierungsalgorithmen in einer für den Anwender visuell verständlichen Darstellung aufbereitet (vgl. [Abschnitt 3.4.3](#)). Durch die Graphenvisualisierung kann eine SC vollständig modelliert werden. So lassen sich Knoten und Kanten sowie ihre Eigenschaften darstellen, so dass die komplette SC abgebildet werden kann (vgl. [Abschnitt 3.3.2](#)). Somit wird die Graphenvisualisierung mit zwei Punkten bewertet.

Die Komplexität von SC wächst seit Beginn des 21. Jahrhunderts stetig an und feste Strukturen der SC beginnen sich aufzulösen, so kann ein Unternehmen bspw. in mehreren SC tätig sein (vgl. [Abschnitt 2.1.1](#)). Als Folge fangen SC an sich zu überlagern (vgl. [Abschnitt 2.1.1](#)). Dies erfordert neue Ansätze der SC-Darstellung und -Strukturierung, damit die hieraus resultierenden kleinen und flexiblen Netzwerke erfasst und koordiniert werden können (vgl. [Abschnitt 2.1.1](#)). Aufgrund der steigenden Komplexität und der steigenden Anzahl an Akteuren ist eine reine Visualisierung der SC für den Anwender nicht mehr ausreichend. Vielmehr benötigt er zusätzlich Werkzeuge, um ein erstes Verständnis für die dargestellte SC zu erlangen (vgl. [Abschnitt 3.1](#)).

Durch das Graph-Clustering lässt sich die SC komprimieren und zusammenfassen, so dass die Gesamtstruktur der SC problemlos erfasst werden kann (vgl. [Tabelle 4.3](#)). Dies ermöglicht die Zusammenfassung von Unternehmen oder Standorten, die sich auf einer Wertschöpfungsstufe befinden, um die Übersichtlichkeit eines Netzwerkes zu erhöhen (vgl. [Tabelle 4.3](#)). Weiterhin könnten durch das Graph-Clustering kleine und flexible SC innerhalb einer großen SC erfasst werden. Durch Analyse der Kantendichte können z. B. besonders stark kooperierende Unternehmen identifiziert werden, die im Betrachtungsmoment eine SC in Form eines Teilgraphen bilden (vgl. [Tabelle 4.3](#)). Dies verhindert die Darstellung eines großen, unübersichtlichen Graphen und ermöglicht den Fokus auf den für den Interessenverbund relevanten Teil der SC. Graph-Clustering-Algorithmen sind jedoch nur ein unterstützendes Werkzeug zur Modellierung der SC. Auch handelt es sich um einen nicht-validierten Lösungsansatz, so dass Graph-Clustering einen Punkt erhält.

Graph-Classification-Algorithmen werden u. a. zur Einteilung einzelner Knoten in Gruppen eingesetzt (vgl. [Tabelle 4.3](#)). So können verschiedene Akteur- oder Standortknoten abhängig von ihrer Klasse und in Verbindung mit Graphenvisualisierung eingefärbt werden. Durch die visuelle Unterstützung erhöht sich die Übersichtlichkeit des dargestellten Graphen für den Anwender. So lassen sich Lieferanten direkt identifizieren oder der Verlauf der einzelnen Wertschöpfungsstufen visuell nachvollziehen. Auch die Einfärbung neu hinzugefügter oder aktuell nicht verfügbarer Teilnehmer könnte durch Label-Propagation realisiert werden (vgl. [Tabelle 4.3](#)). Des Weiteren lassen sich durch die Graph-Classification-Algorithmen Klassenstrukturen analysieren, so dass das Verständnis über die SC, ihre Akteure und ihren Beziehungen vertieft wird (vgl. [Abschnitt 3.4.3](#)). Auch hierbei handelt es sich um eine Erweiterung der Modellierung der SC und einen nicht-validierten Lösungsansatz, so dass Graph-Classification-Algorithmen mit einem Punkt bewertet werden.

Graph-Matching-Queries werden zur Musterabfrage und Filterung eingesetzt (vgl. [Abschnitt 3.4.2](#)). Die Modellierung der SC kann mit einer Vielzahl an Teilnehmern und Verbindungen einhergehen, so dass die Übersichtlichkeit der SC verringert wird (vgl. [Abschnitt 2.1.1](#)). Durch den Einsatz von Graph-Matching-Queries können bestimmte Teilnehmer und Verbindungen gefiltert werden, so dass in Verbindung mit der Graphvisualisierung nur ein Teil der SC abgebildet wird. Auch Graph-Matching-Queries werden mit einem Punkt bewertet.

Simulation der Unternehmensumwelt Die zweite Aufgabenstellung im Rahmen der Erstellung von Planungsszenarien ist die Simulation der Unternehmensumwelt. Hierfür müssen Szenarien der Unternehmensumwelt in Form von Planungsszenarien gebildet und in einer Szenarioanalyse ausgewertet werden (vgl. [Abschnitt 2.3.1](#)).

Gegenstände, Individuen, abstrakte Begriffe oder Ereignisse der realen Welt lassen sich durch Entitäten ausdrücken (vgl. [Abschnitt 2.2](#)). Auf Graphen übertragen können Knoten Institutionen, Konkurrenten oder Umweltereignisse ausgedrückt werden. Die Einflüsse der Knoten auf die SC lassen sich durch Kanten und ihren Eigenschaften darstellen. So könnten bspw. Rechteflüsse in Form von Kanten modelliert werden und durch Kantengewichte in Form von Restriktionen auf den Waren- oder Finanzfluss eines SC-Knoten wirken. So lassen sich Graphen modellieren, die die Unternehmensumwelt abbilden und sich nur durch Parametervariationen unterscheiden (vgl. [Abschnitt 2.3.1](#)). Dies lässt sich durch die Graphenvisualisierung realisieren, die auch zur Modellierung der SC verwendet wird (vgl. [Abschnitt 4.2.1](#)). Im Gegensatz zur Modellierung der SC reicht eine reine Abbildung der Unternehmensumwelt nicht aus, da auch Planungsszenarien gebildet werden müssen (vgl. [Abschnitt 4.1.1](#)). Deshalb wird die Graphenvisualisierung mit einem Punkt bewertet.

Auf der Gestaltungsebene werden monetäre Zielgrößen betrachtet, so dass auf eine Optimierung des Finanzflusses abgezielt wird (vgl. [Abschnitt 2.3.1](#)). Will eine SC günstige Finanzierungsmittel beziehen, lassen sich die Finanzflüsse zwischen Kreditinstituten, Ländern und der SC als Kanten darstellen. Die Gewichte der Kanten können in Form von Steuern ausgedrückt werden, so dass eine Auswertung nach dem kleinsten Kantengewicht bzw. günstigsten Finanzierungsweg ermöglicht wird. Auch Import- oder Umweltrestriktionen durch Erhöhung der Kantengewichte von Transportwegen sind denkbar. Die hierbei ermittelten Szenarien könnten als Hyperknoten in einen Szenarienbaum eingetragen und nach ihren Eintrittswahrscheinlichkeiten angeordnet werden. Der Szenariobaum ähnelt dem Lösungsansatz der Szenarioanalyse aus [Abschnitt 4.1.1](#) zur Simulation der Unternehmensumwelt. Jedoch bieten Szenariobäume keine Möglichkeit zur Generierung der zu

Tabelle 4.4: Bewertung der Lösungsansätze zur Erstellung von Planungsszenarien

	Entscheidungsbäume	Standortbestimmung	Transportplanung	TuZ-Modelle	AdjacencyQueries	Graph-Matching-Queries	Reachability-Queries	Graph-Classification	Graph-Clustering	GPM	Graphvisualisierung	Link-Prediction
Modellierung der Supply Chain	0	0	0	0	0	1	0	1	1	0	2	0
Simulation der Unternehmensumwelt	1	0	0	0	0	0	0	0	1	0	1	0

variierenden Parameter (vgl. [Abschnitt 3.4.1](#)). Somit werden Szenariobäume mit einem Punkt bewertet.

Graph-Clustering-Algorithmen erleichtern das Verständnis des Anwenders über die Unternehmensumwelt durch Zusammenfassung der relevanten Umweltaspekte (vgl. [Tabelle 4.3](#)). Die Unterkategorie wird mit einem Punkt bewertet. Ein Verständnis der inneren Struktur von Akteuren der Unternehmensumwelt wird in [Abschnitt 2.3.1](#) nicht thematisiert, da ihre Wirkung auf die SC fokussiert wird. Deshalb werden Graph-Classification-Algorithmen im Gegensatz zur SC-Modellierung nicht in die Lösungsmenge aufgenommen.

4.2.1.2 Erstellung von Gestaltungsalternativen

Der folgende Abschnitt untersucht die Unterkategorien aus [Abschnitt 4.1.2](#) hinsichtlich der Aufgabenstellungen der Erstellung von Gestaltungsalternativen (vgl. [Abschnitt 4.1.1](#)). Hierbei werden die geografische Standortbestimmung und die Netzwerkkonfiguration untersucht. Die Punktevergabe wird in [Tabelle 4.4](#) dargestellt.

Geografische Standortbestimmung In der geografischen Standortbestimmung werden optimale Standorte hinsichtlich vordefinierter Zielgrößen oder Restriktionen durch Standortmodelle bestimmt (vgl. [Abschnitt 2.3.1](#) und [Abschnitt 3.4.1](#)). Hierbei wird die Auswahl von Distributionslagern oder Servicezentren primär betrachtet, jedoch eignen sich Standortmodelle auch für die Wahl von anderen Standorttypen einer SC (vgl. [Abschnitt 3.4.1](#)). Die verschiedenen Modelle zur Standortbestimmung fokussieren unterschiedliche Zielsetzungen, so dass unterschiedliche Fragestellungen beantwortet werden können (vgl. [Tabelle 4.1](#)). Auch auf Produktionswerke lassen sich die Standortmodelle übertragen, wobei anstatt der Nähe zu Kunden die Nähe zu Lieferanten fokussiert wird (vgl. [Abschnitt 3.4.1](#)). Da Modelle der Standortbestimmung durch die Literatur validiert sind, erhalten sie zwei Punkte.

Neben den Modellen der Standortbestimmung können Adjacency-Queries zur Standortbestimmung eingesetzt werden. So ähneln die k-Neighborhood-Queries den Vorgehensweisen des beschriebenen Median- bzw. p-Median-Problems aus [Abschnitt 3.4.1](#). Die k-Neighborhood-Queries identifizieren alle Knoten innerhalb einer bestimmten Kantenlänge (vgl.

[Abschnitt 3.4.3](#)). Das Median-Problem bestimmt einen Lagerstandort, dessen gewichtete Entfernung zu den Kunden minimal ist (vgl. [Abschnitt 3.4.1](#)). Es ist zu beachten, dass die k-Neighborhood-Queries eine entgegengesetzte Sichtweise verfolgen und die Berechnung der unbekanntenen Nachbarschaftsknoten aus einem bereits vorhandenen Wurzelknoten erfolgt (vgl. [Abschnitt 3.4.3](#)). Jedoch ist eine Modifizierung des Algorithmus denkbar. So könnte der Wurzelknoten unbekannt sein und die Nachbarschaft gegeben. Durch das Ausführen des k-Neighborhood-Queries auf die einzelnen Knoten der Nachbarschaft ließe sich eine Schnittmenge an Knoten bestimmen, die von jedem Nachbarschaftsknoten innerhalb einer bestimmten Weglänge realisierbar wäre. Durch sukzessive Verkleinerung der k-Weglängen könnte die Schnittmenge auf einen Wurzelknoten reduziert werden. Da diese Lösungsansätze im Gegensatz zu den Lösungsansätzen der Modelle der Standortbestimmung noch nicht validiert sind, erhalten Adjacency-Queries einen Punkt.

Durch Graph-Matching-Queries lassen sich Muster mit vorgegebenen und teilweise unbekanntenen Attributeigenschaften abfragen (vgl. [Abschnitt 3.4.3](#)). So kann ein Standort abgefragt werden, der eine minimale Entfernung zu den anderen Kundenstandorten aufweist. Da Graph-Matching-Queries in der Literatur noch nicht zur Standortbestimmung eingesetzt werden, erhält der Algorithmus einen Punkt. Zu berücksichtigen ist, dass beide Graph-Management-Algorithmen eine begrenzte Komplexität des betrachteten Netzwerkes voraussetzen, da der Anwender die Abfrage aktiv durchführt (vgl. [Abschnitt 3.4.3](#)).

Es ist zu beachten, dass quantitative Modelle der Standortbestimmung auf einer Vorauswahl von Standorten basieren. So können Standorte regional, landesweit und weltweit auf Basis einer Vielzahl unterschiedlicher Standortfaktoren ausgewählt werden (vgl. [Abschnitt 3.4.1](#)). Quantitative Standortmodelle unterstützen den Anwender bei der Vorauswahl von Standorten oder Regionen nicht (vgl. [Abschnitt 3.4.1](#)). Graph-Clustering-Algorithmen werden zur Identifizierung dichter Regionen (vgl. [Abschnitt 3.4.3](#)). Zur Auswahl potenzieller Standorte oder Regionen könnten Regionaldaten, Bevölkerungsdaten oder Daten durch die Algorithmen ausgewertet werden. So könnten Graph-Clustering-Algorithmen interessante Regionen bestimmen, die bspw. im hohen Maß von der gewünschten Zielgruppe bevölkert werden. Graph-Clustering-Algorithmen werden jedoch mit einem Punkt bewertet, da sie hinsichtlich der Standortbestimmung noch nicht untersucht wurden und keinen der bisherigen Lösungsansätzen ähneln.

Netzwerkconfiguration Die Netzwerkconfiguration dient zur kostengünstigen Auslegung des Materialflusses, z. B. kürzeste Transportrouten, zwischen den Partnern einer SC. Auf Basis der zuvor modellierten SC werden optimale Verbindungen identifiziert und vorhandene Verbindungen auf ihre Notwendigkeit geprüft (vgl. [Abschnitt 2.3.1](#)).

TuZ-Modellen werden bereits zur Optimierung von SC eingesetzt (vgl. [Abschnitt 3.4.1](#)). Insbesondere Transportmodelle werden, wie in [Abschnitt 3.4.1](#) beschrieben, zur Modellierung und Optimierung von Materialflüssen eingesetzt. Hierfür werden Transportrouten definiert, um die Gesamtkosten des Transports durch Optimierung der transportierten Mengeneinheit, der betrachteten Fix- und Transportkosten im Netzwerk zu minimieren. Auch können durch Transportmodelle mehrere Lagerstufen in Form von Zentral- oder Zwischenlager erfasst werden (vgl. [Abschnitt 3.4.1](#)). Deshalb erhalten TuZ-Modelle zwei Punkte.

Ebenfalls mit zwei Punkten werden Link-Prediction-Algorithmen bewertet. Link-Prediction-Algorithmen dienen zur Analyse von Netzwerken, um fehlende oder zukünftige Kanten zwischen Knoten zu entdecken und vorherzusagen (vgl. [Abschnitt 3.4.3](#)). Hierbei erfolgt

die Bestimmung fehlender Kanten ähnlich zu TuZ-Modellen. So werden Kanten auf Basis minimaler Gewichte wie geringer geografischer Distanz oder geringen Kosten identifiziert. Auch werden vorhandene Kanten auf ihre Notwendigkeit überprüft, so dass verzichtbare Kanten entdeckt und entfernt werden können (vgl. [Abschnitt 3.4.3](#)). Im Gegensatz zu TuZ-Modellen werden nicht nur bereits existierende, sondern auch nicht-existente Kanten in der Netzwerkoptimierung berücksichtigt (vgl. [Abschnitt 3.4.3](#)). Dies bietet Optimierungspotenzial zur Beantwortung der Fragestellung, da sich somit mehr Konfigurationsmöglichkeiten ergeben.

Neben TuZ-Modellen und Link-Prediction können Reachability-Queries zur Optimierung einzelner Routen eingesetzt werden. Reachability-Queries fassen Pfadabfragen zusammen und werden bei Pfad- und Traversierungsproblemen eingesetzt (vgl. [Abschnitt 3.4.2](#)). Erfolgt die Netzwerkkonfiguration aus der Sicht eines Standortes, können durch Reachability-Queries kürzeste Wege zu den SC-Partnern und Partner mit ähnlichen Routen identifiziert werden. So könnten auf Basis ähnlicher Partner Konsolidierungsmaßnahmen zur Einsparung von Kosten untersucht werden. Auch lassen sich durch Reachability-Queries Brücken identifizieren, die übertragen auf eine SC mögliche Engpassstellen darstellen (vgl. [Tabelle 4.2](#)). So könnten frühzeitig Maßnahmenpläne bei Störungen der Engpassstellen ergriffen werden, um eine Unterbrechung der Materialflüsse zu verhindern. Reachability-Queries werden mit einem Punkt bewertet, da die beschriebenen Lösungswege nicht-validierte Lösungsansätze darstellen.

Auch Graph-Clustering-Algorithmen können zur Identifizierung möglicher Konsolidierungsmaßnahmen und Brücken eingesetzt werden (vgl. [Tabelle 4.3](#)). Durch Node-Clustering-Algorithmen lassen sich einzelne Partner der SC in Gruppen hinsichtlich bestimmter Strukturen wie räumlicher Nähe oder gleicher Zulieferer einteilen (vgl. [Abschnitt 3.4.3](#)). So könnten elementare Verbindungen zwischen den Gruppen identifiziert und Konsolidierungsmaßnahmen zur Kosteneinsparung ausgearbeitet werden. Auch mögliche Engpassstellen lassen sich durch die Verbindungen zu den anderen Gruppen identifizieren (vgl. [Tabelle 4.3](#)). In [Abschnitt 3.4.3](#) werden weitere Lösungsansätze von Graph-Clustering-Algorithmen zur Implementierung in der Netzwerkoptimierung durch [Irfan et al. \(2007\)](#) beschrieben, um u. a. Distributionswege zu optimieren. Graph-Clustering-Algorithmen werden mit zwei Punkten bewertet, da ihre Lösungsansätze in der Netzwerkkonfiguration durch die Literatur validiert sind.

Tabelle 4.5: Bewertung der Lösungsansätze zur Erstellung von Gestaltungsalternativen

	Entscheidungsbäume	Standortbestimmung	Transportplanung	TuZ-Modelle	AdjacencyQueries	Graph-Matching-Queries	Reachability-Queries	Graph-Classification	Graph-Clustering	GPM	Graphvisualisierung	Link-Prediction
Geografische Standortbestimmung	0	2	0	0	1	1	0	0	1	0	0	0
Netzwerkkonfiguration	0	0	0	2	0	0	1	0	2	0	0	2

4.2.1.3 Auswahl und Bewertung

Zuletzt werden auf Gestaltungsebene die Aufgabenstellungen des Aufgabenbereichs „Auswahl und Bewertung“ hinsichtlich der Unterkategorien aus [Abschnitt 4.1.2](#) untersucht. Die Aufgabenstellungen bestehen aus der Auswahl von Gestaltungsalternativen, der Auswahl der SC-Partner sowie der Bestimmung von Standortrollen (vgl. [Abschnitt 4.1.1](#)). Die Resultate der Bewertung werden in [Tabelle 4.6](#) zusammengefasst.

Auswahl von Gestaltungsalternativen Nachdem Gestaltungsalternativen auf Basis neu hinzugefügter Standorte und variierender Netzwerkkonfigurationen erstellt wurden, erfolgt die Bewertung der Alternativen unter Berücksichtigung der Planungsumgebungen (vgl. [Abschnitt 4.1.1](#)). Die Planungsumgebungen stellen hierbei die zuvor gebildeten Umweltszenarien dar. Um die Gestaltungsalternativen unter Berücksichtigung unterschiedlicher Planungsumgebungen zu bewerten, müssen beide Szenarien gemeinsam betrachtet werden (vgl. [Abschnitt 2.3.1](#)). Durch die unterschiedlichen Gestaltungsalternativen und Planungsumgebungen resultieren mehrere Graphenkombinationen, die in ihrer Gesamtheit bewertet werden müssen. Da die Komplexität von SC stetig wächst (vgl. [Abschnitt 2.1.1](#)), wird die Annahme getroffen, dass das Anwenderwissen für eine Auswertung der Gestaltungsalternativen auf Basis von Graph-Management-Algorithmen nicht ausreichend ist. Deshalb werden nachfolgend nur Graph-Mining-Algorithmen betrachtet.

Durch GPM-Algorithmen lassen sich zwei Graphen zusammenführen, wenn beide Graphen mindestens einen übereinstimmenden Knoten aufweisen (vgl. [Abschnitt 3.4.3](#)). Um dies zu berücksichtigen, müssen während der Simulation der Unternehmensumwelt alle Knoten der SC modelliert werden, die eine Verbindung zu Knoten aus der Unternehmensumwelt aufweisen. So entstehen gemeinsame Knoten, die eine Verknüpfung der beiden Graphen ermöglichen. Auch könnten die Restriktionen der Unternehmensumwelt über die Verbindungsknoten an das SC-Netzwerk weitergegeben werden und auf die Flüsse der SC wirken. GPM-Algorithmen werden mit einem Punkt bewertet, da sie die Bewertungsgrundlage für nachfolgende Graphalgorithmen schaffen.

Eine Gestaltungsalternative wird in mehrere Planungsumgebungen eingesetzt und hinsichtlich zuvor definierter Zielgrößen, bspw. optimaler Finanzmittel- oder Materialflüsse, bewertet (vgl. [Abschnitt 2.3.1](#)). In der Darstellung können die Gestaltungsalternativen und ihre Unternehmensumwelt durch Graph-Clustering-Algorithmen vereinfacht werden (vgl. [Tabelle 4.3](#)). Auch lassen sich durch Graph-Clustering-Algorithmen minimale Schnitte und maximale Flüsse bestimmen (vgl. [Abschnitt 3.4.3](#)). Minimale Schnitte unterteilen Graphen in zwei Cluster, wobei zuerst die minimale Summe der Kantengewichte zwischen zwei Gruppen gesucht wird (vgl. [Abschnitt 3.4.3](#)). Maximale Flüsse bestimmen die maximale Menge, die von einer Quelle über verschiedene Knoten bis hin zu einer Senke fließen können (vgl. [Abschnitt 3.4.3](#)). So lassen sich durch minimale Schnitte Kanten zwischen Unternehmensumwelt und SC identifizieren, die eine minimale Restriktion, z. B. niedrige Steuern, der Finanzflüsse und somit minimale Kosten aufweisen. Maximale Flüsse können Kanten innerhalb einer SC bestimmen, die maximale Mengen zwischen einer Quelle und einer Senke transportieren können. Dies ist u. a. zur Optimierung von Materialflüssen notwendig. Zu beachten ist, dass unterschiedliche Kanten und Attribute zwischen zwei gleichen Knoten betrachtet werden. So werden Finanzflüsse hinsichtlich minimaler Flüsse und Gewichte betrachtet, wohingegen Materialflüsse und ihre Kapazitäten im Rahmen von maximalen Flüssen fokussiert werden. Graph-Clustering-Algorithmen werden zur

Berechnung von Zielgrößen und zur vereinfachten Darstellung des betrachteten Graphen eingesetzt, ohne diese zu bewerten, und erhalten deshalb einen Punkt.

Zur Bewertung der Gestaltungsalternativen auf Grundlage der zuvor berechneten Zielgrößen können Graph-Classification-Algorithmen eingesetzt werden. Durch Graph-Classification lassen sich Teilgraphen als Ganzes innerhalb eines Graphen oder einer Graphenmenge klassifizieren (vgl. [Abschnitt 3.4.3](#)). Durch die Restriktion, einen Teilgraphen und keine einzelnen Knoten zu bewerten, wird die gesamte Gestaltungsalternative innerhalb der Planungsumgebung bewertet (vgl. [Tabelle 4.3](#)). So könnten Gestaltungsalternativen hinsichtlich derselben Planungsumgebung auf Basis der zuvor berechneten Zielgrößen bewertet und in Klassen von „sehr gut“ bis „ungeeignet“ eingeteilt werden. Anschließend könnte ein Gesamtvergleich mithilfe eines Entscheidungsbaums erfolgen, um die Gestaltungsalternative zu bestimmen, die unter Berücksichtigung der Planungsumgebungen und ihrer Eintrittswahrscheinlichkeit sowie der Zielgrößen den höchsten Erwartungswert aufweist (vgl. [Abschnitt 3.4.1](#)). Graph-Classification-Algorithmen und Entscheidungs bäume werden mit jeweils einem Punkt bewertet, da beide Lösungsansätze keine Validierung durch die Literatur aufweisen.

Auswahl von Supply-Chain-Partnern Die Auswahl von SC-Partnern erfolgt durch Checklisten und Nutzwertanalysen auf Basis quantitativer und qualitativer Merkmale, z. B. Vertrauenswürdigkeit und Kooperationskultur: Meist begrenzt sich die Auswahl auf Unternehmen, die in der Vergangenheit bereits mit der SC zusammengearbeitet haben (vgl. [Abschnitt 2.3.1](#)).

Graph-Classification-Algorithmen dienen zur Beschreibung und Einordnung einzelner Knoten oder Graphen in Gruppen (vgl. [Tabelle 4.3](#)). Wie bei Checklisten kann das Vorhandensein notwendiger Eigenschaften, bspw. Lieferzeit, Produkttyp oder -qualität, der SC-Partnern untersucht werden. So können SC-Partner anhand ihrer Merkmale klassifiziert und unterschiedlichen Gruppen zugeordnet werden, die sich in verschiedene Eignungsstufen einteilen. Jedoch ist zu beachten, dass Graph-Classification-Algorithmen neben quantitativen Faktoren auch qualitative Faktoren, z. B. die Produktqualität, zur Klassifizierung von SC-Partnern erfassen müssen (vgl. [Abschnitt 2.3.1](#)).

GPM-Algorithmen zielen auf die Entdeckung verborgener und für den Anwender interessanter Muster ab (vgl. [Abschnitt 3.4.3](#)). Im Rahmen der Auswahl von SC-Partnern sind Verhaltensmuster interessant, die auf die Güte qualitativer Merkmale wie Vertrauenswürdigkeit oder eine offene Kooperationskultur deuten (vgl. [Abschnitt 2.3.1](#)). Mit der Annahme, dass durch vergangene Geschäftsbeziehungen eine entsprechende Datengrundlage existiert, können GPM-Algorithmen zur Suche nach Verhaltens- oder Qualitätsmustern eingesetzt werden (vgl. [Tabelle 4.3](#)). Durch die Analyse der Anzahl an nachteiligen Vertragsänderungen, z. B. in Form von Preiserhöhungen, oder Vertragsbrüchen, z. B. nicht eingehaltener Liefertermine, könnte ein potenzieller SC-Partner hinsichtlich seiner Vertrauenswürdigkeit bewertet werden. Das Auffinden von auffallend hohen Wartungs- oder Reparationsarbeiten resultierend aus früheren Kooperationen könnte Aufschluss auf die Produktqualität der betrachteten SC-Partner geben. Sowohl Graph-Classification- als auch GPM-Algorithmen werden mit einem Punkt bewertet, da sie erst in Kombination aller Lösungsansätze aus [Abschnitt 4.1.1](#) zu einer Auswahl und Bewertung der SC-Partner führen.

In sozialen Netzwerken werden Link-Prediction-Algorithmen zur Vorhersage von Freundschaften oder Mitautorenschaften verwendet. Die Vorhersage basiert auf verschiedenen

Faktoren wie gemeinsamen Interessen, räumlicher Nähe oder gemeinsamen Bekannten (vgl. [Abschnitt 3.4.3](#)). Ähnlich zu der Bestimmung von Mitautorenschaften können Link-Prediction-Algorithmen auf Basis relevanter Faktoren mögliche SC-Partnerschaften vorher-sagen. So könnte eine mögliche Verbindung zu einem Zulieferer, der bereits mit Unter-nehmen der gleichen Unternehmensart wie das suchende Unternehmen zusammenarbeitet, entdeckt werden. Link-Prediction-Algorithmen werden mit einem Punkt bewertet, da sie einen nicht-validierten Lösungsansatz darstellen.

Neben Link-Prediction-Algorithmen eignen sich Adjacency-Queries zum Vorschlag potenzieller SC-Partner. Sie werden in Empfehlungssystemen eingesetzt, um ähnliche Produkte oder gemeinsame Freunde vorzuschlagen (vgl. [Abschnitt 3.4.3](#)). Auf SC-Partnerschaften bezogen, könnten neue Partner vorgeschlagen werden, die auf Basis vergangener Koope-rationen und Bewertungen vorgeschlagen werden. Auch Adjacency-Queries werden mit einem Punkt bewertet.

Ist das Vorwissen des Anwenders hoch genug oder die Anzahl an zu bewertenden SC-Partner überschaubar, um Entscheidungen ohne Graph-Mining-Unterstützung treffen zu können, eignen sich Entscheidungsbäume, Graph-Matching- und Reachability-Queries zur Auswahl von SC-Partnern. Der Auswahlprozess ähnelt der zuvor beschriebenen Vorge-hensweise auf Basis von Graph-Mining-Algorithmen mit dem Unterschied, dass der An-wender den Prozess aktiv durchführt und die Klassifizierung auf Basis seines vorhandenen Wissens erfolgt (vgl. [Abschnitt 3.4](#)). Entscheidungsbäume werden u. a. bei sukzessiven Entscheidungssituationen und zur Klassifizierung verwendet (vgl. [Tabelle 4.1](#)). So könnte der Anwender ähnlich zu den Graph-Classification-Algorithmen SC-Partner klassifizieren. Zur Unterstützung seiner Entscheidungen könnte er Graph-Matching-Queries zur Suche nach für ihn interessanten Mustern anwenden. Im Gegensatz zu GPM-Algorithmen be-nötigt der Anwender jedoch Wissen über die von ihm gesuchten Muster, so dass er die Suchmuster selbstständig abfragen muss (vgl. [Tabelle 4.2](#)). Ist der Anwender auf der Su-che nach einem Partner in räumlicher Nähe, können Reachability-Queries in Form von Pfadabfragen eingesetzt werden, um Partner mit den kürzesten Wegen zu identifizieren (vgl. [Tabelle 4.2](#)). Entscheidungsbäume, Graph-Matching- und Reachability-Queries wer-den mit einem Punkt bewertet.

Bestimmung von Standortrollen Standortrollen legen fest, welchen Zweck ein Werk in einem Produktionsnetzwerk erfüllen soll (vgl. [Abschnitt 2.3.1](#)). So muss der Anwender zwischen Produktvielfalt oder -fokus, produktbezogener oder funktionaler Spezialisierung oder verschiedenen Flexibilitätsgraden wählen. Es werden abgestimmte Kernkompetenzen fokussiert, so dass sich die Werke optimal ergänzen. Die Einteilung der Werke erfolgt auf Basis von Rollenmodellen anhand bestimmter Kriterien wie technologischem Know-How (vgl. [Abschnitt 2.3.1](#)).

Da Kriterien zur Einteilung von Standortrollen vielmehr qualitative Merkmale, bspw. tech-nologisches Know-How, als quantifizierbare Werte darstellen, muss das Wissen durch Ana-lyse der Zusammenhänge aus den Datensätzen extrahiert werden (vgl. [Abschnitt 2.3.1](#)). Hierfür eignen sich GPM-Algorithmen, die in der Literatur als vorgeschalteter Graph-Mining-Prozess zu Graph-Clustering- oder Graph-Classification-Algorithmen eingesetzt werden (vgl. [Abschnitt 3.4.3](#)). So könnten für die Bestimmung von Standortrollen alle relevanten Muster bestimmt werden, um eine Klassifizierung der betrachteten Werke vorzunehmen. So könnten Kostenstrukturen, Mitarbeiter- oder Produktstrukturen ana-lysiert werden, um hieraus die gesuchten Kriterien zur Bestimmung von Standortrollen

herzuleiten. GPM-Algorithmen werden mit einem Punkt bewertet, da sie zusammen mit Graph-Classification zu einer vollständigen Lösung der Aufgabenstellung führen könnten.

Standortrollen und ihre Kriterien zur Klassifizierung werden durch Rollenmodelle in der Literatur vorgegeben (vgl. [Abschnitt 2.3.1](#)). Deshalb werden zur Bestimmung von Standortrollen Graph-Classification-Algorithmen und keine Graph-Clustering-Algorithmen zur Eingruppierung verwendet. Im Gegensatz zu Graph-Clustering-Algorithmen, die Gruppen auf Basis von Ähnlichkeiten bilden, erfolgt die Einteilung von Knoten oder Graphen durch Graph-Classification-Algorithmen direkt in vorhandene Gruppen und nicht auf Basis von Ähnlichkeiten (vgl. [Abschnitt 3.4.3](#)). Durch die Algorithmen lassen sich vorhandene Standorte in vordefinierte Standortrollen einteilen. Graph-Classification-Algorithmen werden mit einem Punkt bewertet.

Besitzt der Anwender das für die Standortrollenbestimmung notwendige Wissen, kann die Klassifizierung ohne Graph-Mining-Algorithmen durchgeführt werden (vgl. [Abschnitt 3.4.3](#)). Entscheidungsbäume eignen sich zur Klassifizierung von Werken (vgl. [Tabelle 4.1](#)). Die Entscheidung über die Einteilung der Standorte in verschiedene Standortrollen kann durch Graph-Matching- und Reachability-Queries in Form von Muster- und Pfadabfragen unterstützt werden. Ist dem Anwender bekannt, wie sich die betrachteten Kriterien messen lassen, kann er mithilfe von Graph-Matching-Queries gezielt nach Mustern suchen (vgl. [Tabelle 4.2](#)). Graph-Matching-Queries umfassen sowohl einfache als auch komplexe Musterabfragen und Datenbankoperatoren wie Vereinigungen und Differenzen (vgl. [Abschnitt 3.4.2](#)). Um z. B. das technologische Know-How zu messen, lassen sich die Anzahl der Patente oder Entwicklungsmitarbeiter eines Werkes zählen. Durch Muster- und Pfadabfragen könnte das standortbezogene Produktportfolio analysiert werden. So könnte die Anzahl der angebotenen Produkte oder die Komplexität von Produktstrukturen, z. B. in Form eines maximalen Pfades in Stücklisten, bestimmt werden. Entscheidungsbäume, Graph-Matching- und Reachability-Queries werden mit einem Punkt bewertet.

Tabelle 4.6: Bewertung der Lösungsansätze zur Auswahl und Bewertung

	Entscheidungsbäume	Standortbestimmung	Transportplanung	TuZ-Modelle	AdjacencyQueries	Graph-Matching-Queries	Reachability-Queries	Graph-Classification	Graph-Clustering	GPM	Graphvisualisierung	Link-Prediction
Auswahl von Gestaltungsalternativen	1	0	0	0	0	0	0	1	1	1	0	0
Auswahl von Supply-Chain-Partner	1	0	0	0	1	1	1	1	0	1	0	2
Bestimmung von Standortrollen	1	0	0	0	0	1	1	1	0	1	0	0

4.2.2 Lösungsansätze in der Planungsebene

Nach der Untersuchung graphbasierter Lösungsansätze auf der Gestaltungsebene erfolgt die Untersuchung auf der Planungsebene. Die Aufgabenbereiche der Planungsebene gliedern sich in die Bedarfs-, Netzwerk-, Beschaffungs- und Distributionsplanung sowie in das Order Promising ein (vgl. [Abschnitt 2.3.2](#)).

4.2.2.1 Bedarfsplanung

Die Hauptaufgabe der Bedarfsplanung ist die Erstellung von Bedarfsprognosen auf Basis von vergangenheitsbasierten und aktuellen Daten (vgl. [Abschnitt 2.3.2](#)). Durch statistische Prognoseverfahren werden Nachfragemuster und Auffälligkeiten, z. B. Saisonalitäten, Promotionen oder Trends des Gesamtmarktes, aus den Daten extrahiert. Auch werden Bedarfsprognosen nach Regionen, Zielgruppen und Produktgruppen erstellt und Ursache-Wirkungsbeziehungen wie Preisänderungen analysiert (vgl. [Abschnitt 2.3.2](#)). Eine mit der Bedarfsplanung einhergehende Herausforderung ist der Bullwhip-Effekt, der aufgrund verzerrter Nachfrageinformationen entsteht und die Bedarfsplanung verkompliziert (vgl. [Abschnitt 2.3.2](#)). Zur Reduzierung des Bullwhip-Effekts wird die Optimierung des Informationsflusses angestrebt. Hierbei werden Bedarfe und mögliche Einflussfaktoren der Verkaufszahlen direkt weitergegeben, so dass Bedarfsprognosen präziser erstellt werden können (vgl. [Abschnitt 2.3.2](#)).

Zur Erstellung von Bedarfsprognosen eignen sich GPM-Algorithmen, da sie zur Erkennung interessanter Muster dient (vgl. [Tabelle 4.3](#)). In der Bedarfsplanung stellen interessante Muster Nachfragemuster dar, aus denen sich zukünftige Bedarfe herleiten lassen (vgl. [Abschnitt 2.3.2](#)). Durch GPM-Algorithmen lassen sich Zusammenhänge erkennen, die für den Anwender nicht direkt ersichtlich sind (vgl. [Abschnitt 3.4.3](#)). So könnten indirekte Abhängigkeiten zwischen Absatzzahlen verschiedener Produkte oder Branchen aufgedeckt werden, um Bedarfe präziser zu prognostizieren. So könnte ein Umsatzeinbruch in Branche B, die über mehrere Knoten mit Branche A verbunden ist, zu einem zeitversetzten Umsatzeinbruch in Branche A führen. Darüber hinaus werden GPM-Algorithmen zur Aufdeckung von Anomalien eingesetzt (vgl. [Tabelle 4.3](#)). Auf die Erstellung von Bedarfsprognosen übertragen, könnten Nachfragemuster, die auf den Bullwhip-Effekt deuten, erkannt werden. Der Bullwhip-Effekt wird durch das Aufschaukeln der Nachfrage zwischen den SC-Partnern charakterisiert (vgl. [Abschnitt 2.3.2](#)). Durch GPM-Algorithmen könnten Nachfragemuster identifiziert werden, die zu sehr von der zuvor eingegangenen Nachfrage des SC-Partners abweichen. Hierbei könnte ein vordefinierter Schwellenwert bestimmt werden, um eine gewisse Abweichung der Nachfrage zu tolerieren (vgl. [Abschnitt 3.4.3](#)). So könnten Stellen identifiziert werden, an denen der Bullwhip-Effekt entsteht, so dass Gegenmaßnahmen zur Verringerung des Bullwhip-Effekts ergriffen werden könnten. Insgesamt erhalten GPM-Algorithmen in [Tabelle 4.7](#) einen Punkte, da sie zur Datenauswertung und zur Identifikation des Bullwhip-Effekts eingesetzt werden können. Jedoch werden zur Erstellung von Bedarfsprognosen noch keine GPM-Algorithmen genutzt (vgl. [Abschnitt 3.4.3](#)).

Das Graph-Clustering wird in der Bedarfsplanung zur Bildung von Produktclustern eingesetzt, um das Erstellen von Bedarfsprognosen zu vereinfachen. Durch die Zusammenfassung zu Produktclustern lassen sich Wechselwirkungen zwischen Produkt- und Kundengruppen analysieren, so dass zielgerechte Werbung erstellt werden kann (vgl. [Abschnitt 3.4.3](#)). So kann der Einfluss von Rabattaktionen auf die Veränderung der Nachfrage einer bestimmten Kundengruppe präziser bestimmt und in der Bedarfsprognose berück-

sichtigt werden (vgl. [Abschnitt 3.4.3](#)). Graph-Cluster-Algorithmen erhalten in [Tabelle 4.7](#) einen Punkt, da sie die Beantwortung der Aufgabenstellung unterstützen, jedoch nicht vollkommen zur Lösung führen.

Link-Prediction-Algorithmen dienen zur Vorhersage von Beziehungen, die innerhalb eines bestimmten Zeitraums entstehen könnten (vgl. [Abschnitt 3.4.3](#)). Hierbei könnten Kundengruppen, die das Produkt mit erhöhter Wahrscheinlichkeit in Zukunft kaufen werden, bestimmt werden. Analog zu den betrachteten Merkmalen in sozialen Netzwerken könnten Kundenmerkmale, wie Alter, Geschlecht oder räumliche Nähe, zur Vorhersage des Produktkaufs herangezogen werden (vgl. [Abschnitt 3.4.3](#)). Link-Prediction-Algorithmen erhalten in [Tabelle 4.7](#) einen Punkt, da sie die Beantwortung der Aufgabenstellung unterstützen.

Sucht der Anwender gezielt nach einem ihm bekannten Nachfragemuster, eignen sich Graph-Matching-Queries zur Extraktion der Muster (vgl. [Abschnitt 3.4.2](#)). So könnten gezielte Abfragen, bspw. die Bestimmung der Nachfrage nach einem bestimmten Produkt in einer bestimmten Region, gestellt werden. Da die Erstellung von Bedarfsprognosen auf großen Datensammlungen basiert (vgl. [Abschnitt 2.3.2](#)), können Graph-Matching-Queries ergänzend zu GPM-Algorithmen eingesetzt werden. So können Muster extrahiert werden, die durch GPM-Algorithmen nicht erfasst wurden. Eine vollständige Untersuchung der Daten auf Basis von Graph-Matching-Queries setzt jedoch ein entsprechendes Anwenderwissen voraus (vgl. [Abschnitt 3.4.3](#)). Graph-Matching-Queries werden mit einem Punkt in [Tabelle 4.7](#) bewertet.

Tabelle 4.7: Bewertung der Lösungsansätze in der Bedarfsplanung

	Entscheidungsbäume	Standortbestimmung	Transportplanung	TuZ-Modelle	AdjacencyQueries	Graph-Matching-Queries	Reachability-Queries	Graph-Classification	Graph-Clustering	GPM	Graphvisualisierung	Link-Prediction
Bedarfsprognosen erstellen	0	0	0	0	0	1	0	0	1	1	0	1

4.2.2.2 Netzwerkplanung

Nachfolgend werden die Aufgabenstellungen der Netzwerkplanung hinsichtlich potenzieller Lösungsansätze der Unterkategorien aus [Abbildung 4.2](#). Die Aufgabenstellungen teilen sich in die detaillierte Modellierung der SC und die Kapazitätszuordnung auf. Die Darstellung der Punktevergabe erfolgt in [Tabelle 4.8](#).

Detaillierte Modellierung der Supply Chain Auf der Planungsebene steht die detaillierte Modellierung der modifizierten SC im Fokus, um die notwendige Planungsgrundlage für die nachfolgenden Aufgabenstellungen der Planungsebene zu schaffen (vgl. [Abschnitt 2.3.2](#)). So müssen Produktions- und Distributionsstandorte sowie ihre Schlüsselkunden und -lie-

feranten mitsamt der zugehörigen Beziehungen und Eigenschaften erfasst werden (vgl. [Abschnitt 2.3.2](#)).

Eine Vereinfachung der Struktur wird nicht fokussiert, weshalb im Gegensatz zu [Abschnitt 4.2.1](#) Graph-Classification- und Graph-Clustering-Algorithmen sowie Graph-Matching-Queries nicht zur Lösung der Aufgabenstellung betrachtet werden. Im Fokus steht die detaillierte Modellierung der SC (vgl. [Abschnitt 2.3.2](#)). Wie in [Abschnitt 3.4.3](#) beschrieben, eignen sich die Graphenvisualisierung zur vollständigen Abbildung der SC. So können alle Standorte(Knoten), Beziehungen (Kanten) und Eigenschaften (Attribute) erfasst werden (vgl. [Abschnitt 3.3.2](#) und [Abschnitt 3.3.3](#)). Durch die Graphenvisualisierung werden die Elemente so angeordnet, dass die Struktur der abgebildeten SC durch den Anwender intuitiv erfasst werden kann (vgl. [Abschnitt 3.4.2](#)). Die Graphenvisualisierung wird mit zwei Punkten bewertet, da die Aufgabenstellung vollständig durch die Graphenvisualisierung beantwortet wird.

Kapazitätszuordnung Die Kapazitätszuordnung beinhaltet die Aufteilung von Produkten und ihrer Kapazitäten auf die SC-Standorte auf Basis der zuvor erstellten Bedarfsprognosen. Die Zuordnung erfolgt unter Berücksichtigung der regionalen Nachfrage und der Standortkapazitäten, bspw. vorhandene Maschinen, mithilfe von Netzwerkplanungsmodellen (vgl. [Abschnitt 2.3.2](#)).

TuZ-Modelle werden zur kostenoptimalen Zuordnung von Angebot und Nachfrage unter Berücksichtigung von Kapazitätsrestriktionen verwendet. In einer Kostenmatrix werden Anbieterzeilen und Kundenspalten sukzessiv nach aufsteigenden Produktionskosten gesperrt, bis die Nachfrage aller Kunden gedeckt ist (vgl. [Abschnitt 3.4.1](#)). Dies lässt sich auf die betrachteten Produkt- und Standortkapazitäten übertragen, wobei die Produkte eine bestehende Nachfrage und die Standorte ein Angebot darstellen. Nach aufsteigenden Kosten erfolgt die anschließende Kapazitätszuordnung. TuZ-Modelle werden mit zwei Punkten bewertet, da sie als Netzwerkplanungsmodelle angesehen werden können (vgl. [Abschnitt 2.3.2](#) und [Abschnitt 3.4.1](#)).

Zur Berücksichtigung vorhandener Restriktionen kann ein Abgleich der Produkt- mit den Standortvoraussetzungen erfolgen, um die Standorte hinsichtlich ihrer Eignung zu klassifizieren (vgl. [Abschnitt 3.4.3](#)). Sind alle notwendigen Produktionsmittel gegeben, wird der Standort mittels Label-Propagation (Graph-Classification-Algorithmen) in die Gruppe der potenziellen Standorte einsortiert (vgl. [Abschnitt 3.4.3](#)). Die Vorauswahl der Standorte bildet die Basis zur Durchführung der Kapazitätszuordnung und wird mit einem Punkt bewertet.

Graph-Clustering-Algorithmen werden zur Analyse von Flüssen und Bestimmung dichter Regionen eingesetzt (vgl. [Tabelle 4.3](#)). So könnten die betrachteten Produktionswerke als Zentren der jeweiligen Cluster definiert und dichte Nachfrageregionen um den Clusterzentren herum gebildet werden (vgl. [Abschnitt 3.4.3](#)). So könnte durch die Zuordnung von Nachfrageknoten zu den jeweiligen Clustern die Kapazitätszuordnung erfolgen, wobei durch die Berücksichtigung maximaler Flüsse Kapazitätsrestriktionen gesetzt werden könnten. Insgesamt erhalten Graph-Clustering-Algorithmen aufgrund ihres nicht-validierten Lösungsansatzes einen Punkt.

Ist die Komplexität der Abfrage für den Anwender überschaubar, kann die Prüfung manuell durch Graph-Matching-Queries erfolgen. So könnte direkt nach einem Standort gesucht werden, der die vorhandenen Produktionskapazitäten und die insgesamt geringsten Produktionskosten aufweist (vgl. [Abschnitt 3.4.2](#)). Graph-Matching-Queries erhalten einen

Punkt, da sie einen neuen bzw. nicht-validierten Lösungsansatz darstellen und von den bisherigen Lösungsansätzen abweichen.

Tabelle 4.8: Bewertung der Lösungsansätze in der Netzwerkplanung

	Entscheidungsbäume	Standortbestimmung	Transportplanung	TuZ-Modelle	AdjacencyQueries	Graph-Matching-Queries	Reachability-Queries	Graph-Classification	Graph-Clustering	GPM	Graphvisualisierung	Link-Prediction
Modellierung der Supply Chain (detailliert)	0	0	0	0	0	0	0	0	0	0	2	0
Kapazitätszuordnung	0	0	0	2	0	1	0	1	1	0	0	0

4.2.2.3 Beschaffungs- und Distributionsplanung

4.2.2.3 Die Beschaffungs- und Distributionsplanung bildet den dritten Aufgabenbereich der Planungsebene. Im Rahmen der Beschaffungs- und Distributionsplanung werden die Bestands- und Transportplanung betrachtet. Die Resultate der Punktevergabe sind in [Tabelle 4.9](#) zusammengefasst.

Bestandsplanung Die Bestandsplanung basiert auf den Ergebnissen der Bedarfs- und Netzwerkplanung. Innerhalb der Bestandsplanung werden optimale Bestellrhythmen, Beschaffungsmengen und Sicherheitsbestände festgelegt (vgl. [Abschnitt 2.3.2](#)). Hierbei wird zwischen Verfahren zur Bestimmung fester Bestellrhythmen und bestandsorientierten Verfahren zur Bestellauslösung differenziert. Optimale Beschaffungsmengen und Sicherheitsbestände werden durch mathematische Berechnungen bestimmt (vgl. [Abschnitt 2.3.2](#)).

Zur Unterstützung der Bestimmung optimaler Bestellrhythmen können GPM-Algorithmen eingesetzt werden, um Bestellungen der Vergangenheit zu analysieren. Durch GPM-Algorithmen können häufig vorkommende oder charakteristische Bestellmuster identifiziert werden (vgl. [Tabelle 4.3](#)). So können regelmäßige Bestellungen oder besonders hohe Bestellmuster gemessen an einem vordefinierten Schwellenwert identifiziert werden (vgl. [Abschnitt 3.4.3](#)). Durch anschließende Auswertung der Zeitabstände zwischen den betrachteten Bestellmustern könnten optimale Bestellrhythmen hergeleitet werden.

Bei bestandsorientierten Verfahren lassen sich Graph-Matching-Queries einsetzen, um Datenbankabfragen des Lagerbestandes durchzuführen (vgl. [Abschnitt 3.4.2](#)). So könnte überprüft werden, welche Produkte ihren Sicherheitsbestand unterschritten haben, um Bestellungen auszulösen (vgl. [Abschnitt 2.3.2](#)). Da, wie in [Abschnitt 2.3.2](#) beschrieben, optimale Beschaffungsmengen und Sicherheitsbestände durch mathematische Formeln der Bestandsplanung berechnet werden, lassen sich keine graphbasierten Lösungsansätze der untersuchten Kategorien entdecken. Jedoch können in der Datenbank Neo4j mathematische Formeln auf Basis der Programmiersprache Java implementiert werden, so dass Formeln der Be-

standsplanung eingesetzt werden könnten (vgl. [Abschnitt 3.3.3](#)). Da die mathematischen Operationen auf bekannte Muster ausgeführt werden, wird dieser Lösungsansatz den Graph-Matching-Queries zugeordnet. Insgesamt werden sowohl GPM-Algorithmen als auch Graph-Matching-Queries mit einem Punkt bewertet.

Transportplanung In der Transportplanung werden unter Berücksichtigung der Ergebnisse aus der Bedarfs- und Netzwerkplanung optimale Transportrouten und Liefergebiete bestimmt (vgl. [Abschnitt 3.4.1](#)). Die Transportrouten werden unter Berücksichtigung vordefinierter Zielkriterien und Transportrestriktionen gebildet (vgl. [Abschnitt 3.4.1](#)).

Durch TuZ-Modelle werden Liefergebiete festgelegt und Transportrouten zwischen Anbietern und Kunden geschaffen (vgl. [Tabelle 4.1](#)). Ziel ist die Minimierung der Gesamtkosten des Transportes durch Optimierung der transportierten Mengeneinheiten und der betrachteten Fixkosten (vgl. [Abschnitt 3.4.1](#)). TuZ-Modelle stellen durch die Literatur validierte Verfahren zur Transportplanung dar und beantworten die Aufgabenstellung vollständig (vgl. [Abschnitt 3.4.1](#)). Deshalb werden TuZ-Modelle mit zwei Punkten bewertet.

Zur detaillierten Transportplanung werden Touren durch die Tourenplanung bestimmt. Die Tourenplanung ermöglicht die Berücksichtigung einer Vielzahl unterschiedlicher Tourenprobleme und Restriktionen (vgl. [Abschnitt 3.4.1](#)). Die Belieferung von Kunden stellt i. d. R. ein knotenorientiertes Problem dar, da eine Route zwischen einer bestimmten Anzahl an Knoten gesucht wird (vgl. [Abschnitt 3.4.1](#)). Hierbei ist das Travelling-Salesman-Problem vielseitig einsetzbar, da es auch zur Kommissionierung von Aufträgen in einem Lager und zur Planung von Touren zwischen einer bestimmten Anzahl von Kunden eingesetzt werden kann (vgl. [Abschnitt 3.4.1](#)). Auch lassen sich, wie in [Tabelle 4.1](#) beschrieben, Kapazitäts- oder Zeitrestriktionen berücksichtigen. Zur Planung von Transportrouten wird eine Vielzahl an unterschiedlichen graphbasierten Lösungsansätzen verwendet, so dass die Tourenplanung mit zwei Punkten bewertet wird (vgl. [Abschnitt 3.4.1](#)).

Die Festlegung von Liefergebieten könnte durch Adjacency-Queries erfolgen. Adjacency-Queries beschreiben Nachbarschaftsabfragen, die den nächsten Nachbar oder die Nachbarn innerhalb einer festgelegten Weglänge k bestimmen (vgl. [Abschnitt 3.4.2](#)). Auf die Festlegung von Liefergebieten übertragen, könnten Nachbarschaften definiert werden, die von einem Lagerstandort innerhalb einer bestimmten Distanz versorgt werden können. Durch Bestimmung des jeweils nächsten Nachbarn unter Ausschließung der bereits besuchten Nachbarn könnten optimale Transportrouten, ähnlich dem Dijkstra-Algorithmus aus [Abschnitt 3.4](#) bestimmt werden. Da Adjacency-Queries ähnlich zu validierten graphbasierten Verfahren die Aufgabenstellung der Transportplanung lösen, erhalten Adjacency-Queries ebenfalls zwei Punkte.

Zur Berechnung kürzester Transportrouten eignen sich auch Reachability-Queries, da sie zur Pfadabfrage eingesetzt werden und kürzeste Wege unter Berücksichtigung relevanter Attribute bestimmen können (vgl. [Tabelle 4.2](#)). Auch wird getestet, ob Verbindungen zwischen zwei Knoten bestehen bzw. ein Knoten von einem Wurzelknoten erreichbar ist (vgl. [Abschnitt 3.4.2](#)). So könnten kostenminimale Routen, bspw. kürzeste oder schnellste Routen, bestimmt werden. Reachability-Queries werden mit zwei Punkten bewertet, da der Lösungsansatz der Vorgehensweise in der Tourenplanung gleicht.

Graph-Clustering-Algorithmen könnten ebenfalls zur Bestimmung von Liefergebieten eingesetzt werden. So lassen sich durch Graph-Clustering-Algorithmen dichte Regionen bestimmen, die jeweils ein Liefergebiet darstellen könnten (vgl. [Abschnitt 3.4.3](#)). Jedoch eignen sich Graph-Clustering-Algorithmen nicht zur Bestimmung von Transportrouten,

da die Algorithmen auf eine Vereinfachung und Zusammenfassung der Netzwerkstruktur durch Clusterbildung abzielen (vgl. [Abschnitt 3.4.1](#)). Deshalb werden Graph-Clustering-Algorithmen mit einem Punkt bewertet.

Tabelle 4.9: Bewertung der Lösungsansätze in der Beschaffungs- und Distributionsplanung

	Entscheidungsbäume	Standortbestimmung	Transportplanung	TuZ-Modelle	AdjacencyQueries	Graph-Matching-Queries	Reachability-Queries	Graph-Classification	Graph-Clustering	GPM	Graphvisualisierung	Link-Prediction
Bestandsplanung	0	0	0	0	0	1	0	0	0	1	0	0
Transportplanung	0	0	2	2	2	0	2	0	1	0	0	0

4.2.2.4 Order Promising

Das Order promising prüft, ob Liefertermine, Liefermengen (Verfügbarkeitsprüfung) und geforderte Produktkonfigurationen (Machbarkeitsprüfung) des Kunden erfüllt werden können. Hierfür werden gezielte Abfragen und Manipulationen auf einer Datenbank durchgeführt, um Informationen zu erhalten oder Produktmengen zu reservieren (vgl. [Abschnitt 2.3.2](#)).

Verfügbarkeitsprüfung Um vorhandene Kapazitäten abzufragen oder Liefertermine zu berechnen, eignen sich Adjacency-Queries, Graph-Matching- und Reachability-Queries. Durch die Algorithmen lassen sich Knoten, Pfade oder Muster extrahieren, so dass dies Datenabfrage, wie in [Abschnitt 3.3.3](#) beschrieben, entspricht. Adjacency-Queries könnten zur Abfrage des gesuchten Produktes mit dem nächstmöglichen Liefertermin eingesetzt werden. Auch könnten Graph-Matching-Queries Produktdaten extrahieren. Zur Berechnung kürzester Lieferzeiten könnten Graph-Reachability-Queries ebenso eingesetzt werden (vgl. [Tabelle 4.2](#)). Insgesamt werden alle drei Algorithmen mit jeweils einem Punkt bewertet, da sie den in der Literatur beschriebenen Datenbankabfragen entsprechen und ergänzende Funktionen bereitstellen (vgl. [Abschnitt 2.3.2](#)). Zwei Punkte werden nicht vergeben, da die drei Algorithmen als Ganzes aufgefasst werden und erst in Kombination zu einer vollständigen Lösung der Aufgabenstellung führen.

Machbarkeitsprüfung Auch Produktkonfigurationen lassen sich durch die drei zuvor genannten Graph-Management-Algorithmen abfragen. Adjacency-Queries finden u. a. in Empfehlungssystemen Anwendung (vgl. [Tabelle 4.2](#)). Auf Machbarkeitsprüfungen übertragen, könnten alternative Produkte vorgeschlagen werden, falls ein gewünschtes Produkt bzw. eine gewünschte Produktkonfiguration nicht mehr verfügbar ist. Graph-Matching-Queries könnten zur Suche nach bestimmten Produktkonfigurationen in Form von Suchmustern eingesetzt werden, um alle gewünschten Produktinformationen zu extrahieren (vgl. [Abschnitt 3.4.2](#)). Reachability-Queries prüfen, ob Verbindungen zwischen zwei Knoten existieren (vgl. [Tabelle 4.2](#)). So könnte geprüft werden, ob eine Verbindung zwi-

schen den gewünschten Produktkomponenten und der Konfiguration besteht. Die drei Algorithmen werden ebenfalls mit jeweils einem Punkt bewertet.

Es wird die Annahme getroffen, dass das Wissen des Anwenders, bspw. eines Vertriebsmitarbeiters, zur Produktabfrage ausreicht, so dass Graph-Mining-Algorithmen nicht betrachtet werden. So eignen sich die Algorithmen eher für die autonome Entwicklung neuer Produktkonfigurationen oder der Bildung von Produktgruppen. Im Gegensatz zu den Aufgabenbereichen aus der Gestaltungsebene, wird im Order Promising kein neues Wissen abgefragt (vgl. [Abschnitt 2.3.1](#) und [Abschnitt 2.3.2](#)). Aufgabenstellungen des Order Promisings fokussieren die Abfrage vorhandener Informationen, so dass Graph-Mining-Algorithmen nicht betrachtet werden (vgl. [Abschnitt 2.3](#)).

Tabelle 4.10: Bewertung der Lösungsansätze im Order Promising

	Entscheidungsbäume	Standortbestimmung	Transportplanung	TuZ-Modelle	AdjacencyQueries	Graph-Matching-Queries	Reachability-Queries	Graph-Classification	Graph-Clustering	GPM	Graphvisualisierung	Link-Prediction
Verfügbarkeitsprüfung	0	0	0	0	1	1	1	0	0	0	0	0
Machbarkeitsprüfung	0	0	0	0	1	1	1	0	0	0	0	0

4.2.3 Gesamtbewertung und Zusammenfassung

In der Gesamtbewertung werden alle Bewertungsmatrizen aus [Abschnitt 4.2.1](#) und [Abschnitt 4.2.2](#) zu einer Bewertungsmatrix in [Tabelle 4.11](#) zusammengeführt. Hierbei werden auf horizontaler Ebene die Aufgabenbereiche und Aufgabenstellungen aus [Abschnitt 4.1.1](#), auf vertikaler Ebene die Unterkategorien und Kategorien aus [Abschnitt 4.1.2](#) aufgetragen. Dann werden die Punkte auf vertikaler Achse pro Unterkategorie aufsummiert und nach nicht-validierten, validierten und den insgesamt bewerteten Lösungsansätzen unterteilt. Zur Berechnung der nicht-validierten Lösungsansätze werden alle zweier Punktebewertungen herausgerechnet und in der Zeile der validierten Lösungsansätze eingetragen. Dies soll die Interpretation der vergebenen Punkte erleichtern. So heben bspw. die validierten Lösungsansätze Unterkategorien hervor, deren Lösungsansätze sich mit hoher Wahrscheinlichkeit implementieren lassen (vgl. [Abschnitt 4.2](#)). Die unterschiedlichen Bewertungshintergründe gilt es bei der Interpretation der aufsummierten Endwerte zu berücksichtigen, da eine vergleichende Aussage hinsichtlich der Güte der betrachteten Kategorien vermieden werden soll.

Nachfolgend werden die vergebenen Punkte in [Tabelle 4.11](#) analysiert und interpretiert. Die Unterkategorien werden nur innerhalb ihrer Kategorie betrachtet, da sie aufgrund der unterschiedlichen Bewertungshintergründe nur bedingt mit anderen Unterkategorien vergleichbar sind. So erhalten Unterkategorien der graphbasierten Modelle im SCM tendenziell höhere Punkte aufgrund validierter Lösungsansätze, wohingegen Graph-Mining-Algorithmen hohe Punktzahlen durch flexible Einsatzmöglichkeiten von nicht-validierten Lösungsansätzen erreichen (vgl. [Abschnitt 4.2.1](#) und [Abschnitt 4.2.2](#)). Nur die drei Kategorien

werden auf Basis ihrer durchschnittlichen Punktzahlen miteinander verglichen, um die vergebene Punktzahl interpretieren zu können.

[Tabelle 4.12](#) zeigt die durchschnittlich vergebene Punktzahl hinsichtlich der drei untersuchten Kategorien. Hierbei wird nach der Gestaltungsebene, der Planungsebene sowie der insgesamt durchschnittlich vergebenen Punktzahl unterschieden. Dies soll die Einsatzschwerpunkte der einzelnen Kategorien aufdecken und die Interpretation der Ergebnisse unterstützen.

Tabelle 4.11: Gesamtbewertung graphbasierter Lösungsansätze hinsichtlich logistischer Aufgabenstellungen

		Graphbasierte Modelle im SCM				Graph-Management-Algorithmen			Graph-Mining-Algorithmen					
		Entscheidungsbäume	Modelle der Standortbestimmung	Tourenplanung	TuZ-Modelle	Adjacency-Queries	Graph-Matching-Queries	Reachability-Queries	Graph-Classification	Graph-Clustering	Graph-Pattern-Mining	Graphvisualisierung	Link-Prediction	
		0: Keine entdeckten Lösungsansätze 1: Keine Validierung durch Literatur oder bestehende Lösungsansätze 2: Validierung durch Literatur oder bestehende Lösungsansätze												
Gestaltungsebene	Erstellung von Planungsszenarien	Modellierung der SC	0	0	0	0	0	1	0	1	1	0	2	0
		Simulation der Unternehmensumwelt	1	0	0	0	0	0	0	0	1	0	1	0
	Erstellung von Gestaltungsalternativen	Geografische Standortbestimmung	0	2	0	0	1	1	0	0	1	0	0	0
		Netzwerkkonfiguration	0	0	0	2	0	0	1	0	2	0	0	2
	Auswahl und Bewertung	Auswahl von Gestaltungsalternativen	1	0	0	0	0	0	0	1	1	1	0	0
		Auswahl von SC-Partnern	1	0	0	0	1	1	1	1	0	1	0	2
Bestimmung von Standortrollen		1	0	0	0	0	1	1	1	0	1	0	0	
Planungsebene	Bedarfsplanung	Bedarfsprognose erstellen	0	0	0	0	0	1	0	0	1	1	0	1
	Netzwerkplanung	Modellierung der SC (det.)	0	0	0	0	0	0	0	0	0	0	2	0
		Kapazitätszuordnung	0	0	0	2	0	1	0	1	1	0	0	0
	Beschaffungs- und Distributionsplanung	Bestandsplanung	0	0	0	0	0	1	0	0	0	1	0	0
		Transportplanung	0	0	2	2	2	0	2	0	1	0	0	0
	Order Promising	Verfügbarkeitsprüfung	0	0	0	0	1	1	1	0	0	0	0	0
Machbarkeitsprüfung		0	0	0	0	1	1	1	0	0	0	0	0	
Nicht-validierte Lösungsansätze		4	0	0	0	4	9	5	5	7	5	1	1	
Validierte Lösungsansätze		0	2	2	6	2	0	2	0	2	0	4	4	
Gesamtbewertung Lösungsansätze		4	2	2	6	6	9	7	5	9	5	5	5	

Tabelle 4.12: Durchschnittliche Punktzahl pro Kategorie nach Aufgabenebene

Kategorie	Gestaltungsebene	Planungsebene	Gesamte Punktzahl
Graphbasierte Modelle im SCM	2	1,5	3,5
Graph-Management- Algorithmen	3	4,3	7,3
Graph-Mining-Algorithmen	4	1,8	5,8
Gesamte Punktzahl	3	2,5	5,5

Graphbasierte Modelle im SCM werden in der Literatur bei ausgewählten logistischen Fragestellungen eingesetzt (vgl. [Abschnitt 3.4.1](#)). Dies erklärt die im Vergleich zu den Graph-Management- und Graph-Mining-Algorithmen niedrig ausfallende Punktzahl der untersuchten Modelle (vgl. [Tabelle 4.12](#)). Im Gegensatz zu den betrachteten Algorithmen wurden die Modelle für den Einsatz im SCM entwickelt, so dass die maximale Punktzahl in den entsprechenden Aufgabenstellungen vergeben wurde (vgl. [Tabelle 4.11](#)). Jedoch sind die SCM-Modelle der jeweiligen Aufgabenstellung so angepasst, dass eine Übertragung in eine andere Aufgabenstellung nur begrenzt möglich ist. Dementsprechend ergeben sich wenig neue Lösungsansätze, so dass die Punktevergabe in dieser Kategorie insgesamt niedriger als in den anderen Kategorien ausfällt (vgl. [Tabelle 4.12](#)). Die Differenz von 0,5 Punkten zwischen der Gestaltungsebene und Planungsebene zeigt, dass die graphbasierten Modelle in beiden Ebenen gleichermaßen Einsatz finden (vgl. [Tabelle 4.12](#)).

Eine Ausnahme bilden Entscheidungsbäume, die nicht speziell für den Einsatz im SCM entwickelt wurden und ein breiteres Einsatzspektrum aufweisen (vgl. [Abschnitt 3.4.1](#)). So sind ihre Lösungsansätze noch nicht durch vergleichbare Lösungsansätze in der Literatur validiert. Als einzige Kategorie weisen Entscheidungsbäume Lösungsansätze in vier verschiedene Aufgabenstellungen auf und erhalten in ihrem Bereich die zweithöchste Punktebewertung. Nur TuZ-Modelle weisen mit sechs Punkten eine höhere Punktzahl auf ([Tabelle 4.11](#)). Dies liegt einerseits an der Zusammenfassung von zwei Modelltypen, andererseits an den validierten Lösungsansätzen hinsichtlich der betrachteten Aufgabenstellungen (vgl. [Abschnitt 4.2.1](#) und [Abschnitt 4.2.2](#)). Transportmodelle werden auf verschiedenen Ebenen für die Gestaltung und Planung des Transportes verwendet. Auch lassen sich Zuordnungsmodelle neben der Mitarbeiterplanung für die Zuordnung von Kapazitäten und Nachfragen einsetzen (vgl. [Tabelle 4.1](#)).

Insgesamt weisen die untersuchten graphbasierten Modelle im SCM mit einer durchschnittlichen Bewertung von 3,5 Punkten pro Kategorie die niedrigste durchschnittliche Punktzahl der untersuchten Kategorien auf (vgl. [Tabelle 4.12](#)). Da die Modelle durch die Literatur validiert sind, können sie jedoch als Forschungsgrundlage nicht-validierter Lösungsansätze anderer Kategorien verwendet werden.

Graph-Management-Algorithmen weisen mehr Lösungsansätze in den nicht-validierten als in den validierten Lösungsansätzen auf (vgl. [Tabelle 4.11](#)). Dies lässt sich durch die unterschiedlichen Beispiele in [Tabelle 4.2](#) erklären: Graph-Management-Algorithmen werden in vielen Einsatzfeldern, z. B. in sozialen Netzwerken, angewendet (vgl. [Tabelle 4.2](#)). Auf Basis der Beispiele aus [Tabelle 4.2](#) lassen sich Lösungsansätze zur Lösung logistischer Aufgabenstellungen herleiten. So ergibt sich eine Vielzahl an nicht-validierten Lösungsansätzen, die sich in den vergebenen Punkten widerspiegeln. Jedoch lassen sich durch Gra-

ph-Management-Algorithmen auch an vorhandene Lösungsansätze im SCM anschließen. So kann in der Transportplanung an TuZ-Modelle sowie an die Tourenplanung angeknüpft werden (vgl. [Abschnitt 4.2.2](#)). So erhalten Graph-Management-Algorithmen die höchste Gesamtbewertung mit durchschnittlich 7,3 Punkten (vgl. [Tabelle 4.12](#)).

Die meisten Lösungsansätze weisen Graph-Matching-Queries mit neun Punkten auf (vgl. [Tabelle 4.11](#)). Dies begründet sich mit der Flexibilität von Musterabfragen. Ein Anwender, der das benötigte Hintergrundwissen hat, kann von einfachen bis komplexen Suchmustern mit unbekanntem Attributeigenschaften eine Vielzahl an Muster abfragen (vgl. [Abschnitt 3.4.2](#)). Insgesamt liegt die Mehrzahl der Lösungsansätze von Graph-Management-Queries im Bereich der Planungsebene. So wurden im Durchschnitt 4,3 Punkte auf der Planungsebene und drei Punkte auf der Gestaltungsebene vergeben (vgl. [Tabelle 4.12](#)). Dies lässt sich durch die Komplexität der Aufgabenstellungen auf der Gestaltungsebene erklären (vgl. [Abschnitt 2.3.1](#)). Aufgabenstellungen der Gestaltungsebene verlangen eine Betrachtung auf strategischer Ebene, die häufig eine Aufbereitung von Zusammenhängen und Ursache-Wirkungsbeziehungen erfordert (vgl. [Abschnitt 2.3.1](#)). So ergibt sich die Herausforderung, unbekanntes Wissen, das sich auf die Entscheidung des Anwenders auswirken könnte, aufzudecken (vgl. [Abschnitt 4.1.1](#) und [Abschnitt 3.1](#)). Da Graph-Management-Algorithmen auf den Input, bspw. vorgegebene Suchmustern, Pfaden oder Knoten, des Anwenders basieren, könnten bei unzureichendem Anwenderwissen nicht alle notwendigen Zusammenhänge erkannt werden (vgl. [Abschnitt 3.4.2](#) und [Abschnitt 3.4.3](#)). Bei dieser Herausforderung können, wie in [Abschnitt 3.4.3](#) beschrieben, Graph-Mining-Algorithmen genutzt werden.

Mit durchschnittlich 5,8 Punkten sind die Lösungsansätze von Graph-Mining-Algorithmen um 1,5 Punkte geringer als die Punkte der Graph-Management-Algorithmen bewertet (vgl. [Tabelle 4.12](#)). Dies lässt sich durch die größtenteils nicht-validierten Lösungsansätze der Kategorie erklären, die mit je einem Punkt geringer als die validierten Lösungsansätze gewichtet werden (vgl. [Tabelle 4.11](#)). So lassen sich wenig vergleichbare Lösungsansätze zu den bisherigen Ansätzen finden, so dass zwei Punkte im Bereich von Graph-Mining-Algorithmen selten vergeben wurden (vgl. [Tabelle 4.11](#)).

Dennoch lässt sich eine Vielzahl unterschiedlicher Lösungsansätze identifizieren (vgl. [Tabelle 4.11](#)). Insbesondere auf der Gestaltungsebene lassen sich Einsatzschwerpunkte der Graph-Mining-Algorithmen mit einer durchschnittlichen Punktzahl von vier zu 1,8 Punkten auf der Planungsebene erkennen (vgl. [Tabelle 4.12](#)). Dies liegt, wie zuvor begründet, an der Komplexität der Aufgabenstellungen auf der Gestaltungsebene. Das Graph-Mining wird zur Extraktion von Wissen in Form von Mustern eingesetzt, die den Anwender bei seiner Entscheidung unterstützen sollen (vgl. [Abschnitt 3.1](#) und [Abschnitt 3.4.3](#)). Auf der Gestaltungsebene werden längerfristige Entscheidungen im Planungszeitraum von mehreren Jahren hinsichtlich der Gestaltung der SC unter Berücksichtigung ihrer Umwelt getroffen (vgl. [Abschnitt 2.3.1](#)). Dies erfordert ein umfassendes Wissen über Ursache-Wirkungs-Beziehungen zwischen SC-Teilnehmern sowie ihrer Umwelt und unbekanntem Zusammenhängen in der SC (vgl. [Abschnitt 3.1](#)). Es müssen unterschiedliche Planungsszenarien und Gestaltungsalternativen gebildet und bewertet werden, wobei die Komplexität von SC stetig wächst, so dass Anwender in ihrer Entscheidung unterstützt werden müssen (vgl. [Abschnitt 2.1.1](#) und [Abschnitt 3.1](#)). Hierbei helfen Graph-Mining-Algorithmen, die unbekanntes Wissen für den Anwender aufbereiten und deshalb unterschiedliche Lösungsansätze auf Gestaltungsebene bieten (vgl. [Tabelle 4.11](#)). Auch werden sie zur Vereinfachung der dargestellten SC eingesetzt, um das Verständnis des Anwenders über die SC zu erhöhen (vgl. [Tabelle 4.3](#)).

Graph-Clustering-Algorithmen weisen mit neun Punkten, davon sieben Punkte für nicht-validierte Lösungsansätze, die meisten Lösungsansätze auf (vgl. [Tabelle 4.11](#)). Sie werden einerseits zur Vereinfachung von Strukturen, andererseits zur Analyse von Sachverhalten eingesetzt (vgl. [Tabelle 4.3](#)). So werden zur Vereinfachung Gruppen mit ähnlichen Strukturen gebildet und elementare Verbindungen hervorgehoben. Hierdurch lassen sich Netzwerkstrukturen oder die Interaktion zwischen Gruppen vereinfacht analysieren (vgl. [Abschnitt 4.2.1](#) und [Abschnitt 4.2.2](#)). Ähnlich variierende Lösungsansätze zeigen Graph-Classification-Algorithmen, die sich zur Klassifizierung und Bewertung einzelner Elemente oder Gruppen eignen (vgl. [Abschnitt 4.2.1](#)). Die anderen Graph-Mining-Algorithmen liegen mit je fünf vergebenen Punkten knapp unter dem Durchschnitt von 5,5 Punkten. Insgesamt lässt sich eine Vielzahl unterschiedlicher Lösungsansätze für Graph-Mining-Algorithmen entdecken, die jedoch meist noch validiert werden müssen. Nachfolgend werden die zuvor erörterten Ergebnisse zusammengefasst.

Insgesamt ergeben sich im Gegensatz zu den Lösungsansätzen der anderen Kategorien für graphbasierte Modelle mit Ausnahme von Entscheidungsbäumen, wenige nicht-validierte Lösungsansätze (vgl. [Tabelle 4.11](#)). Vielmehr werden die Modelle anwendungsorientiert eingesetzt und bieten meist eine vollständig validierte Lösung der betrachteten Aufgabenstellung (vgl. [Abschnitt 4.2.1](#) und [Abschnitt 4.2.2](#)). Dementsprechend erhalten graphbasierte Modelle eine unterdurchschnittliche Punkteanzahl in beiden Aufgabenebenen von insgesamt 3,5 Punkten (vgl. [Tabelle 4.12](#)).

Graph-Management-Algorithmen knüpfen an bestehende Lösungsansätze an und bergen eine Vielzahl an graphbasierten Lösungsansätzen mit der Voraussetzung, dass das zur Lösung der Aufgabenstellung benötigte Anwenderwissen vorhanden ist (vgl. [Abschnitt 4.2.1](#) und [Abschnitt 4.2.2](#)). So erzielen Graph-Management-Algorithmen auf der Gestaltungsebene durchschnittliche Werte mit drei Punkten (vgl. [Tabelle 4.12](#)). Auf der Planungsebene erhalten Graph-Management-Algorithmen jedoch überdurchschnittliche Punktebewertungen (4,3 zu 2,5 Punkten). Dies deutet auf ein hohes Untersuchungspotenzial der entdeckten Lösungsansätze hin, weshalb in [Kapitel 5](#) Graph-Management-Algorithmen in einem Fallbeispiel angewendet werden.

Graph-Mining-Algorithmen eignen sich bei Aufgabenstellungen, die das Anwenderwissen übersteigen und komplexe Zusammenhänge aufbereitet (vgl. [Abschnitt 4.2.1](#) und [Abschnitt 4.2.2](#)). Dies spiegelt sich in der Auswertung in [Tabelle 4.12](#) wider. Auf der Gestaltungsebene erhalten Graph-Mining-Algorithmen mit vier Punkten die höchste durchschnittliche Punktzahl auf dieser Aufgabenebene. Auf der Planungsebene fällt die Punktevergabe mit durchschnittlich 1,8 Punkten zu 2,5 Punkten jedoch unterdurchschnittlich aus, weshalb mehr Untersuchungspotenzial auf der Gestaltungsebene verborgen ist (vgl. [Tabelle 4.12](#)).

Insgesamt wurde in [Abschnitt 4.2.1](#) und [Abschnitt 4.2.2](#) eine Vielzahl unterschiedlicher Ansätze zur Lösung der Aufgabenstellungen aus [Abschnitt 4.1.1](#) aufgedeckt. Zum Abschluss werden alle zuvor identifizierten Lösungsansätze hinsichtlich ihrer Unterkategorien und Aufgabenstellungen in [Tabelle 4.13](#), [Tabelle 4.14](#) und [Tabelle 4.15](#) zusammengefasst. Ist die Herleitung der Lösungsansätze aufgrund unterschiedlicher Funktionen der betrachteten Unterkategorie nicht ersichtlich, werden die Ausgangsfunktionen nach den Beschreibungen aus [Tabelle 4.1](#), [Tabelle 4.2](#) und [Tabelle 4.3](#) genannt.

Tabelle 4.13: Lösungsansätze graphbasierter Modelle des SCM hinsichtlich logistischer Aufgabenstellungen

Unterkategorien	Aufgabenstellungen	Lösungsansätze
Entscheidungs- bäume	<ul style="list-style-type: none"> • Simulation der Unternehmensumwelt • Auswahl von Gestaltungsalternativen • Auswahl von SC-Partnern • Bestimmung von Standortrollen 	<ul style="list-style-type: none"> • Darstellung unterschiedlicher Planungsszenarien als Hyperknoten in einem Szenariobaum • Auswahl und Klassifizierung von Gestaltungsalternativen, Standorten oder SC-Partnern (Sukzessive Entscheidungssituationen, Klassifizierung)
Standort- bestimmung	<ul style="list-style-type: none"> • Geografische Standortbestimmung 	<ul style="list-style-type: none"> • Standortbestimmung (Distributionslager, Servicezentren etc.)
Tourenplanung	<ul style="list-style-type: none"> • Transportplanung 	<ul style="list-style-type: none"> • Bestimmung optimaler Touren unter Berücksichtigung von Restriktionen (Kosten, Zeit, etc.)
TuZ-Modelle	<ul style="list-style-type: none"> • Netzwerkkonfiguration • Kapazitätszuordnung • Transportplanung 	<ul style="list-style-type: none"> • Modellierung und Optimierung von Materialflüssen • Optimale Zuordnung von Kapazitäten zu Standorten • Konfiguration von Lieferantennetzwerken • Minimierung der Gesamtkosten von Transporten

Tabelle 4.14: Lösungsansätze von Graph-Management-Algorithmen hinsichtlich logistischer Aufgabenstellungen

Unterkategorien	Aufgabenstellungen	Lösungsansätze
Adjacency-Queries	<ul style="list-style-type: none"> • Geografische Standortbestimmung • Auswahl von SC-Partnern • Transportplanung • Verfügbarkeitsprüfung • Machbarkeitsprüfung 	<ul style="list-style-type: none"> • Bestimmung von Standorten (Umsetzung des Median-Problems durch k-Neighborhood-Queries) • Vorschlag eines SC Partners (Empfehlungssystem) • Festlegung von Liefergebieten (k-Neighborhood-Queries) • Bestimmung kürzester Routen auf (nächster Nachbar) • Produktabfrage und Vorschlag alternativer Produkte (Empfehlungssystem)
Graph-Matching-Queries	<ul style="list-style-type: none"> • Modellierung der SC • Geografische Standortbestimmung • Auswahl von SC-Partnern • Bestimmung von Standortrollen • Bedarfsplanung • Kapazitätszuordnung • Bestandsplanung • Verfügbarkeitsprüfung • Machbarkeitsprüfung 	<ul style="list-style-type: none"> • Filterung der betrachteten SC • Standortbestimmung • Bestimmung gesuchter Kennzahlen • Suche von Nachfragemustern • Manuelle Zuordnung von Kapazitäten zu Standorten • Suche nach Produkten und ihrer Anzahl • Suche nach Produktkonfigurationen
Reachability-Queries	<ul style="list-style-type: none"> • Netzwerkkonfiguration • Auswahl von SC-Partnern • Bestimmung von Standortrollen • Transportplanung • Verfügbarkeitsprüfung • Machbarkeitsprüfung 	<ul style="list-style-type: none"> • Optimierung von Transportrouten aus Sicht eines Standortes (Pfadabfragen hinsichtlich kürzester, schnellster oder günstigster Pfade) • Identifizierung von Engpassstellen (Brücken) • Suche nach räumlich nahen SC-Partnern (kürzester Pfad) • Analyse von Produktstrukturen (maximaler Pfad) • Berechnung kürzester Lieferzeiten (kürzester Pfad) • Prüfung von Verbindungen zwischen Produktkomponenten (Abfrage der Erreichbarkeit)

Tabelle 4.15: Lösungsansätze von Graph-Mining-Algorithmen hinsichtlich logistischer Aufgabenstellungen

Unterkategorien	Aufgabenstellung	Lösungsansätze
Graph-Classification	<ul style="list-style-type: none"> • Modellierung der SC • Auswahl von Gestaltungsalternativen • Auswahl von SC-Partnern • Bestimmung von Standortrollen • Kapazitätszuordnung 	<ul style="list-style-type: none"> • Einteilung der SC-Akteure in Gruppen (Label-Propagation) • Analyse von Strukturen innerhalb von SC-Akteurgruppen (Beschreibung von Gruppen) • Klassifizierung und Bewertung von Gestaltungsalternativen, SC oder Standortrollen (Label-Propagation, Graph-Classification) • Vorauswahl geeigneter SC-Standorte zur Kapazitätszuordnung (Label-Propagation)
Graph-Clustering	<ul style="list-style-type: none"> • Modellierung der SC • Simulation der Unternehmensumwelt • Geografische Standortbestimmung • Netzwerkkonfiguration • Auswahl von Gestaltungsalternativen • Bedarfsprognosen erstellen • Kapazitätszuordnung • Transportplanung 	<ul style="list-style-type: none"> • Vereinfachung und Zusammenfassung von SC (Komprimierung) • Analyse von Wechselwirkungen (SC ↔ Unternehmensumwelt, Produkte ↔ Kunden) (Flüsse) • Identifizierung kleiner Netzwerke (Graph-Clustering) • Identifizierung potenzieller Standorte oder Liefergebiete (Dichte Regionen) • Bestimmung von Konsolidierungsmaßnahmen (Komprimierung) • Identifizierung von Engpassstellen und maximaler Flüsse • Bildung von Produktgruppen (Node-Clustering)
Graph-Pattern-Mining	<ul style="list-style-type: none"> • Auswahl von Gestaltungsalternativen • Auswahl von SC-Partnern • Bestimmung von Standortrollen • Bedarfsplanung • Bestandsplanung 	<ul style="list-style-type: none"> • Zusammenführung von Gestaltungsalternativen (SC) und Planungsumgebung (Unternehmensumwelt) (Zusammenführung von Graphen) • Analyse von Verhaltens-, Qualitäts-, Nachfrage- oder Bestellmustern • Suche nach (Nachfrage-)Anomalien
Graphvisualisierung	<ul style="list-style-type: none"> • Modellierung der SC • Detaillierte Modellierung der SC 	<ul style="list-style-type: none"> • Modellierung der SC oder der Unternehmensumwelt • Detaillierte Modellierung der SC (Erfassung aller Attribute)
Link-Prediction	<ul style="list-style-type: none"> • Netzwerkkonfiguration • Auswahl von SC-Partnern • Bedarfsplanung 	<ul style="list-style-type: none"> • Konfiguration der SC (Analyse fehlender Kanten, Vorhersage von Kanten, Identifizierung überflüssiger Kanten) • Vorhersage möglicher Partnerschaften • Identifizierung potenzieller Kundengruppen

5 Validierung der Taxonomie und Beantwortung logistischer Fragestellungen auf Basis einer Graphenrepräsentation

In [Kapitel 5](#) erfolgt die Anwendung der Taxonomie aus [Abbildung 4.1](#) sowie die Beantwortung logistischer Fragestellungen der Transportplanung zur Validierung graphbasierter Lösungsansätze im SCM. Zur Validierung der Taxonomie werden ausgewählte Fragestellungen aus [Abschnitt 2.3](#), [Stich et al.](#) (vgl. [2013c](#), S. 42), [Günther](#) (vgl. [2005](#), S. 20) und [Daskin und Maass](#) (vgl. [2019](#), S. 280) entnommen und in [Abschnitt 5.1](#) mithilfe der Taxonomie sortiert. In [Abschnitt 5.2](#) erfolgt die Transportplanung in einer beispielhaften SC. Hierbei werden typische Fragestellungen der Transportplanung untersucht, hergeleitet aus [Abschnitt 2.3.2](#) und auf Basis einer Graphenrepräsentation in der Datenbank Neo4j beantwortet.

5.1 Fallbeispiel 1: Vorbereitung zur Gestaltung und Planung einer SC

Die SCM-Abteilung eines fokalen Unternehmens einer SC wurde beauftragt, die SC des gesamten Unternehmensnetzwerkes neu zu gestalten, zu planen und hinsichtlich ihrer aktuellen Konfiguration zu prüfen. Zur Anforderungsspezifikation werden ausgewählte Akteure der SC interviewt, so dass folgender Fragenkatalog gebildet wurde:

- Welche Produkte sind an welchen Standorten herzustellen?
- An welchen Orten sind neue Standorte einzurichten?
- Welchen Standorten sind welche logistischen Aufgaben zuzuordnen?
- Zwischen welchen Standorten sollten welche Beziehungen für die Beschaffung, Produktion und Distribution von Rohstoffen, Zwischen- und Fertigprodukten verwirklicht werden?
- Inwiefern besteht die Notwendigkeit, eine SC zu rekonfigurieren, um neue wettbewerbsfähige Produkte herzustellen?
- Wie sollen die Produktionsmengen zwischen den Werken unter Berücksichtigung ihrer Produktionskapazitäten und Produktionsstückkosten aufgeteilt werden?
- Von welchem Distributionszentrum aus sollen die Produkte unter Berücksichtigung der Transportkosten und der Nachfrage an die Kunden und Marktgebiete geliefert werden?
- Welcher Umfang der Kapazitätserweiterung soll genutzt werden?
- Welcher Umfang an Lagerbeständen soll aufgebaut werden, um saisonale Nachfragezyklen auszugleichen?

- Wie viele Standorte sollten in der SC genutzt werden?
- Ist der bereitgestellte Servicelevel, den der Kunde durch die vorliegende Netzwerkkonfiguration erhält, ausreichend?
- Wie sind die Backup-Pläne der vorliegenden Gestaltungsalternative für den Fall von Standortausfällen zu bewerten?

Aufgrund der heterogenen Fragestellungen soll die Abteilung Aufgabenpakete zusammenstellen und in einen Projektplan eintragen. Hierbei wird die Taxonomie aus [Abschnitt 4.1.1](#) zur Klassifizierung der zusammengetragenen Fragestellungen verwendet. Die Aufgabenpakete werden auf Basis der Aufgabenstellungen gebildet, so dass alle Fragestellungen unterhalb der Aufgabenstellungen einsortiert werden. So sollen Fragestellungen der Planungsebene in dem angefertigten Projektplan erst nach Bearbeitung der Gestaltungsebene beantwortet werden, da die Gestaltungsebene die Rahmenbedingungen festlegt. Die Bearbeitung einer Fragestellung auf der Planungsebene würde zu einem Mehraufwand führen, da eine erneute Bearbeitung der Fragestellungen nach Beendigung der SC-Gestaltung notwendig wäre (vgl. [Abschnitt 2.3](#)). Mithilfe der Taxonomie aus [Abschnitt 4.1.1](#) werden die gesammelten Fragestellungen in die Taxonomie einsortiert und aufgelistet. [Abbildung 5.1](#) zeigt die eingeordneten Fragestellungen. Um die Übersichtlichkeit zu erhöhen, werden nur Aufgabenstellungen und -bereiche abgebildet, die eine zugeordnete Fragestellung enthalten.

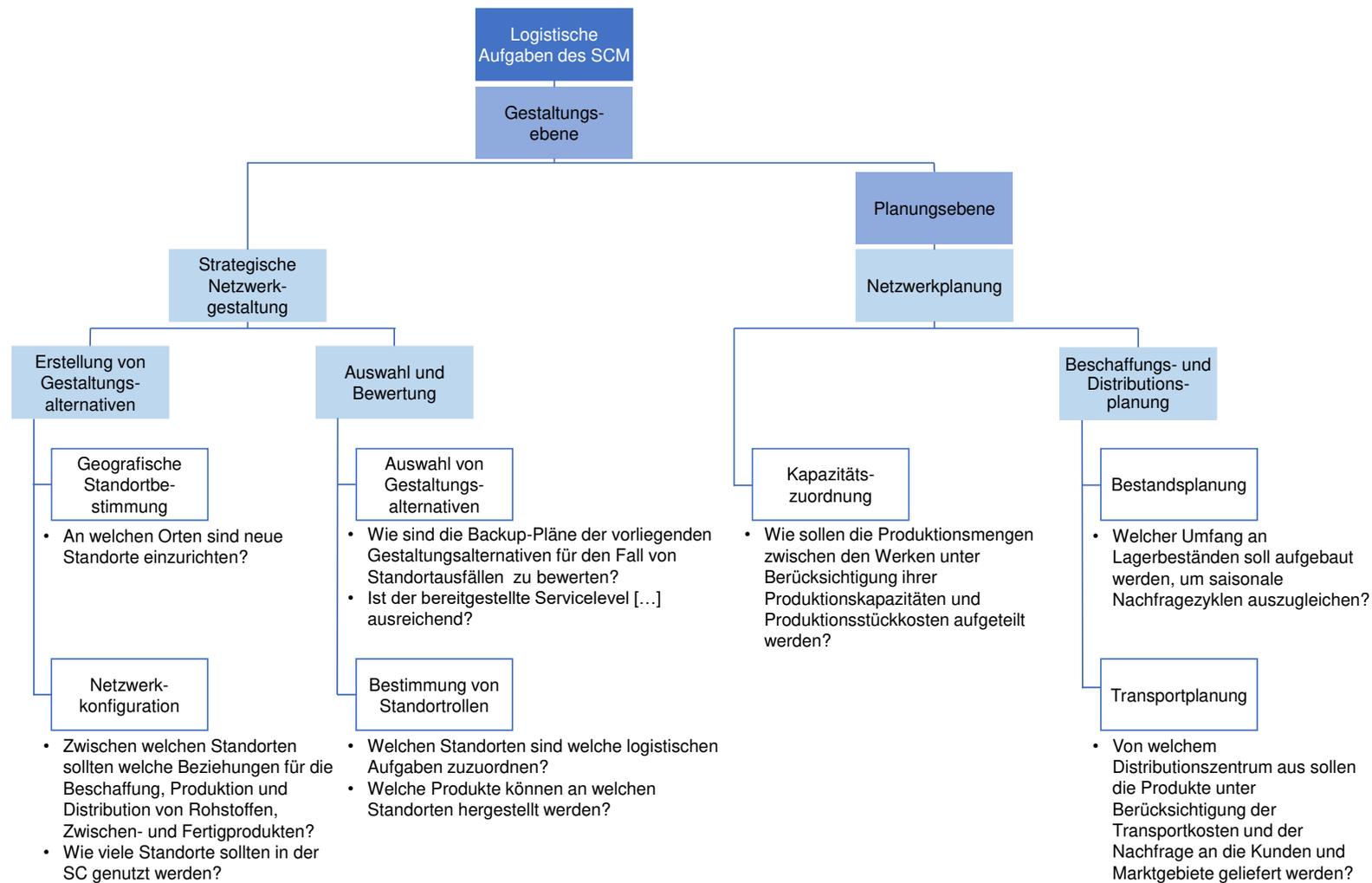


Abbildung 5.1: Klassifizierung logistischer Fragestellungen anhand der Taxonomie aus [Abbildung 4.1](#)

Durch die Taxonomie wird eine systematische Aufarbeitung der Fragestellungen ermöglicht. Hierdurch bilden sich Arbeitspakete, die von der Gestaltungsebene bis zur Planungsebene nacheinander durchgeführt werden können. Jedoch arbeitet die Abteilung heraus, dass die nachfolgenden Fragen keiner Aufgabenstellung zugeordnet werden konnten:

- Inwiefern besteht die Notwendigkeit, eine bestehende SC zu rekonfigurieren, um neue wettbewerbsfähige Produkte herzustellen?
- Welcher Umfang der Kapazitätserweiterung soll genutzt werden?

In der Abteilung wird eine anwendungsbezogene Erweiterung der Taxonomie in Erwägung gezogen, weshalb beide Fragestellungen diskutiert werden. Die erste Fragestellung stellt eine Entscheidungsfrage hinsichtlich der Netzwerkkonfiguration dar. So muss entschieden werden, ob eine Rekonfiguration der SC durchgeführt werden soll oder ob die bestehende SC ausreichend für die Herstellung wettbewerbsfähiger Produkte ist. Die Taxonomie dient jedoch nur zur Klassifizierung von Fragestellungen, eine Begründung der Entscheidung zur Durchführung der betrachteten Aufgabenstellung findet nicht statt (vgl. [Abschnitt 4.1.1](#)). Deshalb diskutiert die Abteilung über eine mögliche Erweiterung der Taxonomie um Freigabe- oder Entscheidungspunkte, die zu einer Freigabe und Bearbeitung der Fragestellung führen. So kann vor Beginn des Gestaltungs- und Planungsprozesses eine begründete Entscheidung auf Basis definierter Entscheidungskriterien zur Freigabe der Aufgabenstellungen getroffen werden. Da die Entscheidung zur Umgestaltung des Netzwerkes in diesem Beispiel auf Managementebene bereits getroffen wurde, wird die Fragestellung mit sekundärer Priorität behandelt.

Die zweite Fragestellung ist standortbezogen und behandelt den Umfang der Nutzung vorhandener Kapazitätserweiterungen. Durch das Heranziehen des SCM-Aufgabenmodells aus [Abschnitt 2.2.2](#) wird ersichtlich, dass die Fragestellung der Produktionsplanung zugeordnet wird. Durch die Erweiterung der Taxonomie um den Aufgabenbereich der Produktionsplanung und entsprechender Aufgabenstellungen, bspw. Produktionsprogrammplanung, lässt sich die zweite Fragestellung in die Taxonomie einordnen. Der SCM-Abteilung fällt auf, dass einige Aufgabenstellungen keine Fragestellungen enthalten. Zur vollständigen Gestaltung und Planung werden weitere Besprechungen und Interviews geplant, um die Durchführung der verbliebenen Aufgabenstellungen zu besprechen.

5.2 Fallbeispiel 2: Beantwortung ausgewählter Fragestellungen der Transportplanung

Nach Ausgestaltung der SC wird die SCM-Abteilung mit der Transportplanung beauftragt. So besteht zwischen einigen Akteuren der SC die Möglichkeit unterschiedlicher Transportarten, bspw. über Schiffs- oder Landverkehr, so dass verschiedene Lieferszenarien abgefragt werden sollen. Auch wurde in der Netzwerkkonfiguration eine neue potenzielle Verbindung zwischen dem Produzenten und dem Einzelhandel geschaffen, da durch vermehrte Kooperationen mit Einzelhändlern in räumlicher Nähe eine direkte Lieferung ggf. kostengünstiger ausfallen könnte als die Lieferung über einen Großhändler. Zur Veranschaulichung der Situation wurde die SC in [Abbildung 5.2](#) modelliert und als Eigenschaftsgraph dargestellt.

Zusätzlich zu den Akteuren der SC wurden die Rohstoffe und Transportmittel modelliert, da mehrere Lieferanten u. a. die gleichen Rohstoffe anbieten und abhängig von der betrachteten Zielgröße ausgewählt werden sollen. Je nach Akteur können unterschiedliche Transportmittel ausgewählt werden, die sich auf die Lieferkosten und -wege auswirken,

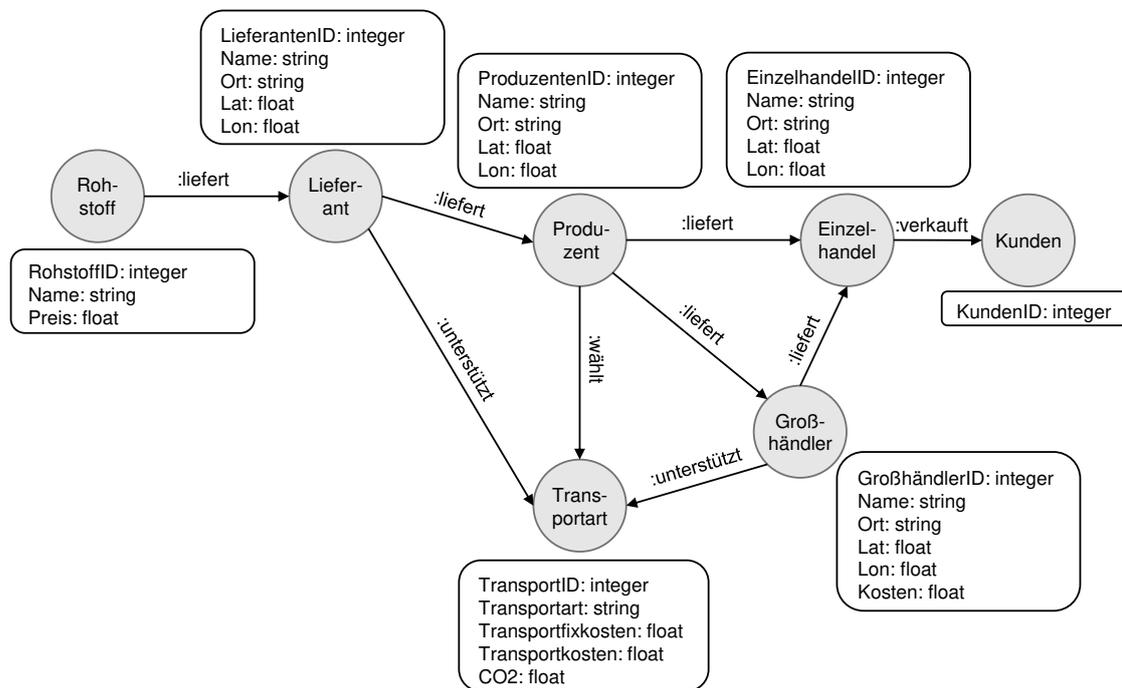


Abbildung 5.2: Darstellung der SC durch ein Graphenmodell

wobei die Wahl durch die SCM-Abteilung des Produzenten erfolgt. Zu beachten ist, dass aufgrund lokaler Gegebenheiten nicht jeder Akteur jedes Transportmittel unterstützt, da bspw. kein Hafen im näheren Umkreis vorhanden ist. Darüber hinaus werden die unterschiedlich ausfallenden Strecken der Transportdepots der Transportarten bis zum SC-Teilnehmer vernachlässigt, da eine Transportart nur bei unmittelbarer Nähe des Lieferzieles unterstützt wird. Die Belieferung kann aufgrund struktureller Restriktionen ausschließlich durch Lastkraftwagen (LKW) erfolgen. Die Datensätze der SC werden aktuell in Tabellenform in einem CSV-Dateiformat gespeichert. Da Zusammenhänge, bspw. Transportrouten, in Tabellenform nicht intuitiv nachvollzogen werden können, beschließt die Abteilung, Daten als Graphenrepräsentation zu speichern. Hierfür werden die Datensätze mittels Cypher umgewandelt und in einer Datenbank in Neo4j gespeichert. Anschließend werden exemplarische Fragestellungen der Transportplanung von der Beschaffungs- bis zur Distributionsplanung beantwortet.

Modellierung der SC Durch den Befehl LOAD werden als CSV-Datei gespeicherte Datensätze in Entitäten oder Beziehungen nach Vorbild eines Eigenschaftsgraphen umgewandelt und gespeichert. Hierbei werden die Tabellen, wie in Abschnitt 3.2 beschrieben, gelesen. So stellen Tabellen Entitäten oder aus Entitäten zusammengesetzte Beziehungen dar. Die einzelnen Tabellenspalten bilden die Eigenschaften.

Die Erstellung der Datenbank erfolgt in der im März 2020 erschienenen Version 4.0.3 von Neo4j. Hierbei wurde zur Erstellung der Abfragen Anleitungen des Entwicklers Neo4j (2020a) verwendet. Eine Kennzeichnung von Fremdcode erfolgt nur, wenn er nicht im Rahmen der Neo4j-Anleitungen zu finden ist. Die Datensätze werden anhand ihrer Kopfzeilen importiert, wobei nur ausgefüllte Zeilen importiert werden. Knoten und Kanten können durch die Befehle CREATE oder MERGE erstellt werden, hierbei verhindert MERGE

Duplikate hinsichtlich ausgewählten Variablen. Durch SET werden die Eigenschaften der Knoten und Kanten erstellt und spezifiziert. Die Akteure der SC (Lieferanten, Produzenten, Großhändler, Einzelhändler) werden durch ihre Identifikationsnummer, ihren Namen sowie ihre Position in Längen- und Breitengraden ausgedrückt. Darüber hinaus entstehen Lagerkosten, wenn der Produzent seine Ware über den Großhändler liefert. Der [Algorithmus 5.1](#) beschreibt den Import eines Lieferanten über eine CSV-Datei.

Algorithmus 5.1: Code zum Import eines Lieferanten

```

1 LOAD CSV WITH HEADERS FROM 'file:///Lieferanten.csv' AS line
2 WITH line WHERE line.LieferantenID IS NOT NULL
3 MERGE(l:Lieferant {LieferantenID: tointeger(line.LieferantenID)
  })
4 SET l.Name= line.Name, l.ort=line.Ort, l.lat=toFloat(line.lat),
5 l.lon=toFloat(line.lon);

```

Der Abteilung liegt eine Tabelle aus geschätzten Referenzwerten zu unterschiedlichen Transportarten vor. Hierbei werden die Transportfixkosten pro Route, die Transportkosten pro Tonnenkilometer (*tkm*), die Zeitdauer pro 100 km in Stunden sowie der CO₂-Ausstoß in Gramm pro Tonnenkilometer angegeben. Die Werte orientieren sich an Beispielwerten aus der Literatur.

Tabelle 5.1: Kostentabelle der Transportarten

Transport-ID	Transportart	Transportfixkosten	Transportkosten	Zeitdauer	CO ₂
1	LKW	600,00	0,12	1,3	112
2	Bahn	1.052,00	0,05	3,28	18
3	Schiff	1.200,00	0,03	19,2	31
4	Flugzeug	1.660,00	0,15	0,5	171

Quellen: [Bühler \(2006\)](#), [Gleißner und Möller \(2009\)](#), [Umweltbundesamt \(2020\)](#)

Die Tabelle über die Transportarten wird nach dem zuvor erläuterten Prinzip ähnlich zu [Algorithmus 5.1](#) in die Datenbank überführt. Die Erstellung von Beziehungen erfolgt durch die Abfrage von Anfangs- und Endknoten und der anschließenden Erstellung einer Beziehung zwischen den betrachteten Knoten. Wenn die Beziehungen in Tabellenform vorliegen, bspw. in Form von Transportrouten, werden die CSV-Dateien extrahiert. Ansonsten werden die Beziehungen wie in [Algorithmus 5.2](#) beschrieben erstellt.

Algorithmus 5.2: Code zur Erstellung von Beziehungen

```

1 MATCH (p:Produzent), (l:Lieferant)
2 CREATE (l)-[l:liefert {distanz: NULL }]->(p)

```

Die Transportrouten zwischen den Akteuren der SC werden durch ihre Distanzen zueinander charakterisiert. Die Distanzen berechnen sich auf Basis eines Beispiels von [Neo4j \(2019\)](#). Dazu werden alle Distanzen zwischen zwei Knoten berechnet, die eine Transportroute aufweisen. Der Code wird in späteren Abfragen wiederverwendet. Die Distanz wird wie folgt berechnet:

Algorithmus 5.3: Code zur Berechnung der Distanz

```

1 MATCH (a)-[r:liefert]->(b)

```

```

2 WITH r, a, b, 2 * 6371 * asin(sqrt(haversin(radians(toInteger(a.
    lat) - toInteger(b.lat)))) + cos(radians(a.lat))*
3 cos(radians(b.lat))* haversin(radians(a.lon - b.lon)))) AS dist
4 SET r.distanz = round(dist)

```

Nachdem die SC aufgebaut wurde, erfolgt die Visualisierung der gesamten SC in Neo4j durch die Befehle MATCH und RETURN. Da sich die SCM-Abteilung auf die Visualisierung der Transportrouten fokussiert, werden die Knoten der Transportarten manuell ausgeblendet. Anschließend erfolgt die logische Anordnung der SC-Teilnehmer. Da die Transportrouten zu den Einzelhändlern neu bestimmt werden sollen, existieren noch keine Kanten zu diesen Knoten. Deshalb sind die Akteure des Einzelhandels als graue Knoten dargestellt. Erst nach Schaffung einer Transportroute werden die Farben der Einzelhändlerknoten manuell in Rosa geändert. Die Knoten des Großhändlers sind grün eingefärbt. Der Produzent ist rot. Die Knoten der Lieferanten werden in Orange dargestellt. Die Knoten der Rohstoffe sind bräunlich eingefärbt.

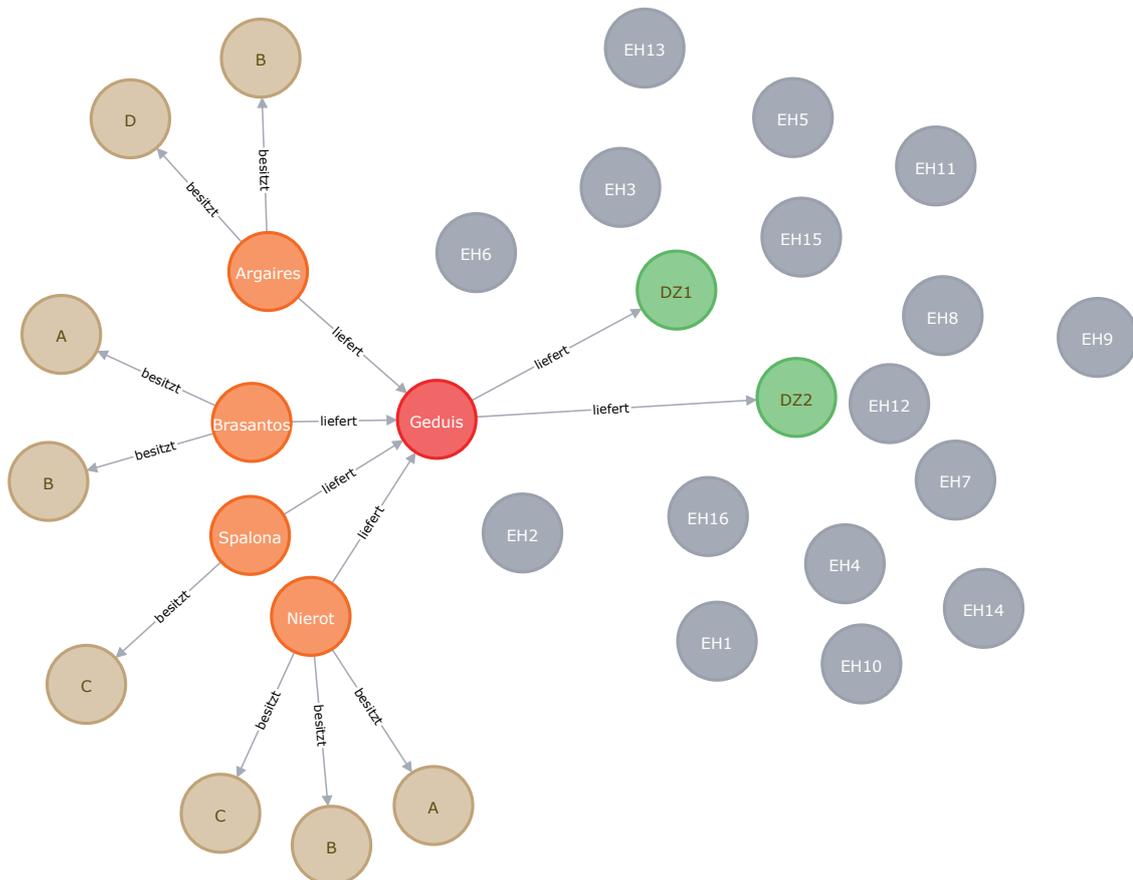


Abbildung 5.3: Visuelle Darstellung der SC

Nach der Visualisierung der SC sollen Fragestellungen der Transportplanung beantwortet werden. Hierbei werden ausgewählte Fragestellungen zusammengetragen, die sich an Fragestellungen der Beschaffungs- und Distributionsplanung orientieren (vgl. [Abschnitt 2.3.2](#)):

1. Wie sollten die Liefergebiete zwischen dem Produzenten und den Großhändlern aufgeteilt werden?
2. Welcher Lieferant bietet den Rohstoff A an?

3. Wie lang ist der kürzeste Pfad über die gesamte SC, wenn der Produzent über einen Großhändler liefert?
4. Lassen sich auch umweltfreundlichste, schnellste oder kostengünstigste Pfade über die gesamte SC bestimmen?
5. Welcher Lieferant und welche Lieferart sollten für den Rohstoff B unter Berücksichtigung der Kosten gewählt werden?
6. In welcher Reihenfolge sollte der Produzent die Einzelhändler beliefern, um die kostengünstigste Route zu wählen?

1. Wie sollten die Liefergebiete zwischen dem Produzenten und den Großhändlern aufgeteilt werden

Zur Vervollständigung der SC sollen optimale Transportrouten zwischen Einzelhändler und Produzent oder Großhändler bestimmt werden. Primär sollen die Gesamtkosten minimiert werden, wobei auch der CO₂-Ausstoß sowie die Transportdistanzen untersucht werden. Da alle Transportrouten geprüft und erneuert werden sollen, werden die zuvor existierenden Transportrouten nicht betrachtet. So muss für jeden Einzelhändler eine Transportroute, die direkt über den Produzenten oder über einen Großhändler verläuft, geprüft werden. Nachfolgend erfolgt die Abfrage exemplarisch für die ersten zwei Lieferanten.

In einer ersten Abfrage in [Algorithmus 5.4](#) werden durch die Befehle MATCH, WHERE, WITH und RETURN die Distanzen über einen Großhändler geprüft, indem die fehlende Distanz zwischen Einzelhändler und Großhändler berechnet und mit der Distanz zwischen Produzenten und Großhändler aufsummiert wird. Der CO₂-Ausstoß berechnet sich durch die Multiplikation der Gesamtdistanz mit den CO₂-Werten der Transportart. Anschließend werden die Gesamtkosten bestimmt, indem die Transportkosten mit der Distanz multipliziert und mit den Transportfixkosten sowie den Kosten des Großhändlers aufsummiert werden. Aufgrund effizienter Transportmittel fallen die Fix- und Transportkosten sowie der CO₂-Ausstoß über den Großhändler um insgesamt 10 Prozent geringer als über den Produzenten aus. Die zweite Abfrage berechnet analog zur ersten Abfrage die Transportdistanz, den CO₂-Ausstoß und die Gesamtkosten zwischen dem Einzelhändler und Produzenten, ohne Zwischenstation des Großhändlers.

Da die Spaltenbezeichnungen zur Zusammenführung identisch sein müssen, wird bei einer direkten Transportrouten „keiner“ in der Spalte des Großhändlers eingetragen. Durch UNION ALL und CALL werden die Abfragen zusammengeführt und in einer Tabelle zusammengefasst. Die Ausgabe erfolgt nach Transportroute, Großhändler, Gesamtdistanz, CO₂-Ausstoß und Gesamtkosten. Abschließend wird die Ausgabe durch ORDER BY nach aufsteigenden Gesamtkosten sortiert.

Algorithmus 5.4: Code zur Bestimmung optimaler Transportrouten

```

1 CALL { MATCH (p:Produzent),(e:Einzelhändler),(p:Produzent)-[li:
    liefert]->(g:Großhändler),(t:Transportart {TransportID: 1})
2 WHERE e.EinzelhändlerID = 2
3 WITH p, e, g, li, t, (2 * 6371 * asin(sqrt(haversin(radians(
    toInteger(g.lat) - toInteger(e.lat)))) + cos(radians(g.lat))
    *
4 cos(radians(e.lat))* haversin(radians(g.lon - e.lon)))) AS
    Gesamtdistanz
5 WITH p, g,e, toInteger(Gesamtdistanz) AS Gesamtdistanz,
    toInteger((Gesamtdistanz*t.Transportkosten)*0.9+ t.
```

```

        Transportfixkosten*0.9+ g.Kosten) AS Gesamtkosten, g.Name
        AS Großhändler
6  RETURN e.EinzelhändlerID AS Lieferverbindung, Großhändler AS
        Großhändler, Gesamtdistanz AS Gesamtdistanz, Gesamtkosten
        AS Gesamtkosten
7  UNION ALL
8  MATCH (p:Produzent),(e:Einzelhändler), (t:Transportart {
        TransportID: 1})
9  WHERE e.EinzelhändlerID = 2
10 WITH p, e, t, (2 * 6371 * asin(sqrt(haversin(radians(
        toInteger(p.lat) - toInteger(e.lat)))) + cos(radians(p.lat))
        *
11 cos(radians(e.lat))* haversin(radians(p.lon - e.lon)))) AS
        Gesamtdistanz
12 WITH p,e,t, tointeger(Gesamtdistanz) AS Gesamtdistanz,
        tointeger(Gesamtdistanz*t.Transportkosten+ t.
        Transportfixkosten) AS Gesamtkosten, "keiner" AS Großhändler
13 RETURN e.EinzelhändlerID AS Lieferverbindung, Großhändler AS
        Großhändler, Gesamtdistanz AS Gesamtdistanz, Gesamtkosten
        AS Gesamtkosten
14 }
15
16 RETURN Lieferverbindung, Großhändler, Gesamtdistanz,
        Gesamtkosten
17 ORDER BY Gesamtkosten

```

Die geringsten Gesamtkosten für die erste Transportroute weist in [Abbildung 5.4](#) eine direkte Verbindung zwischen dem Produzenten und dem Einzelhändler auf. Auch die Gesamtdistanz sowie der CO₂-Ausstoß sind für diese Transportroute am geringsten, weshalb eine Transportroute zwischen dem Produzenten und dem Einzelhändler EZ1 erstellt wird. Nach der Festlegung der ersten Transportroute erfolgt die Festlegung der zweiten Route.

Transportroute	Großhändler	Gesamtdistanz	CO ₂ Ausstoß	Gesamtkosten
1	"keiner"	47	5313	605
1	"DZ1"	591	59660	628
1	"DZ2"	728	73423	643

Abbildung 5.4: Ausgabe zur Bestimmung der günstigsten Transportrouten 1

Das Ergebnis der zweiten Transportroute zeigt, dass die Gesamtdistanz nicht ausschlaggebend für die Wahl der Transportroute ist. So ist eine direkte Lieferung um 50 km kürzer als über einen Großhändler, trotzdem ist eine Lieferung über den Großhändler DZ2 günstiger als eine direkte Lieferung. Durch die umweltfreundlichen Fahrzeuge der Großhändler liegt der CO₂-Ausstoß beider Transportrouten über einen Großhändler auf einem ähnlich hohen Niveau, wobei eine direkte Lieferung einen höheren CO₂-Ausstoß aufweist.

Transportroute	Großhändler	Gesamtdistanz	CO2Ausstoß	Gesamtkosten
2	"DZ2"	519	52372	621
2	"DZ1"	564	56852	625
2	"keiner"	466	52292	656

Abbildung 5.5: Ausgabe zur Bestimmung der günstigsten Transportrouten 2

Nach dem zuvor beschriebenen Prinzip werden die restlichen Transportrouten bestimmt, wobei eine Automatisierung der Abfrage an die IT-Abteilung des Unternehmens in Auftrag gegeben wird, da Neo4j eine Einbindung von Algorithmen ermöglicht (vgl. Abschnitt 3.3.3). Nachdem alle Transportrouten gebildet wurden, wird die fertige SC erneut visualisiert und angeordnet.

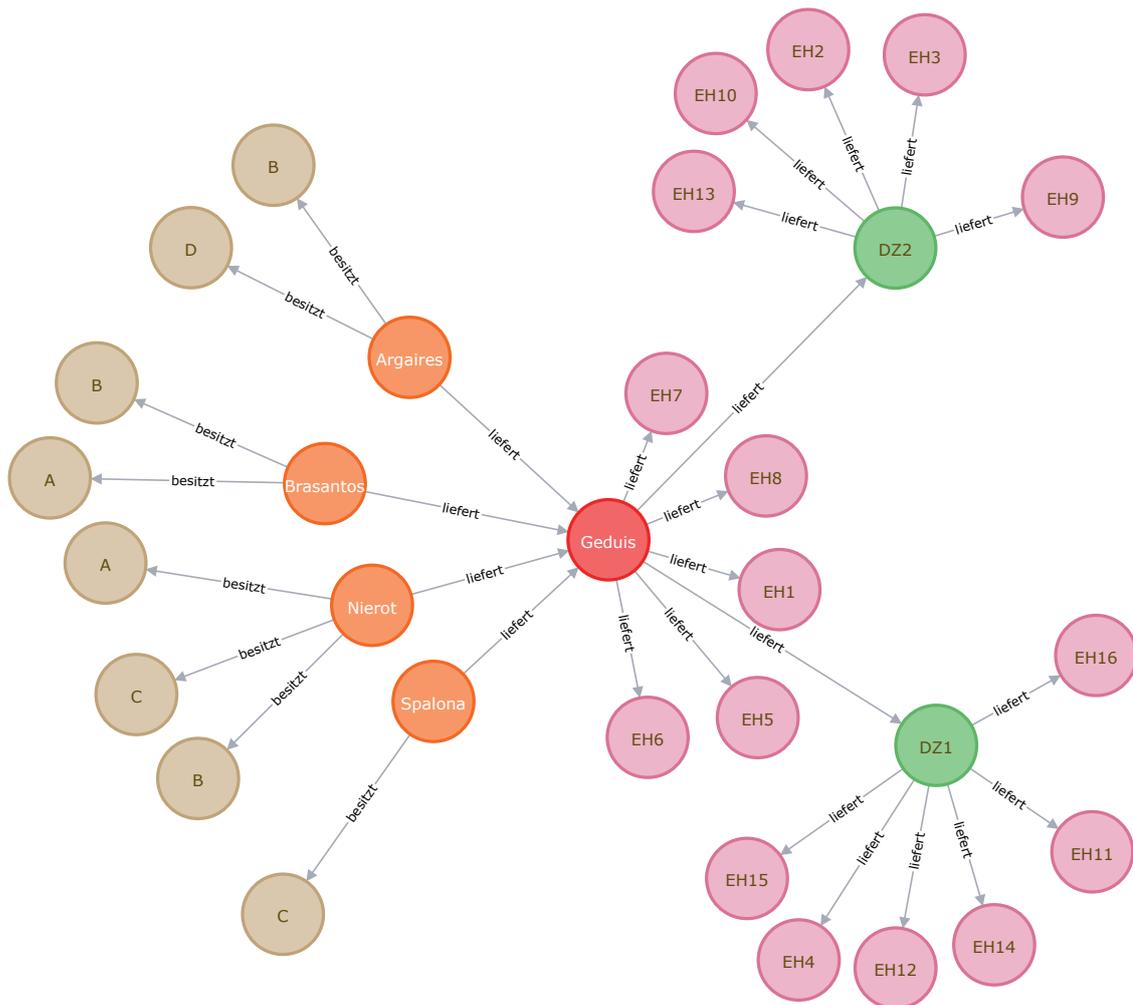


Abbildung 5.6: Visuelle Darstellung der fertigen SC

2. Welcher Lieferant bietet den Rohstoff A an?

Nach der Festlegung der Liefergebiete wird eine Datenbankabfrage in [Algorithmus 5.5](#) zur Suche eines bestimmten Rohstoffs durchgeführt. Diese gestaltet sich wie in [Abschnitt 3.3.3](#) erläutert: durch MATCH wird das Suchmuster „Lieferant besitzt Rohstoff“ beschrieben. Durch WHERE wird die Bedingung, ein Rohstoff mit der Bezeichnung „A“, zur Filterung der Ergebnisse gesetzt. Mit RETURN werden Knoten, Beziehungen, Pfade oder Eigenschaften zurückgegeben.

Algorithmus 5.5: Code zur Abfrage eines bestimmten Lieferanten

```

1 MATCH (l:Lieferant)-[:besitzt]->(r:Rohstoff)
2 WHERE r.Name="A"
3 RETURN l.Name AS Lieferantennamen,r.Name AS Rohstoffbezeichnung,r
   .Preis AS Preis

```

Die erste Abfrage gibt die Ergebnisse sortiert nach Lieferantennamen, Rohstoffbezeichnung und Preis zurück. Durch den Befehl ORDER BY werden die Lieferanten nach aufsteigendem Preis aufgelistet. Hierbei ist der Lieferant „Brasantos“ mit dem günstigsten Preis von 1200 Euro pro Tonne zu bevorzugen.

Lieferantennamen	Rohstoffbezeichnung	Preis
"Brasantos"	"A"	1200.0
"Nierot"	"A"	3500.0

Abbildung 5.7: Ausgabe zur Abfrage eines bestimmten Lieferanten

3. Wie lang ist der kürzeste Pfad über die gesamte SC, wenn der Produzent über einen Großhändler liefert?

Zur Suche eines kürzesten Pfades wird über den Befehl MATCH ein Pfad über alle Akteure der SC in [Algorithmus 5.6](#) definiert. Der Pfad durchläuft alle Lieferkanten. Durch den Befehl REDUCE wird die Liste von Lieferkanten des Pfades durchlaufen. Hierbei werden die Distanzen auf die Startdistanz 0 addiert und als Gesamtdistanz ausgegeben. Anschließend werden die Gesamtdistanzen sortiert. Durch den Befehl LIMIT lassen sich die Ergebnisse auf eine vordefinierte Anzahl beschränken.

Algorithmus 5.6: Code zur Abfrage des kürzesten Pfades

```

1 MATCH pfad = (l:Lieferant)-[li:liefert]->(p:Produzent)-[li2:
   liefert]->(g:Großhändler)-[li3:liefert]->(e:Einzelhändler)
2 WITH reduce (dist = 0, r in relationships(pfad)| dist +r.distanz
   ) AS Gesamtdistanz, pfad
3 RETURN [x in nodes(pfad)|x.Name] AS Pfad, Round(Gesamtdistanz)
   AS Gesamtdistanz
4 ORDER BY Gesamtdistanz
5 LIMIT 1

```

Als Ergebnis werden die gesamten Akteure des Pfades und die Gesamtdistanz ausgegeben. Dazu werden die Distanzen in [km] aufsteigend nach der geringsten Gesamtdistanz sortiert. Mit 470 km ist der Pfad über den Lieferanten „Nierot“, den Produzenten „Geduis“, dem Großhändler DZ1 bis hin zum Einzelhändler EH12 am kürzesten.

Pfad	Gesamtdistanz
["Nierot", "Geduis", "DZ1", "EH12"]	470.0

Abbildung 5.8: Ausgabe zur Abfrage des kürzesten Pfades

4. Lassen sich auch umweltfreundlichste, schnellste oder kostengünstigste Pfade über die gesamte SC bestimmen?

Aufgrund der unterschiedlichen Transportarten sind die Dauer und Kosten pro Transportroute variabel, weshalb die Eigenschaften der Wege nicht als feste Werte vorliegen. Jedoch soll die grundsätzliche Durchführbarkeit von Pfadabfragen getestet werden, so dass die SCM-Abteilung Zufallswerte für die Dauer, Kosten und den CO₂-Ausstoß der Transportrouten bestimmt. Hierbei werden neue Attribute durch FOREACH und SET für alle existierenden Transportrouten bestimmt. [Algorithmus 5.7](#) zeigt den Code anhand der Bestimmung von CO₂-Werten.

Algorithmus 5.7: Code zur Bestimmung von Zufallszahlen

```

1 MATCH (m) -[li:liefert] ->(n)
2 FOREACH(r IN RANGE(0,22) | SET li.CO2= rand ()*100)
3 RETURN li

```

Anschließend erfolgt die Abfrage analog zur dritten Fragestellung, wobei die Gesamtdistanz, wie in [Algorithmus 5.6](#) beschrieben, durch Zeit, Kosten oder CO₂ ersetzt werden kann. Nachfolgend wird die Ausgabe für die Abfrage nach der umweltfreundlichsten Route angezeigt. Hierbei ist der Pfad über den Lieferanten „Nierot“, den Produzenten „Geduis“, dem Großhändler DZ2 bis hin zum Einzelhändler EH2 mit 122 Gramm CO₂ am umweltfreundlichsten.

Pfad	CO ₂ Ausstoß
["Nierot", "Geduis", "DZ2", "EH2"]	122.0

Abbildung 5.9: Ausgabe des umweltfreundlichsten Pfades

Wie [Abbildung 5.9](#) zeigt, sind Pfadabfragen in der Graphdatenbank Neo4j unkompliziert umzusetzen und lassen sich anwenderabhängig anpassen. Da die Eigenschaften von Knoten und Kanten jedoch aufsummiert werden, müssen die Knoten- und Kantengewichte zuvor festgelegt werden. Auf dieses Fallbeispiel angewendet, könnten mehrere Lieferwege, bspw. Seewege, mit unterschiedlichen Eigenschaften vordefiniert werden. Hierbei würde die Visualisierung der SC durch mehrere Transportverbindungen zwischen denselben Akteuren jedoch schnell unübersichtlich werden.

5. Welcher Lieferant und welche Lieferart sollten für den Rohstoff B unter Berücksichtigung der Kosten gewählt werden?

Zur Suche des günstigsten Lieferanten unter Berücksichtigung der Gesamtkosten wird eine Datenbankabfrage in [Algorithmus 5.8](#) geschrieben, die eine Restriktion auf den betrachteten Rohstoff setzt und die Gesamtkosten berechnet. Hierbei bilden sich die Gesamtkosten

aus der Gesamtdistanz und den entstehenden Transportkosten und Transportfixkosten. Aus Vergleichsgründen wird keine Begrenzung der Ergebnisse gesetzt.

Algorithmus 5.8: Code zur Abfrage bestimmter Lieferanten

```

1 MATCH (t:Transportart)<-[u:unterstützt]-(l:Lieferant)-[li:liefert]->(p:Produzent), (l:Lieferant)-[:besitzt]->(r:Rohstoff)
2 WHERE r.Name = "B"
3 WITH toInteger(li.distanz*t.Transportkosten + t.Transportfixkosten+r.Preis) AS Gesamtkosten, l, t
4 RETURN l.Name AS Lieferantennamen, t.Transportart AS Transportart, Gesamtkosten
5 ORDER BY Gesamtkosten

```

Als Ergebnis wird eine Übersicht der Lieferanten, der Transportart sowie der Gesamtkosten ausgegeben. Analog zu den Gesamtkosten lassen sich die Gesamtdistanz, die Dauer oder der CO₂-Ausstoß berechnen. In diesem Beispiel ist der Lieferant „Argaires“ mit der Transportart „Schiff“ mit Gesamtkosten von 3554 Euro pro Tonne des Rohstoffes B zu bevorzugen.

Lieferantennamen	Transportart	Gesamtkosten
"Argaires"	"Schiff"	3554
"Brasantos"	"Schiff"	3661
"Nierot"	"LKW"	4359
"Nierot"	"Bahn"	4799
"Nierot"	"Schiff"	4944
"Brasantos"	"Flug"	5285
"Argaires"	"Flug"	5474

Abbildung 5.10: Ausgabe bestimmter Lieferanten

6. In welcher Reihenfolge sollte der Produzent die Einzelhändler beliefern, um die kostengünstigste Route zu wählen?

Zuletzt soll eine Route zwischen dem Produzenten und den zu beliefernden Einzelhändler berechnet werden. Da die Version Neo4j 4.0.3 die vordefinierten Algorithmen der Vorgängerversionen zur Durchführung des Fallbeispiels (01.06.2020) nicht einbinden kann, erfolgt die Berechnung durch manuelle Abfragen (Neo4j 2020b). Zuerst wird in Algorithmus 5.9 jedem Einzelhändler ein Wahrheitswert „Lieferstatus“ zugeordnet. Wird der Einzelhändler nicht beliefert, ist der Wahrheitswert nicht erfüllt. So wird in der nachfolgenden Abfrage gewährleistet, dass nur Einzelhändler betrachtet werden, die noch nicht beliefert wurden. Anschließend werden die Distanzen berechnet. Ergebnis des Algorithmus 5.9 ist die Ausgabe des Start- und Endpunktes sowie die Gesamtdistanz. Nach Abfrage des kürzesten Händlers wird in Algorithmus 5.10 eine Routenverbindung mit dem Attribut „Distanz“

erstellt und der Lieferstatus des belieferten Einzelhändlers auf „true“ geändert. Dies verhindert ein erneutes Anfahren des Händlers.

Algorithmus 5.9: Code zur Abfrage der kürzesten Distanz

```

1 MATCH (p:Produzent),(e:Einzelhändler)
2 WHERE e.beliefert =false
3 WITH p, e, (2 * 6371 * asin(sqrt(haversin(radians(toInteger(p.
  lat) - toInteger(e.lat)))) + cos(radians(p.lat))*
4 cos(radians(e.lat))* haversin(radians(p.lon - e.lon)))) AS
  Gesamtdistanz
5 RETURN p.Name AS ANFANG, e.Name AS ENDE, Gesamtdistanz AS
  Gesamtdistanz
6 ORDER BY Gesamtdistanz
7 LIMIT 1

```

Algorithmus 5.10: Code zur Erstellung einer Routenverbindung

```

1 MATCH (a:Einzelhändler {EinzelhändlerID: 7}),(e:Einzelhändler {‘
  EinzelhändlerID’: 5})
2 MERGE (a)-[1:beliefert {distanz: 1 }]->(e)
3 SET e.beliefert = TRUE

```

Zur Bestimmung der weiteren Route müssen die Start- und Endknoten durch die Angabe der Identifikationsnummern der jeweiligen Einzelhändler spezifiziert werden. Zur Visualisierung wird die fertige Route durch Abfrage der Beziehung „beliefert“ ausgegeben. Hierbei wurde zwischen den Beziehungen „liefert“ und den Beziehungen „beliefert“ unterschieden. Durch die Visualisierung der Route ohne zusätzliche Darstellung der Transportrouten „liefert“ erhöht sich die Nachvollziehbarkeit der Ergebnisse für den Anwender.

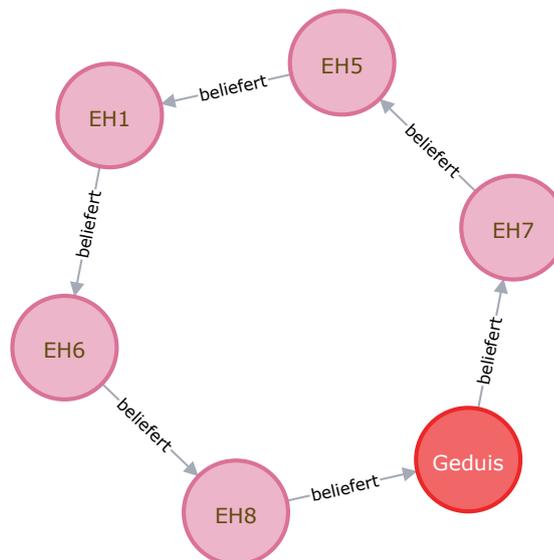


Abbildung 5.11: Ausgabe der fertigen Route

Abschließend erfolgt die Abfrage der Gesamtdistanz der Route. Hierbei kann die Gesamtdistanz auf Basis des [Algorithmus 5.6](#) oder durch ein Aufsummieren der Distanzwerte bestimmt werden. Durch [Algorithmus 5.11](#) wird die Gesamtdistanz von 457 km berechnet.

Algorithmus 5.11: Code zur Berechnung der Gesamtdistanz

```

1 MATCH (n)-[li:beliefert]->(m)
2 WITH SUM(li.distanz) AS Gesamtdistanz
3 RETURN Gesamtdistanz

```

Die einzelnen Abfrageblöcke zur Bestimmung der kürzesten Strecke sind einfach gestaltet und ändern sich lediglich in den betrachteten Identifikationsnummern. Durch die wiederholte Abfrage jeder einzelnen Verbindung erhöht sich der Aufwand zur Bestimmung einer Route. Jedoch existieren für vergangene Neo4j-Versionen vordefinierte Algorithmen zur Berechnung von kürzesten Wegen, so dass sich der Aufwand zur Bestimmung einer Route verringern könnte (Neo4j 2020b). Auch bietet Neo4j die Programmierung eigener Algorithmen und Funktionen, so dass die zuvor durchgeführte Abfrage in Java programmiert werden könnte (vgl. Abschnitt 3.3.3). Nach der Durchführung der Fallbeispiele in Abschnitt 5.1 und Abschnitt 5.2 werden die Ergebnisse der bearbeiteten Fallbeispiele beurteilt und zusammengefasst.

5.3 Fazit

Fallbeispiel 1 dient zur Validierung der in Abschnitt 4.1.1 erstellten Taxonomie. Hierfür wurden der Literatur beispielhafte Fragestellungen entnommen und in die Taxonomie eingeordnet. Hierbei zeigt sich, dass sich ein Großteil der Fragestellungen direkt oder durch eine Erweiterung in die Taxonomie einordnen lassen. Jedoch lassen sich Entscheidungsfragen nicht durch die Taxonomie abdecken. Dies begründet sich durch das SCOR-Modell, das die Grundlage für das SCM-Aufgabenmodell in Abbildung 2.2 und die Taxonomie in Abbildung 4.1 darstellt (vgl. Abschnitt 2.2.2). Das SCOR-Modell gibt eine Übersicht über unternehmensinterne und -übergreifende Prozesse sowie Aufgabenbereiche und zielt auf die Standardisierung, Beschreibung und Bewertung von SC ab (vgl. Abschnitt 2.2.2). So werden ereignisauslösende Entscheidungen und damit einhergehende Entscheidungsfragen nicht betrachtet. Zur Erweiterung der Taxonomie zu einem Entscheidungsmodell könnten, wie in Abschnitt 5.1 beschrieben, Freigabe- oder Entscheidungspunkte ergänzt werden.

Weiterhin benötigt der Anwender zur Einordnung der betrachteten Fragestellungen das Wissen über die Art der Fragestellung und wie sie zu lösen ist. Ansonsten könnte eine Einordnung der Fragestellung in eine falsche Klasse mit ähnlichen Lösungsansätzen erfolgen. Die Charakterisierung der Klassen anhand von Lösungsansätzen begründet sich mit der Zielsetzung: graphbasierte Lösungsansätze zur Lösung von logistischen Fragestellungen des SCM herzuleiten. Hierfür müssen vorhandene Lösungsansätze identifiziert werden, so dass eine Charakterisierung anhand anderer Merkmale für die Zielsetzung nicht notwendig ist. Auch wurden die Lösungsansätze in der Taxonomie weniger ausführlich als die graphbasierten Lösungsansätze behandelt. Dies lässt sich ebenfalls durch die Zielsetzung begründen. So stellt die Erstellung der Taxonomie lediglich ein unterstützendes Teilziel dar (vgl. Kapitel 1). Ein höherer Detaillierungsgrad der Lösungsansätze aus Abbildung 4.1 würde deshalb eine Verschiebung der Zielsetzung bedeuten.

Positiv zu bewerten ist die Flexibilität der Taxonomie, da sie auch für Erweiterungen der Klassen konzipiert wurde. Eine Validierung hierfür bietet das erste Fallbeispiel aus Abschnitt 4.1, das die Taxonomie durch den Aufgabenbereich der Produktionsplanung ergänzt. Insgesamt eignet sich die Taxonomie somit zur Einordnung logistischer Fragestellungen, wobei Entscheidungsfragen durch entsprechende Erweiterung der Taxonomie zu

einem Entscheidungsmodell ergänzt werden müssen. Durch die Erweiterung von Ebenen und Klassen lässt sich die Taxonomie zusammenfassend flexibel im SCM einsetzen.

Fallbeispiel 2 untersucht graphbasierte Lösungsansätze zur Beantwortung beispielhafter Fragestellungen der Transportplanung. Zur Untersuchung wurden die Fragestellungen, angelehnt an [Abschnitt 2.3.2](#), mithilfe der Graphdatenbank Neo4j beantwortet. Insgesamt konnten alle Fragestellungen durch Pfad- und Musterabfragen gelöst werden, wobei der Aufwand zur Beantwortung der Fragestellungen unterschiedlich hoch ausfiel. So wurden die Fragen nach dem kürzesten/ günstigsten/ umweltfreundlichsten Weg durch Pfadabfragen mit kompakten Abfrageblöcken beantwortet (vgl. [Algorithmus 5.6](#)). Auch die Frage nach bestimmten Produkten unter Berücksichtigung von Lieferanten und Lieferarten kann durch Musterabfragen gut beantwortet werden (vgl. [Algorithmus 5.8](#)).

Der Aufwand zur Bestimmung einer Route, ein elementarer Bestandteil der Transportplanung (vgl. [Abschnitt 2.3.2](#)), ist durch die rekursive Abfrage und Manipulation nach den kürzesten Wegen zwischen den SC-Akteuren hoch (vgl. [Algorithmus 5.9](#)). Dies lässt sich jedoch durch die eingesetzte Version 4.0.3 von Neo4j begründen, die im Gegensatz zu früheren Versionen noch keine Unterstützung der existierenden Algorithmen-Bibliothek bietet ([Neo4j 2020b](#)). Auch wurde die Implementierung selbst erstellter Algorithmen mithilfe der Programmiersprache Java nicht berücksichtigt. So könnte sich der Aufwand zur Tourenplanung durch einen vordefinierten Algorithmus verringern, da die einzelnen Datenbankabfragen und -manipulationen sich nur durch Variation der betrachteten Knoten unterscheiden. Auch die Fertigstellung der SC durch Bestimmung von Transportrouten zwischen Einzelhändlern, Großhändlern und dem Produzenten ist durch die rekursive Abfrage von Transportverbindungen aufwändig (vgl. [Algorithmus 5.9](#)). Hier würde sich ebenfalls der Aufwand durch vordefinierte Algorithmen verringern.

Im zweiten Fallbeispiel wurden nur Fragestellungen der Transportplanung untersucht, so dass die identifizierten Lösungsansätze in anderen Aufgabenstellungen nicht validiert wurden. Des Weiteren wurden für die Beantwortung von Fragestellungen der Transportplanung keine Graph-Mining-Algorithmen untersucht, da die Lösungsansätze aus [Abschnitt 4.2.2](#) größtenteils auf graphbasierte Modelle oder Graph-Management-Algorithmen basieren. So sind die Lösungsansätze von Graph-Mining-Algorithmen weiterhin nicht-validiert, wobei die Graphdatenbank Neo4j Graph-Mining-Algorithmen, bspw. Link-Prediction, anbietet (vgl. [Neo4j 2020b](#), S. 1).

Insgesamt zeigt das Fallbeispiel 2, dass sich Fragestellungen aus der Transportplanung gut auf Basis einer Graphenrepräsentation, in Fallbeispiel 2 mithilfe einer Graphdatenbank realisiert, beantworten lassen. Insbesondere Pfadabfragen über mehrere Entitäten hinweg lassen sich durch graphbasierte Abfragen problemlos beantworten. So werden drei Befehle zur grundlegenden Pfadabfrage benötigt (vgl. [Algorithmus 5.6](#)). Dies entspricht der Länge eines grundlegenden Abfrageblocks (vgl. [Abschnitt 3.3.3](#)). Auch Musterabfragen sind für eine Vielzahl der Fragestellungen ausreichend, wobei die Implementierung von selbst erstellten Algorithmen die Beantwortung einiger Fragestellung vereinfachen würde.

Nach der Durchführung der zwei Fallbeispiele in [Abschnitt 5.1](#) und [Abschnitt 5.2](#) sowie ihrer Beurteilung durch das Fazit in [Abschnitt 5.3](#) erfolgt in [Kapitel 6](#) eine Zusammenfassung wissenschaftlichen Grundlagen aus [Kapitel 1](#) und [Kapitel 2](#) sowie der Ergebnisse aus [Kapitel 4](#) und [Kapitel 5](#). Hierbei erfolgt ein Abgleich und eine Zusammenfassung der bearbeiteten Aufgabenbereiche hinsichtlich der in [Kapitel 1](#) definierten Ziele sowie ein Ausblick in weitere Forschungsfelder.

6 Zusammenfassung und Ausblick

Das Hauptziel dieser Arbeit ist die Untersuchung graphbasierter Lösungsansätze zur Beantwortung logistischer Fragestellungen. Zur Zielerreichung wurden eine Taxonomie zur Klassifizierung logistischer Fragestellungen erstellt und Kategorien, bestehend aus graphbasierten Modellen und Graphalgorithmen, gebildet. Hierfür wurden wissenstheoretische Grundlagen auf Basis einer Literaturrecherche zu Themen des SCM, DBS und graphbasierter Verfahren zusammengetragen.

Zu Beginn wurden Grundlagen des SCM in [Kapitel 2](#) betrachtet. So wurden logistische Begrifflichkeiten, u. a. die Logistik, SC und SCM, in [Abschnitt 2.1](#) und [Abschnitt 2.2](#) erläutert und definiert. Hierbei wurde die Vielschichtigkeit des SCM-Begriffs und der behandelten Aufgaben verdeutlicht, so dass eine Eingrenzung der Untersuchungsobjekte auf logistische Aufgaben des SCM vorgenommen wurde. Im Rahmen der Beschreibung logistischer Aufgaben wurden zunächst das SCOR- und das Aufgabenmodell in [Abschnitt 2.2.2](#) vorgestellt. Anschließend wurden in [Abschnitt 2.3](#) logistische Aufgaben, ihre Inhalte, beispielhafte Fragestellungen und vorhandene Lösungsansätze erläutert. Jedoch wurde keine Klassifizierungsmöglichkeit logistischer Fragestellungen auf Basis vorhandener Literatur entdeckt.

In [Kapitel 3](#) wurden Grundlagen der Datenhaltung mit Fokus auf GDBS und graphbasierter Verfahren erörtert. Zunächst wurden Daten und Informationssysteme im Zusammenhang mit dem SCM erläutert (vgl. [Abschnitt 3.1](#)). Anschließend wurden in [Abschnitt 3.2](#) und [Abschnitt 3.3](#) RDBS und GDBS vorgestellt. Das Kapitel schließt in [Abschnitt 3.4](#) mit der Beschreibung graphbasierter Modelle im SCM sowie Graphalgorithmen, wobei deutlich wurde, dass eine Kategorisierung der Algorithmen nicht existiert.

In [Kapitel 4](#) wurde das Hauptziel, die Untersuchung graphbasierter Lösungsansätze zur Beantwortung logistischer Fragestellungen, bearbeitet. In [Abschnitt 4.1.1](#) wurde durch die Einteilung von logistischen Aufgaben in Aufgabenbereiche und Aufgabenstellungen eine Taxonomie zur Klassifizierung logistischer Fragestellungen erstellt. So können die Fragestellungen in Aufgabenstellungen klassifiziert und durch bestehende Lösungsansätze der Aufgabenstellungen systematisch beantwortet werden. Durch die Kategorisierung und Beschreibung graphbasierter Verfahren in [Abschnitt 4.2](#) wurde die Grundlage zur systematischen Herleitung graphbasierter Lösungsansätze geschaffen. Anschließend wurden die Aufgabenstellungen der Taxonomie sowie die Unterkategorien in einer Bewertungsmatrix zusammengefasst und in [Abschnitt 4.2](#) untersucht.

Die Untersuchung in [Abschnitt 4.2](#) verdeutlicht, dass die graphbasierten Kategorien aus [Abschnitt 4.1.2](#) Lösungsansätze zur Beantwortung logistischer Fragestellungen aufweisen. Insgesamt wurden 12 Unterkategorien pro Aufgabenstellung analysiert, wobei in jeder der insgesamt 14 Aufgabenstellungen Lösungsansätze entdeckt wurden. Zusammenfassend bieten die Ergebnisse aus [Abschnitt 4.2.3](#) eine Übersicht graphbasierter Lösungsansätze und ihren Einsatz in logistischen Aufgabenstellungen. Es werden jedoch keine finalen Aussagen über die Güte oder Existenz von Lösungsansätzen der Kategorien getätigt, da auf-

grund der hohen Abstraktionsebene nicht alle Teilaspekte der Aufgabenstellungen aus [Abschnitt 4.1.1](#) erfasst wurden.

Validiert wurden die Ergebnisse durch zwei Fallbeispiele in [Kapitel 5](#). Im ersten Fallbeispiel wurden logistische Fragestellungen mithilfe der Taxonomie aus [Abbildung 4.1](#) klassifiziert. Hierbei konnten 10 von 12 Fragestellungen klassifiziert werden. Eine der zwei verbliebenen Fragestellungen konnte durch Erweiterung der Aufgabenbereiche ebenfalls nachträglich klassifiziert werden. Durch die letzte Fragestellung wurden Defizite der Taxonomie aufgedeckt und in [Abschnitt 5.3](#). In diesem Rahmen wurde deutlich, dass Entscheidungsfragen nicht beantwortet werden können und die Charakteristika der Klassen weiter spezifiziert werden müssen.

Das zweite Fallbeispiel hat graphbasierte Lösungsansätze in der Transportplanung anhand beispielhafte Fragestellungen validiert. Zur Validierung wurde eine Graphdatenbank genutzt, so dass die Fragestellungen durch Datenbankabfragen (Graph-Management-Algorithmen) beantwortet wurden. Jedoch wurde herausgearbeitet, dass eine Validierung der Ergebnisse aus [Kapitel 4](#) aufgrund der Fokussierung auf die Transportplanung und den Graph-Management-Algorithmen beschränkt möglich ist. Jedoch konnten alle Fragestellungen des betrachteten Fallbeispiels gelöst werden. Zusammenfassend konnten durch die Taxonomie und die systematische Untersuchung der graphbasierten Kategorien logistische Fragestellungen klassifiziert und unter Berücksichtigung einer Graphenrepräsentation größtenteils beantwortet werden. Durch die zuvor beschriebenen Ausführungen ergeben sich weitere Forschungsmöglichkeiten und offene Forschungsfragen, die nachfolgend erörtert werden.

Zur Identifizierung weiterer graphbasierter Lösungsansätze sollten logistische Aufgabenstellungen tiefergehend untersucht und beschrieben werden. Insbesondere auf der Gestaltungsebene könnten durch die Spezifizierung der Aufgabenbereiche und -stellungen weitere Lösungsansätze aufgedeckt werden. Weiterhin könnten die Merkmale der Klassen modifiziert werden, da sie in dieser Arbeit durch Lösungsansätze im SCM charakterisiert werden. So könnten Merkmale durch Analyse der Aufgabenstellungen, bspw. Planungshorizont oder Komplexität, gebildet werden. Auch eine Ausgestaltung der Taxonomie durch Erweiterung auf nicht-logistische Aufgaben des SCM könnte erforscht werden. Weiterhin sollten bestehende SCM-Modelle hinsichtlich ihrer Einteilungen und Beschreibungen von SC und entstehender SCM-Aufgaben überprüft werden.

Neben der Taxonomie sollten auch die in [Abschnitt 4.1.2](#) vorgestellten Kategorien weiter erforscht werden. So existiert noch keine klar definierte Abgrenzung von Graph-Management- und Graph-Mining-Algorithmen. Auch werden graphbasierte Modelle des SCM in der Literatur nicht mit Graphalgorithmen in Verbindung gebracht. Deshalb sollten graphbasierte Kategorien charakterisiert, erneut kategorisiert, auf Vollständigkeit überprüft und definiert werden.

Insbesondere für Graphalgorithmen existiert keine verbreitete Kategorisierung. Zur Herleitung der Kategorisierung könnten bestehende Ansätze, bspw. nach [Aggarwal \(2010\)](#), analysiert und erweitert werden. Weiterhin sollten Begrifflichkeiten und Eigenschaften von Graph-Management- und Graph-Mining-Algorithmen erforscht werden. So könnten z. B. Eigenschaften von Graph-Mining-Algorithmen hinsichtlich der Eigenschaften von Data-Mining-Algorithmen verglichen werden.

Im Gegensatz zu Graph-Mining-Algorithmen weisen Graph-Management-Algorithmen Anknüpfungspunkte an Modelle des SCM auf. Da graphbasierte Modelle im SCM bereits durch die Literatur validiert sind, könnte eine Implementierung der Modelle durch Graph-

Management-Algorithmen erforscht werden. Jedoch ergeben sich durch Graph-Mining-Algorithmen Lösungsansätze, die trotz fehlender Anknüpfungspunkte zu dem Themengebiet des SCM, erforscht werden sollten.

Um alle Forschungsergebnisse aus [Kapitel 4](#) zu validieren, sollte jede Aufgabenstellung, wie in Fallbeispiel 2, mitsamt der identifizierten Lösungsansätze durch eine Form der Graphenrepräsentation, bspw. einer Graphdatenbank, implementiert und untersucht werden. Hierbei sollten nicht nur Graph-Management-Algorithmen, sondern auch Graph-Mining-Algorithmen berücksichtigt werden. So eignet sich z. B. die Untersuchung und der Vergleich angebotener Graph-Mining-Algorithmen in verschiedenen Datenbanken. Insgesamt ergibt sich aus den Forschungsergebnissen die Notwendigkeit der tiefergehenden Betrachtung graphbasierter Lösungsansätze zur Beantwortung logistischer Fragestellungen des SCM.

Literaturverzeichnis

- Abdelhamid, E.; Canim, M.; Sadoghi, M.; Bhattacharjee, B.; Chang, Y.-C.; Kalnis, P.: Incremental Frequent Subgraph Mining on Large Evolving Graphs. *IEEE Transactions on Knowledge and Data Engineering* 29 (2017) 12, S. 2710–2723.
- Aggarwal, C. C. (Hg.): *Managing and Mining Graph Data*. Bd. 40. *Advances in Database Systems*. New York [u.a.]: Springer, 2010.
- Aggarwal, C. C.; Wang, H.: A Survey of Clustering Algorithms for Graph Data. In: Aggarwal, C. C. (Hg.): *Managing and Mining Graph Data*. *Advances in Database Systems*. New York [u.a.]: Springer, 2010a, S. 275–302.
- Aggarwal, C. C.; Wang, H.: An Introduction to Graph Data. In: Aggarwal, C. C. (Hg.): *Managing and Mining Graph Data*. Bd. 40. *Advances in Database Systems*. New York [u.a.]: Springer, 2010b, S. 1–11.
- Aggarwal, C. C.; Wang, H.: Graph Data Management and Mining: A Survey of Algorithms and Applications. In: Aggarwal, C. C. (Hg.): *Managing and Mining Graph Data*. *Advances in Database Systems*. New York [u.a.]: Springer, 2010c, S. 13–68.
- Aigner, M.: *Graphentheorie*. Wiesbaden: Springer Fachmedien Wiesbaden, 2015.
- Albrecht, M.; Rohde, J.; Wagner, M.: Master Planning. In: Stadtler, H.; Kilger, C.; Meyr, H. (Hg.): *Supply Chain Management and Advanced Planning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, S. 156–176.
- Anane, R.; Younas, M.; Tsai, C.-F.; Chao, K.-M.: Agent-based transactional framework for the supply chain. In: *Proceedings. International Conference on Machine Learning and Cybernetics*. IEEE, 4-5 Nov. 2002, S. 1956–1961.
- Angles, R.; Gutierrez, C.: Survey of graph database models. *ACM Computing Surveys* 40 (2008) 1, S. 1–39.
- Angles, R.; Gutierrez, C.: An Introduction to Graph Data Management. In: Fletcher, G.; Hidders, J.; Larriba-Pey, J. L. (Hg.): *Graph Data Management*. Cham: Springer International Publishing, 2018, S. 1–32.
- Arndt, H.: *Supply Chain Management: Optimierung logistischer Prozesse*. 4. Wiesbaden: Gabler Verlag / GWV Fachverlage GmbH Wiesbaden, 2008.
- Baumgarten, H.: Trends der Logistik. In: Baumgarten, H. (Hg.): *Supply Chain Steuerung und Services*. Engineering online library. Berlin: Springer, 2004, S. 1–11.
- Baumgarten, H.: *Das Beste der Logistik*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
- Bissantz, N.; Hagedorn, J.: Data Mining (Datenmustererkennung). *Wirtschaftsinformatik* 51 (2009) 1, S. 139–144.
- Bodendorf, F.: *Daten- und Wissensmanagement*. 2., aktualisierte und erweiterte Auflage. Springer-Lehrbuch. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2006.
- Bousonville, T.: *Logistik 4.0*. Wiesbaden: Springer Fachmedien Wiesbaden, 2017.
- Bühler, G.: *Verkehrsmittelwahl im Güterverkehr*. 1. Aufl. Umwelt- und Ressourcenökonomie. s.l.: Physica-Verlag, 2006.
- Bunke, H.; Neuhaus, M.: Graph Matching—Exact and Error-Tolerant Methods and Automatic Learning of Edit Costs. In: Cook, D. J.; Holder, L. B. (Hg.): *Mining graph data*. Hoboken, N.J: Wiley-Interscience, 2007, S. 17–34.

- Busch, A.; Dangelmaier, W.: Einführung in die Integration unternehmensübergreifender Logistikprozesse. In: Busch, A.; Dangelmaier, W. (Hg.): Integriertes Supply Chain Management. 2. Auflage. Wiesbaden: Gabler, 2004a, S. 1–21.
- Busch, A.; Dangelmaier, W. (Hg.): Integriertes Supply Chain Management: Theorie und Praxis effektiver unternehmensübergreifender Geschäftsprozesse. 2. Auflage. Busch, Axel (editor.) Dangelmaier, Wilhelm (editor.) Wiesbaden: Gabler, 2004b.
- Busch, A.; Dangelmaier, W.; Pape, U.; Rüter, M.: Differenzierungsmerkmale von SCM-Systemen. In: Busch, A.; Dangelmaier, W.; Pape, U.; Rüter, M. (Hg.): Marktspiegel Supply Chain Management Systeme. Wiesbaden: Gabler Verlag, 2003, S. 37–48.
- Chakrabarti, D.: Graph Mining. In: Sammut, C.; Webb, G. I. (Hg.): Encyclopedia of Machine Learning. Boston, MA: Springer US, 2010, S. 469–471.
- Chopra, S.; Meindl, P.: Supply Chain Management: Strategie, Planung und Umsetzung. 5., aktualisierte Auflage. Always learning. Chopra, Sunil (VerfasserIn) Meindl, Peter (VerfasserIn) Kummer, Sebastian (Sonstige Person, Familie und Körperschaft) Groschopf, Wolfram (Sonstige Person, Familie und Körperschaft) Gahm, Christian (Sonstige Person, Familie und Körperschaft) Sahamie, Ramin (Sonstige Person, Familie und Körperschaft). Hallbergmoos: Pearson, 2014.
- Christopher, M.: Logistics and supply chain management: Creating value-adding networks. 3. ed., [Nachdr.] Harlow: Financial Times/Prentice Hall, 2005.
- Christopher, M.: Logistics & supply chain management. fifth edition. Always learning. Christopher, Martin (VerfasserIn). Harlow, England und New York: Pearson Education, 2016.
- Cleve, J.; Lämmel, U.: Data Mining. München: De Gruyter Oldenbourg, 2014.
- Cook, D. J.; Holder, L. B. (Hg.): Mining graph data. Hoboken, N.J.: Wiley-Interscience, 2007.
- Corsten, H.; Gössinger, R.: Einführung in das Supply Chain Management. 2., vollst. überarb. und wesentlich erw. Aufl. Lehr- und Handbücher der Betriebswirtschaftslehre. München: Oldenbourg, 2008.
- Council of Supply Chain Management Professionals: Supply Chain Management Terms and Glossary. 2013. URL: https://cscmp.org/CSCMP/Educate/SCM_Definitions_and_Glossary_of_Terms.aspx (zuletzt geprüft am 11.06.2020).
- Daskin, M. S.; Maass, K. L.: Location Analysis and Network Design. In: Zijm, H.; Klumpp, M.; Regattieri, A.; Heragu, S. (Hg.): Operations, Logistics and Supply Chain Management. Cham: Springer International Publishing, 2019, S. 379–398.
- Dippold, R.; Meier, A.; Schnider, W.; Schwinn, K.: Unternehmensweites Datenmanagement. Wiesbaden: Vieweg+Teubner Verlag, 2005.
- Dominguez-Sal, D.; Martinez-Bazan, N.; Mentes-Mulero, V.; Baleta, P.; Larriba-Pey, J. L.: A Discussion on the Design of Graph Database Benchmarks. In: Nambiar, R.; Poess, M. (Hg.): Performance Evaluation, Measurement and Characterization of Complex Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, S. 25–40.
- Dörn, S.: Programmieren für Ingenieure und Naturwissenschaftler. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017.
- Drexl, A.; Domschke, W.: Logistik: Standorte. 4., überarbeitete und erweiterte Auflage. Oldenbourgs Lehr- und Handbücher der Wirtschafts- u. Sozialwissenschaften. München: Oldenbourg, 1996.
- Düsing, R.: Knowledge Discovery in Databases - Begriffe, Forschungsgebiete, Prozesse und Systeme. In: Chamoni, P.; Gluchowski, P. (Hg.): Analytische Informationssysteme. Berlin Heidelberg: Springer-Verlag GmbH, 2006, S. 241–262.

- Eades, P.; Klein, K.: Graph Visualization. In: Fletcher, G.; Hidders, J.; Larriba-Pey, J. L. (Hg.): Graph Data Management. Cham: Springer International Publishing, 2018, S. 33–70.
- Engelmann, F.; Großmann, C.: Was wissen wir über Informationen? In: Hildebrand, K.; Gebauer, M.; Hinrichs, H.; Mielke, M. (Hg.): Daten- und Informationsqualität. 3. Auflage. Wiesbaden: Springer Vieweg, 2015, S. 3–24.
- DB-Engines: DB-Engines Ranking - popularity ranking of graph DBMS. 2020. URL: <https://db-engines.com/en/system/Neo4j> (zuletzt geprüft am 10.06.2020).
- Eßig, M.; Hofmann, E.; Stölzle, W.: Supply chain management. Vahlens Handbücher der Wirtschafts- und Sozialwissenschaften. München: Vahlen, 2013.
- Fasel, D.: Übersicht über NoSQL-Technologien und - Datenbanken. In: Fasel, D.; Meier, A. (Hg.): Big Data. Wiesbaden: Springer Fachmedien Wiesbaden, 2016, S. 109–138.
- Fayyad, U.: From Data Mining to Knowledge Discovery in Databases. AI Magazine (1996) 3, S. 37–54.
- Fink, T.: E-Business in der Logistik. In: Koether, R. (Hg.): Taschenbuch der Logistik. 5., aktualisierte Auflage. München: Fachbuchverlag Leipzig im Carl Hanser Verlag, 2018, S. 225–241.
- Flake, G. W.; Tarjan, R. E.; Tsioutsoulouklis, K.: Graph Clustering and Minimum Cut Trees. Internet Mathematics 1 (2004) 4, S. 385–408.
- Fleischmann, B.; Koberstein, A.: Strategic Network Design. In: Stadtler, H.; Kilger, C.; Meyr, H. (Hg.): Supply Chain Management and Advanced Planning. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, S. 107–123.
- Fleischmann, B.; Kopfer, H.: Transport- und Tourenplanung. In: Tempelmeier, H. (Hg.): Planung logistischer Systeme. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018, S. 71–98.
- Francis, N.; Taylor, A.; Green, A.; Guagliardo, P.; Libkin, L.; Lindaaker, T.; Marsault, V.; Plantikow, S.; Rydberg, M.; Selmer, P.: Cypher: An Evolving Query Language for Property Graphs. In: Das, G.; Jermaine, C.; Bernstein, P. (Hg.): Proceedings of the 2018 International Conference on Management of Data - SIGMOD '18. New York, New York, USA: ACM Press, 2018, S. 1433–1445.
- Friedli, T.; Thomas, S.; Mundt, A.: Management globaler Produktionsnetzwerke. München: Carl Hanser Verlag GmbH & Co. KG, 2013.
- Gallagher, B.: Matching Structure and Semantics: A Survey on Graph-Based Pattern Matching. AAAI fall symposium on capturing and using pattern for evidence detection (2006), S. 45–53.
- Ganesh Chandra, D.: BASE analysis of NoSQL database. Future Generation Computer Systems 52 (2015), S. 13–21.
- Ganzha, M.: Federated Conference on Computer Science and Information Systems (FedCSIS), 2013: 8 - 11 Sept. 2013, Kraków, Poland. Piscataway, NJ: IEEE, 2013.
- Gleißner, H.; Möller, K.: Fallstudien Logistik: Logistikwissen in der praktischen Anwendung. Wiesbaden: Gabler, 2009.
- Göpfert, I.: Logistik: Führungskonzeption und Management von Supply Chains. 3. Aufl. Vahlens Handbücher der Wirtschafts- und Sozialwissenschaften. Vahlens Handbücher der Wirtschafts- und Sozialwissenschaften. München: Verlag Franz Vahlen, 2013.
- Günther, H.-O.: Supply Chain Management and Advanced Planning Systems: A Tutorial. In: Günther, H.-O. (Hg.): Supply-chain-Management und Logistik. [Online-ausg.] Heidelberg: Physica-Verl., 2005, S. 3–40.
- Häberle, J.; Kilger, C.: Strategic Network Design in the Chemical Industry. In: Stadtler, H.; Kilger, C.; Meyr, H. (Hg.): Supply Chain Management and Advanced Planning. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, S. 363–376.

- Han, J.; Kamber, M.; Pei, J.: *Data mining: Concepts and techniques*. 3. ed. The Morgan Kaufmann series in data management systems. Amsterdam: Elsevier/Morgan Kaufmann, 2012.
- Hausladen, I.: *IT-gestützte Logistik*. Wiesbaden: Springer Fachmedien Wiesbaden, 2016.
- Hellingrath, B.; Laakmann, F.; Nayabi, K.: Auswahl und Einführung von SCM-Softwaresystemen. In: Beckmann, H. (Hg.): *Supply Chain Management*. Berlin: Springer, 2004.
- Holder, L. B.; Cook, D. J.: Introduction. In: Cook, D. J.; Holder, L. B. (Hg.): *Mining graph data*. Hoboken, N.J: Wiley-Interscience, 2007, S. 1–16.
- Ifan, D.; Xiaofei, X.; Shengchun, D.; Khan, I. A.: Clustering Framework for Supply Chain Management (SCM) System. In: *Second Workshop on Digital Media and its Application in Museum and Heritages (DMAMH 2007)*. IEEE, 2007, S. 422–426.
- Ito, Y.; Kato, S. In: *2016 International Conference on Agents*. Piscataway, NJ: IEEE, 2016, S. 114–115.
- Junghanns, M.; Petermann, A.; Neumann, M.; Rahm, E.: Management and Analysis of Big Graph Data: Current Systems and Open Challenges. In: Zomaya, A. Y.; Sakr, S. (Hg.): *Handbook of Big Data Technologies*. Cham: Springer International Publishing, 2017, S. 457–506.
- Khan, A.; Ranu, S.: Big-Graph: Querying, Mining, and Beyond. In: Zomaya, A. Y.; Sakr, S. (Hg.): *Handbook of Big Data Technologies*. Cham: Springer International Publishing, 2017, S. 531–582.
- Kilger, C.: Computer Assembly. In: Stadtler, H.; Kilger, C.; Meyr, H. (Hg.): *Supply Chain Management and Advanced Planning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, S. 425–442.
- Kilger, C.; Wagner, M.: Demand Planning. In: Stadtler, H.; Kilger, C.; Meyr, H. (Hg.): *Supply Chain Management and Advanced Planning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, S. 125–154.
- Klösgen, W.; Zytkow, J. M.: Knowledge discovery in databases terminology. In: Fayyad, U. M. (Hg.): *Advances in knowledge discovery and data mining*. Menlo Park, Calif.: AAAI Press, 1996, S. 573–592.
- Krischke, A.; Röpcke, H.: *Graphen und Netzwerktheorie: Grundlagen - Methoden - Anwendungen ; mit 137 Bildern und zahlreichen Beispielen. Quantitative Methoden*. Krischke, André (VerfasserIn) Röpcke, Helge (VerfasserIn). München: FachbuchverlagLeipzig im Carl Hanser Verlag, 2015.
- Krumke, S. O.; Noltemeier, H.: *Graphentheoretische Konzepte und Algorithmen ; mit 9 Tabellen und 90 Aufgaben. 2., aktualisierte Aufl.* Wiesbaden: Vieweg + Teubner, 2009.
- Kudraß, T.; Brinkhoff, T.: *Taschenbuch Datenbanken. 2., neu bearbeitete Aufl.* München: Hanser, 2015.
- Larson, P. D.; Halldorsson, A.: Logistics versus supply chain management: An international survey. *International Journal of Logistics Research and Applications* 7 (2004) 1, S. 17–31.
- Liben-Nowell, D.; Kleinberg, J.: The Link Prediction Problem for Social Networks. In: Kraft, D. (Hg.): *Proceedings of the twelfth international conference on Information and knowledge management*. New York: ACM, 2003, S. 556–559.
- Lü, L.; Zhou, T.: Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications* 390 (2011) 6, S. 1150–1170.
- Mattfeld, D.; Vahrenkamp, R.: *Logistiknetzwerke*. Wiesbaden: Springer Fachmedien Wiesbaden, 2014.
- Meier, A.: Zur Nutzung von SQL- und NoSQL-Technologien. *HMD Praxis der Wirtschaftsinformatik* 53 (2016) 4, S. 415–427.

- Meier, A.: Werkzeuge der digitalen Wirtschaft: Big Data, NoSQL & Co. Wiesbaden: Springer Fachmedien Wiesbaden, 2018.
- Meier, A.; Kaufmann, M.: SQL- & NoSQL-Datenbanken. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016.
- Mentzer, J. T.; Stank, T. P.; Esper, T. L.: Supply Chain Management and its Relationship to Logistics, Marketing, Production, and Operation Management. *Journal of Business Logistics* 29 (2008) 1, S. 31–46.
- Miller, J. J. (Hg.): Graph Database Applications and Concepts with Neo4j. Bd. Vol. 2324, No.36. 2013.
- Muchna, C.; Brandenburg, H.; Fottner, J.; Gutermuth, J.: Grundlagen der Logistik. Wiesbaden: Springer Fachmedien Wiesbaden, 2018.
- Nahrstedt, H.: Algorithmen für Ingenieure. Wiesbaden: Springer Fachmedien Wiesbaden, 2018.
- Namata, G.; Getoor, L.: Link Prediction. In: Sammut, C.; Webb, G. I. (Hg.): *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010, S. 609–612.
- Nebel, M.; Wild, S.: Entwurf und Analyse von Algorithmen. Wiesbaden: Springer Fachmedien Wiesbaden, 2018.
- Neo4j: graphgist - Neo4j Graph Database Platform. 2019. URL: <https://neo4j.com/graphgist/supply-chain-management> (zuletzt geprüft am 01.06.2020).
- Neo4j: APOC User Guide 4.0. 2020a. (Zuletzt geprüft am 06.06.2020).
- Neo4j: Chapter 5. Algorithms - The Neo4j Graph Data Science Library Manual v1.2. 2020b. URL: <https://neo4j.com/docs/graph-data-science/current/algorithms/> (zuletzt geprüft am 11.06.2020).
- Neo4j: The Neo4j Java Developer Reference v4.0. 2020c. URL: <https://neo4j.com/docs/java-reference/4.0/> (zuletzt geprüft am 10.06.2020).
- Nickerson, R. C.; Varshney, U.; Muntermann, J.: A method for taxonomy development and its application in information systems. *European Journal of Information Systems* 22 (2013) 3, S. 336–359.
- Petersohn, H.: *Data Mining*. Oldenbourg Wissenschaftsverlag, 2005.
- Pfohl, H.-C.: *Logistiksysteme*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018.
- Pokorny, J.: NoSQL databases: a step to database scalability in web environment. *International Journal of Web Information Systems* 9 (2013) 1, S. 69–82.
- Pokorny, J.: Graph Databases: Their Power and Limitations. In: Saeed, K.; Homenda, W. (Hg.): *Computer Information Systems and Industrial Management*. Cham: Springer International Publishing, 2015, S. 58–72.
- Ramraj, T.; Prabhakar, R.: Frequent Subgraph Mining Algorithms – A Survey. *Procedia Computer Science* 47 (2015), S. 197–204.
- Rao, B.; Mitra, A.: An approach to merging of two community subgraphs to form a community graph using graph mining techniques. 2014.
- Riesen, K.; Bunke, H.: Graph Classification Based on Vector Space Embedding. *International Journal of Pattern Recognition and Artificial Intelligence* 23 (2009) 06, S. 1053–1081.
- Robak, S.; Franczyk, B.; Robak, M. (Hg.): *Applying Big Data and Linked Data Concepts in Supply Chains Management*. Krakau: IEEE, 2013.
- Robinson, I.; Webber, J.; Eifrem, E.: *Graph databases*. 2nd ed. Robinson, Ian, (author.) Webber, James, (author.) Eifrem, Emil, (author.) Sebastopol, CA: O’Reilly, 2015.
- Runkler, T. A.: *Data Mining: Methoden und Algorithmen intelligenter Datenanalyse. Computational intelligence*. Wiesbaden: Vieweg + Teubner, 2010.
- Schade, K.: *Stochastische Optimierung: Bestandsoptimierung in mehrstufigen Lagernetzwerken*. 1. Aufl. Vieweg+Teubner Research. Wiesbaden: Vieweg+Teubner, 2012.

- Schaeffer, S. E.: Graph clustering. *Computer Science Review* 1 (2007) 1, S. 27–64.
- Scheidler, A. A.: Methode zur Erschließung von Wissen aus Datenmustern in Supply-Chain-Datenbanken. 1st ed. Bd. v.1. Schriftenreihe Fortschritte in der IT in Produktion und Logistik. Göttingen: Cuvillier Verlag, 2017.
- Schemm, J. W.: Zwischenbetriebliches Stammdatenmanagement: Lösungen für die Datensynchronisation zwischen Handel und Konsumgüterindustrie. Business-Engineering. Berlin: Springer, 2009.
- Schmaltz, R.: Semantic Web Technologien für das Wissensmanagement. Schumann, M. (Hg.). Georg-August-Universität Göttingen: göttingen, 2004.
- Scholl, A.; Domschke, W.: Logistik: Rundreisen und Touren. 5. Aufl. Oldenbourgs Lehr- und Handbücher der Wirtschafts- u. Sozialwissenschaften. 2007.
- Schönsleben, P.: Integrales Logistikmanagement. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016.
- Schröder, M.: Strukturierte Verbesserung des Supply Chain Risikomanagements. Wiesbaden: Springer Fachmedien Wiesbaden, 2019.
- Schuh, G.; Hering, N.; Brummer, A.: Einführung in das Logistikmanagement. In: Schuh, G.; Stich, V. (Hg.): Logistikmanagement. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013a, S. 1–31.
- Schuh, G.; Stich, V.; Helmig, J.: Konzept des Supply-Chain-Managements. In: Schuh, G.; Stich, V. (Hg.): Logistikmanagement. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013b.
- Schuh, G.; Stich, V.; Helmig, J.: Konzepte des Supply-Chain-Managements. In: Schuh, G.; Stich, V. (Hg.): Logistikmanagement. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013c, S. 209–255.
- Schulte, C.: Logistik: Wege zur Optimierung der Supply Chain. 7., vollständig überarbeitete und erweiterte Auflage. München, Germany: Verlag Franz Vahlen, 2017.
- Shao, B.; Li, Y.: Parallel Processing of Graphs. In: Fletcher, G.; Hidders, J.; Larriba-Pey, J. L. (Hg.): Graph Data Management. Cham: Springer International Publishing, 2018, S. 143–162.
- Steven, M.; Pollmeier, I.: Aufgaben von Controllingsystemen zur Koordination von Supply Chains. In: Becker, J.; Knackstedt, R.; Pfeiffer, D. (Hg.): Wertschöpfungsnetzwerke. Heidelberg: Physica-Verlag HD, 2008, S. 49–63.
- Stewens, M.: Gestaltung und Steuerung von Supply Chains: Zugl.: Chemnitz, Techn. Univ., Diss., 2004 u.d.T.: Stewens, Michael: Supply Chain Management - Gestaltung und Steuerung unternehmensübergreifender Wertschöpfungsnetze. 1. Aufl. Bd. 14. Reihe. Lohmar: Eul, 2005.
- Stich, V.; Hering, N.; Brosze, T.: Beschaffungslogistik. In: Schuh, G.; Stich, V. (Hg.): Logistikmanagement. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013a, S. 77–114.
- Stich, V.; Oedekoven, D.; Brosze, T.: Informationssysteme für das Logistikmanagement. In: Schuh, G.; Stich, V. (Hg.): Logistikmanagement. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013b, S. 257–304.
- Stich, V.; Hering, N.; Cuber, S.: Konfiguration logistischer Netzwerke. In: Schuh, G.; Stich, V. (Hg.): Logistikmanagement. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013c, S. 35–76.
- Struckmann, W.; Wätjen, D.: Mathematik für Informatiker: Grundlagen und Anwendungen. 2., überarbeitete und erweiterte Auflage. Berlin und Heidelberg: Springer Vieweg, 2016.
- Studer, T.: Relationale Datenbanken. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019.

- Sucky, E.; Hönscheidt, I.: Konfiguration von Wertschöpfungsnetzwerken. In: Becker, J.; Knackstedt, R.; Pfeiffer, D. (Hg.): Wertschöpfungsnetzwerke. Heidelberg: Physica-Verlag HD, 2008.
- Sürle, C.; Reuter, B.: Supply Chain Analysis. In: Stadtler, H.; Kilger, C.; Meyr, H. (Hg.): Supply Chain Management and Advanced Planning. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, S. 29–53.
- Tiwari, V.; Tiwari, V.; Gupta Shailendra; Tiwari, R.: Association Rule Mining: A Graph Based Approach for Mining Frequent Itemsets. In: Thatcher, S. (Hg.): International Conference on Networking and Information Technology (ICNIT), 2010. Piscataway, NJ: IEEE, 2010.
- Töllner, A.; Jungmann, T.; Bücken, Matthias, Brutscheck, Tobias: Modelle und Modellierung. In: Bandow, G.; Holzmueller, H. H. (Hg.): Das ist gar kein Modell! 1. Aufl. Gabler Research. Wiesbaden: Gabler, 2010, S. 3–22.
- Tsuda, K.; Saigo, H.: Graph Classification. In: Aggarwal, C. C. (Hg.): Managing and Mining Graph Data. Advances in Database Systems. New York [u.a.]: Springer, 2010, S. 337–363.
- Umweltbundesamt: Emissionsdaten. 2020. URL: <https://www.umweltbundesamt.de/themen/verkehr-laerm/emissionsdaten#handbuch-fur-emissionsfaktoren-hbefa> (zuletzt geprüft am 10.06.2020).
- Unterstein, M.; Matthiessen, G.: Relationale Datenbanken und SQL in Theorie und Praxis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.
- Vahrenkamp, R.: Quantitative Logistik für das Supply-chain-Management. München: de Gruyter, 2003.
- Wannenwetsch, H.: Integrierte Materialwirtschaft, Logistik und Beschaffung. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.
- Werner, H.: Supply Chain Management: Grundlagen, Strategien, Instrumente und Controlling. 6. Aufl. Wiesbaden: Springer Fachmedien Wiesbaden, 2017.
- Yan, X.; Han, J.: Discovery of Frequent Substructures. In: Cook, D. J.; Holder, L. B. (Hg.): Mining graph data. Hoboken, N.J.: Wiley-Interscience, 2007, S. 99–116.

Abbildungsverzeichnis

Abbildung 2.1	Beispielhafte Darstellung einer SC in Anlehnung an Busch und Dangelmaier (2004b, S. 5)	8
Abbildung 2.2	SCM-Aufgabenmodell in Anlehnung an Schulte (2017, S. 791) nach Hellingrath et al. (2004)	12
Abbildung 3.1	Architektur und Komponenten eines Informationssystems (Meier und Kaufmann 2016, S. 3)	21
Abbildung 3.2	Rahmenkonzept des Data-Minings (Bissantz und Hagedorn 2009, S. 141)	36
Abbildung 4.1	Taxonomie zur Klassifizierung logistischer Fragestellungen	42
Abbildung 4.2	Kategorisierung graphbasierter Verfahren	46
Abbildung 5.1	Klassifizierung logistischer Fragestellungen anhand der Taxonomie aus Abbildung 4.1	78
Abbildung 5.2	Darstellung der SC durch ein Graphenmodell	80
Abbildung 5.3	Visuelle Darstellung der SC	82
Abbildung 5.4	Ausgabe zur Bestimmung der günstigsten Transportrouten 1	84
Abbildung 5.5	Ausgabe zur Bestimmung der günstigsten Transportrouten 2	85
Abbildung 5.6	Visuelle Darstellung der fertigen SC	85
Abbildung 5.7	Ausgabe zur Abfrage eines bestimmten Lieferanten	86
Abbildung 5.8	Ausgabe zur Abfrage des kürzesten Pfades	87
Abbildung 5.9	Ausgabe des umweltfreundlichsten Pfades	87
Abbildung 5.10	Ausgabe bestimmter Lieferanten	88
Abbildung 5.11	Ausgabe der fertigen Route	89

Tabellenverzeichnis

Tabelle 4.1	Beschreibung graphbasierter Modelle des SCM	48
Tabelle 4.2	Beschreibung von Graph-Management-Algorithmen	49
Tabelle 4.3	Beschreibung von Graph-Mining-Algorithmen	50
Tabelle 4.4	Bewertung der Lösungsansätze zur Erstellung von Planungsszenarien .	54
Tabelle 4.5	Bewertung der Lösungsansätze zur Erstellung von Gestaltungsalternativen	56
Tabelle 4.6	Bewertung der Lösungsansätze zur Auswahl und Bewertung	60
Tabelle 4.7	Bewertung der Lösungsansätze in der Bedarfsplanung	62
Tabelle 4.8	Bewertung der Lösungsansätze in der Netzwerkplanung	64
Tabelle 4.9	Bewertung der Lösungsansätze in der Beschaffungs- und Distributionsplanung	66
Tabelle 4.10	Bewertung der Lösungsansätze im Order Promising	67
Tabelle 4.11	Gesamtbewertung graphbasierter Lösungsansätze hinsichtlich logistischer Aufgabenstellungen	69
Tabelle 4.12	Durchschnittliche Punktzahl pro Kategorie nach Aufgabenebene	70
Tabelle 4.13	Lösungsansätze graphbasierter Modelle des SCM hinsichtlich logistischer Aufgabenstellungen	73
Tabelle 4.14	Lösungsansätze von Graph-Management-Algorithmen hinsichtlich logistischer Aufgabenstellungen	74
Tabelle 4.15	Lösungsansätze von Graph-Mining-Algorithmen hinsichtlich logistischer Aufgabenstellungen	75
Tabelle 5.1	Kostentabelle der Transportarten	81

Abkürzungsverzeichnis

DBMS	Datenbankmanagementsystem
DBS	Datenbanksystem
GDBS	Graphdatenbanksystem
GPM	Graph-Pattern-Mining
KDD	Knowledge Discovery in Databases
LKW	Lastkraftwagen
RDBS	Relationale Datenbanksysteme
SC	Supply Chain
SCM	Supply-Chain-Management
tkm	Tonnenkilometer
TuL-Prozesse	Transport-, Umschlag-, und Lagerungs-Prozesse
TuZ-Modelle	Transport- und Zuordnungsmodelle

Algorithmenverzeichnis

Algorithmus 5.1	Code zum Import eines Lieferanten	81
Algorithmus 5.2	Code zur Erstellung von Beziehungen	81
Algorithmus 5.3	Code zur Berechnung der Distanz	81
Algorithmus 5.4	Code zur Bestimmung optimaler Transportrouten	83
Algorithmus 5.5	Code zur Abfrage eines bestimmten Lieferanten	86
Algorithmus 5.6	Code zur Abfrage des kürzesten Pfades	86
Algorithmus 5.7	Code zur Bestimmung von Zufallszahlen	87
Algorithmus 5.8	Code zur Abfrage bestimmter Lieferanten	88
Algorithmus 5.9	Code zur Abfrage der kürzesten Distanz	89
Algorithmus 5.10	Code zur Erstellung einer Routenverbindung	89
Algorithmus 5.11	Code zur Berechnung der Gesamtdistanz	90