

Verfahren zur Kostenoptimierung von Topologien eingebetteter Kommunikationsnetze

Von der Fakultät für Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde
eines Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

vorgelegt von

Jörg Sommer

geb. in Heidenheim an der Brenz

Hauptberichter:	Prof. em. Dr.-Ing. Dr. h.c. mult. Paul J. Kühn
1. Mitberichter:	Prof. Dr.-Ing. Rudi Knorr, Universität Augsburg
2. Mitberichter:	Prof. Dr.-Ing. Andreas Kirstädter
Tag der Einreichung:	29. Oktober 2010
Tag der mündlichen Prüfung:	19. Juli 2011

Institut für Kommunikationsnetze und Rechnersysteme
der Universität Stuttgart

2011

Kurzfassung

Zur Steuerung, Regelung und Datenverarbeitung werden elektronische Systeme in technische Systeme, Prozesse und Umgebungen eingebettet. Diese sind häufig mit der physikalischen Umwelt verbunden, sammeln über Sensoren Informationen und nehmen über Aktoren Einfluss auf ihre Umgebung. Anfangs waren die eingebetteten Systeme autonom und interagierten lediglich mit ihren angeschlossenen Sensoren und Aktoren. Um technische Systeme und Prozesse effizienter und sicherer zu gestalten sowie innovative Dienste zu realisieren, wurden eingebettete Systeme direkt mittels Punkt-zu-Punkt-Verbindungen miteinander vernetzt. So entstanden die ersten verteilten eingebetteten Systeme.

Zunehmend erhöhten sich die Anzahl eingebetteter Systeme und der Informationsaustausch, was zu einem größeren Verkabelungsaufwand führte. Jedoch kann aus Platz-, Ressourcen- und Kostengründen nicht jedes eingebettete System mit jedem verbunden werden. Deshalb wurden Kommunikationsnetze zur Vernetzung der verteilten eingebetteten Systeme eingesetzt. Ein solches *eingebettetes Kommunikationsnetz* ermöglicht den Informationsaustausch und die transparente Nutzung gemeinsamer Ressourcen. Üblicherweise sind eingebettete Kommunikationsnetze nach außen nicht direkt als Kommunikationsnetz erkennbar. Bekannte Anwendungsbereiche sind die Fahrzeug- und Flugzeugtechnik sowie die Fertigungs-, Prozess- und Automatisierungstechnik.

Seit Beginn der Vernetzung wurde aufgrund teilweise differenzierter Anforderungen in den verschiedenen Anwendungsbereichen eine Vielzahl digitaler, serieller (Feld-) Bussysteme entwickelt. Im Verlauf stiegen die Anzahl der in einer Umgebung gleichzeitig eingesetzten Bussysteme und damit die Komplexität und die Kosten. Ende der neunziger Jahre entstand die Idee, zur Homogenisierung und zur Kostensenkung die in lokalen Netzen verbreitete Ethernet-Technologie einzusetzen.

Aufgrund der weiten Verbreitung von Ethernet sind kostengünstige Komponenten am Markt verfügbar, es existiert eine breite Produktpalette und eine nachhaltige Entwicklung ist gesichert. Ein weiterer Vorteil von Ethernet gegenüber den (Feld-) Bussystemen ist die wesentlich größere Übertragungsrate. So wurden zuerst für die Fertigungs-, Prozess- und Automatisierungstechnik geeignete Ethernet-basierte Lösungen entwickelt. Andere Bereiche wie die Flugzeugtechnik folgten einige Jahre später. Die fehlende Echtzeitfähigkeit wurde durch teilweise proprietäre Erweiterungen ergänzt. Aus Wirtschaftlichkeitsgründen und dank standardisierter Erweiterungen werden zukünftig verstärkt Komponenten eingesetzt, die vollständig kompatibel zu den bereits heute in lokalen Netzen eingesetzten Ethernet-Komponenten sind. Durch den Einsatz

von *Standard Ethernet* entstehen auch neue Möglichkeiten für die Kommunikation zwischen eingebetteten Kommunikationsnetzen und Netzen der Außenwelt.

Durch den Einsatz von Vermittlungsknoten und Vollduplexleitungen werden bei heutigen Ethernet-Netzen Kollisionen vermieden und die Leistungsfähigkeit gesteigert. Das erhöht allerdings die Kosten und den benötigten Platzbedarf. Viele Produkte mit eingebetteten Kommunikationsnetzen werden in einer großen Stückzahl hergestellt, unterliegen einem enormen Wettbewerb und sind somit besonders kostensensitiv. Zudem spielen vor allem bei Transportmitteln und mobilen Geräten die Größe und das Gewicht eine außerordentliche Rolle, weshalb diese minimiert werden müssen. Der Wunsch, Ethernet als Basistechnologie für eingebettete Kommunikationsnetze einzusetzen, und die Anforderungen, die in eingebetteten Bereichen existieren, stehen somit im Widerspruch zueinander. Um diesen Widerspruch weitestgehend aufzulösen, werden in dieser Dissertation Verfahren zur Kostenoptimierung von Topologien (Ethernet-basierter) eingebetteter Kommunikationsnetze entwickelt und bewertet.

Zunächst werden in **Kapitel 2** die Grundlagen und Anwendungsbereiche eingebetteter Kommunikationsnetze behandelt. Zudem werden die wesentlichen Eigenschaften eingebetteter Kommunikationsnetze sowie deren funktionale und nichtfunktionale Anforderungen beschrieben. Diese grenzen eingebettete Kommunikationsnetze von den klassischen Kommunikationsnetzen ab. Danach werden verschiedene Topologien, d. h. die Struktur der Verbindungen sowie die Anzahl installierter Vermittlungsknoten und deren Positionen, ausgeführt. Des Weiteren werden in diesem Kapitel verschiedene Technologien zur Vernetzung verteilter eingebetteter Systeme vorgestellt. Abschließend wird ein Ausblick über die aktuellen Entwicklungen und Trends gegeben.

In **Kapitel 3** werden die für diese Arbeit notwendigen Grundlagen der Topologieoptimierung behandelt. Viele Probleme der Topologieoptimierung können mit existierenden Verfahren der Graphentheorie gelöst werden. Deshalb gliedert sich dieses Kapitel in zwei Teile: (i) Grundlagen der Graphentheorie und (ii) Grundlagen der kombinatorischen Optimierung. Im Grundlagenteil der kombinatorischen Optimierung werden nach einer Einführung exakte Lösungsverfahren sowie bekannte Metaheuristiken vorgestellt.

In **Kapitel 4** erfolgen die umfassende Beschreibung und die Modellierung des Optimierungsproblems. Es werden die Kosten für ein eingebettetes Kommunikationsnetz erläutert und deren Zusammenhang dargestellt. Auch die Eigenschaften, die bei einer Optimierung zwingend berücksichtigt werden müssen, werden in diesem Kapitel beschrieben. Hierzu zählt, dass in eingebetteten Kommunikationsnetzen mehrere physikalische Leitungen gebündelt werden, wodurch eine Kostenreduzierung entsteht.

Beim Entwurf und der Optimierung eingebetteter Kommunikationsnetze muss die Umgebung, in die das Netz später eingebettet wird, berücksichtigt werden, da diese die Vernetzungskosten beeinflusst. So können die Installationskosten in unterschiedlichen Bereichen innerhalb der Umgebung voneinander abweichen. Aus diesem Grund wird ein Modell zur Berücksichtigung positionsabhängiger Kosten eingeführt. Nach der Beschreibung und Modellierung der zu berücksichtigenden Eigenschaften wird eine Ziel- bzw. Kostenfunktion formuliert, welche die Grundlage für die spätere Optimierung bildet. Die formulierte Zielfunktion berücksichtigt zudem die durch die Leitungs Bündelung erzielte Kostenreduzierung.

Neben den Eigenschaften müssen auch gewisse Anforderungen bei der Optimierung berücksichtigt werden. So müssen für die Übertragung des aufkommenden Verkehrs ausreichend Ressourcen im Netz verfügbar sein. Die Ressourcen einfach großzügig zu dimensionieren, würde im Widerspruch zur Kostenminimierung stehen. Deshalb wird ein Modell zur Berücksichtigung der Verkehrsanforderungen beschrieben. Anschließend werden die Anforderungen an eine ausfallsichere Vernetzung formuliert. Abgeschlossen wird dieses Kapitel mit einer Komplexitätsabschätzung des Optimierungsproblems.

In **Kapitel 5** werden neue anwendungsreife und leistungsfähige Verfahren zur Kostenoptimierung von Topologien eingebetteter Kommunikationsnetze entwickelt. Für eine kostenoptimale Topologie müssen sowohl die Vermittlungsknoten optimal positioniert als auch die Komponenten optimal miteinander vernetzt werden. Aus diesem Grund werden sowohl Verfahren zur kostengünstigen Positionierung von Vermittlungsknoten als auch zur Erstellung einer kostengünstigen Vernetzungsstruktur entwickelt. Für eine optimale Leitungsbündelung bestimmen die Verfahren darüber hinaus eine Anzahl sinnvoller Bündelungspunkte und positionieren diese zusätzlich zu den Vermittlungsknoten kostengünstig.

Für die Erstellung einer kostengünstigen Vernetzungsstruktur werden mehrere Verfahren entwickelt. Für eine kostengünstige Vernetzungsstruktur müssen sowohl die verteilten eingebetteten Systeme kostengünstig an die Vermittlungsknoten angebunden als auch die Vermittlungsknoten untereinander kostengünstig miteinander vernetzt werden. Da eingebettete Kommunikationsnetze häufig in sicherheitskritischen Bereichen zum Einsatz kommen, werden neben den Verfahren zur Erstellung einer kostengünstigen baumförmigen Vernetzungsstruktur auch Verfahren zur Erstellung einer kostengünstigen ausfallsicheren Vernetzungsstruktur entwickelt. Bei sämtlichen Verfahren zur Erstellung einer kostengünstigen Vernetzungsstruktur werden auch die Verkehrsanforderungen berücksichtigt und die Leitungen ausreichend dimensioniert.

Die Kosten einer Topologie setzen sich aus den Kosten für die Vermittlungsknoten und für die Vollduplexleitungen zusammen. Für eine kostenoptimale Topologie müssen also beide Kostenarten minimiert werden. Die Gesamtlänge der Leitungen ist umso geringer, umso größer die Anzahl installierter Vermittlungsknoten ist und umgekehrt. Die beiden Ziele – Minimierung der Anzahl der Vermittlungsknoten und Minimierung der Leitungskosten – sind also widersprüchlich zueinander. In Kapitel 5 wird auch ein Verfahren zum Bestimmen einer Kompromisslösung entwickelt.

In **Kapitel 6** werden die Verfahren evaluiert. Hierfür werden Probleminstanzen unterschiedlicher Größe zufällig erzeugt und deren Topologie mit den entwickelten Verfahren optimiert. Neben der Qualität der Lösungen wird auch der Rechenaufwand betrachtet. Es werden die Ergebnisse der Verfahren mit gleichem Anwendungszweck verglichen und diskutiert. In **Kapitel 7** wird die Anwendungsreife der entwickelten Optimierungsverfahren anhand exemplarischer, praxisrelevanter Fragestellungen gezeigt. Es werden Untersuchungen zur Maximalausstattung, d. h. maximale Anzahl sinnvoller Vermittlungsknoten und Bündelungspunkte, durchgeführt. Zudem wird untersucht, wie hoch die zusätzlichen Kosten für eine ausfallsichere Vernetzungsstruktur sind. Mit dem Bestimmen von Kompromisslösungen für exemplarische Probleminstanzen endet dieses Kapitel. Abgeschlossen wird die Dissertation in **Kapitel 8** mit einer Zusammenfassung und einem Ausblick.

Abstract

Algorithms for Cost-based Topology Optimization of Embedded Communication Networks

Today, computer systems are often embedded in other technical systems that comprise hardware and mechanical parts, or they are closely integrated into their environment. Such embedded systems are dedicated to specific tasks like data processing, control, and monitoring of their environment. Typically, they interact with the real world via a set of sensors and actuators. In the early days, embedded systems were isolated devices and interacted only with dedicated sensors and actuators. With the emergence of complex distributed applications and their need for information exchange, this approach quickly turned out to be insufficient. Consequently, point-to-point links were deployed to enable communication (i.e. information exchange) between embedded systems. This was the very beginning of distributed embedded systems.

Subsequently, the number of embedded systems and the amount of exchanged information grew exponentially. Due to space constraints, resource limitations, and cost, the point-to-point interconnections of embedded systems became increasingly impracticable. Thus, the strategy of interconnect the embedded systems by means of point-to-point links proved to be insufficient and needed to replace. The next step in this evolution was the deployment of communication networks that support the establishment of a communication path between embedded systems and the multiplexed transport of information. This was the birth of *embedded communication networks*. Typically, users get not encounter with such an embedded communication network and does not realize it as what it is. Today, embedded communication networks exist in a variety of fields of application including aviation, automotive, automation, and process control.

In recent decades, a multitude of embedded communication network technologies, referred to as field buses, have been developed. They were designed according to specific cost, performance, dependability, and data rate requirements of different fields of application. Over the years, the number of field buses deployed in a single system or environment steadily increased, and hence the system's complexity and cost. At the end of the 1990s, Ethernet gained attraction as an alternative to the field buses, promising to reduce complexity and cost. At that time, Ethernet was already the predominant local area network (LAN) technology.

The advantages of Ethernet include its high data rates, a sustained development guaranteeing planning reliability and the future availability of compatible hardware, and simple management (i.e. plug-n-play capabilities). The idea driving the deployment of Ethernet was to have one single, inexpensive network technology that suits all fields of application, and to establish one

de facto standard for embedded network infrastructures. Mostly under the label of *Industrial Ethernet*, different vendors adapted Ethernet for industrial fields such as factory automation and motion control. Later on, the introduction of Ethernet in other fields of application like avionics followed. Over the years, various, partially proprietary extensions have supplemented Ethernet in order to provide real-time communication support and to achieve high predictability.

One motivation to use Ethernet technology for embedded communication networks is the availability of commercial off-the-shelf (COTS) components. It allows integrators to cut down development cost and time. Furthermore, the large number of Ethernet equipment vendors and the wide range of products promote competition, assuring a supply of better equipment at lower prices. The use of Ethernet COTS components also facilitates the convergence of embedded and enterprise communication networks based on Ethernet.

Since its invention, Ethernet evolved from a bus topology to a so-called micro-segmented network with full duplex links between network components. On the one side, this prevents collisions and improves the performance of Ethernet. It also paves the way for real-time capabilities. On the other side, it requires additional components, namely switches, and additional links. This is particularly problematic for embedded communication networks that are often integrated in mass-marked products. Since such products face a highly competitive, cost-sensitive market, additional component costs are a serious disadvantage. In addition, embedded communication networks have to obey installation space and weight limitations. Such non-functional requirements of embedded networks are in contradiction to the concept of the micro-segmented Ethernet. In view of solving this contradiction, this thesis proposes and evaluates novel algorithms for cost-based topology optimization of embedded communication networks. Herein, topology refers to the layout pattern of the interconnects of the components as well as the number of switches and their positions.

Chapter 2 first introduces the fundamentals of embedded communication networks and their fields of application. It describes the characteristics of such networks and their functional and non-functional requirements. These characteristics and requirements also allow to distinguish embedded communication networks from traditional local area and enterprise networks. Furthermore, this chapter presents basic topologies for embedded communication networks. Finally, it gives an outlook on on-going discussions and major trends.

Chapter 3 discusses the fundamental methods of topology optimization. Many optimization problems are solvable by applying well-known algorithms of graph theory. This chapter is accordingly divided into two parts: (i) Introduction to graph theory and (ii) introduction to combinatorial optimization. The latter includes a short survey of algorithms that are able to find the exact or optimal solution and of algorithms that are able to find a near-optimal solution (heuristics).

Chapter 4 provides a comprehensive description of the optimization problem and introduces a mathematical optimization model. It defines the cost types of an embedded communication network and describes their relationship. Then, it lists the characteristics and constraints that have to be taken into account in order to solve the optimization problem. These characteristics of embedded communication networks include the bundling of individual links that are installed into common ducts. The motivations of such bundling of links or smaller link bundles together are cost savings.

Many embedded communication networks operate in harsh environments (facing vibrations, high temperature, moisture, electromagnetic interference, etc.). The ambient conditions of such environments impose constraints on the network deployment. In particular, the cost of links may depend on their position. For instance, a better link shielding or heat resistant links are needed at some places due to electromagnetic interference or high ambient temperatures, respectively. At other places, it might be extremely expensive or even impossible to install a link. The concept of a cost map is introduced to take this cost variation into account. After the problem formulation, the chapter introduces an objective function, which will be used to evaluate the solutions. This objective function also accounts for the cost benefit of bundling links and installing them in a common duct (link harness).

In order to find an optimal solution, it is mandatory to consider communication demands. In an embedded network, each system interacts with a defined number of other systems and all occurring traffic is predictable (or shaped). Thus, it is possible to specify all communication demands in detail. These communication demands allow to determine the minimum number of required links. Chapter 4 introduces the concept of a demand matrix that specifies the quantity of traffic exchanged between each pair of embedded systems. Besides, some embedded communication networks have to fulfill resilience requirements, e.g. if deployed in safety-critical domains. This chapter formulates such requirements in detail, too. This chapter closes with a complexity estimation of the optimization problem.

Chapter 5 proposes novel, readily usable algorithms for cost-based topology optimization of embedded communication networks. For an optimal solution, the algorithms have to find the optimal position for a given number of switches as well as the optimal link structure that connects the network entities. Since links in a harness may have different endpoints, some full duplex links leave or join a duct at specific points referred to as junctions or breakouts. At these junction points, a bifurcation of the duct is needed to protect all links (in both the original duct and the branching harness) against environmental impacts. When minimizing the cost of a topology, the number of junctions and their positions have to be optimized. Accordingly, this chapter proposes novel algorithms that find cost-saving combinations of switch and junction positions.

Beside algorithms that find (near-) optimal switches' and junctions' positions, this chapter proposes novel algorithms to find a link design that satisfies defined communication requirements at lowest cost. Such a design requires cost-saving connections of the embedded systems to the switches as well as cost-saving interconnections among the switches. Since resilience requirements have to be fulfilled in some cases, section 5.3 proposes both algorithms that find a cost-saving tree (non-resilient) link design and algorithms that find a cost-saving resilient link design.

The overall costs of an embedded communication network consist of switch costs and link costs. For an optimal solution, both types of costs have to be minimized. However, decreasing the number – and thus costs – of switches results in increased link costs due to longer interconnects, and vice versa. Consequently, minimizing the overall costs constitutes a trade-off between optimizing the interconnect structure and the number of switches. The resulting problem is a multi-objective optimization problem. Section 5.4 proposes an approach to solving this problem and to finding a compromise solution.

Chapter 6 addresses the performance evaluation of the optimization algorithms. The evaluation compares the algorithms for positioning switches and junctions and the algorithms for designing the interconnection layout, respectively. For this purpose, sets of instances of different problem sizes (different numbers of network entities, etc.) are generated randomly. In each instance, the network entities are placed randomly.

By means of exemplary, practical scenarios, **chapter 7** illustrates that the developed optimization algorithms are suitable for real-world scenarios. Investigations address the maximal equipage, i.e. the sensible maximum number of switches and junctions. Furthermore, the additional costs of a resilient topology are analyzed in comparison to a tree topology. The chapter closes with an illustration of the trade-off between switch costs and link costs using the example of two network instances with 35 and 50 nodes. Finally, **chapter 8** concludes the thesis and gives an outlook on future work.

Inhaltsverzeichnis

Kurzfassung	I
Abstract	V
Inhaltsverzeichnis	IX
Abbildungsverzeichnis	XII
Tabellenverzeichnis	XV
Algorithmenverzeichnis	XVI
Abkürzungen und Symbole	XIX
1 Einführung	1
1.1 Einordnung und Herausforderungen	1
1.2 Übersicht über die Arbeit	3
2 Eingebettete Kommunikationsnetze	5
2.1 Begriffe und Einordnung	5
2.2 Anwendungsbereiche	6
2.3 Eigenschaften	8
2.4 Anforderungen	10
2.4.1 Funktionale Anforderungen	10
2.4.2 Nichtfunktionale Anforderungen	12
2.5 Netztopologien	14
2.6 Vernetzungstechnologien	19
2.6.1 Überblick bekannter (Feld-) Bussysteme	19
2.6.2 Ethernet	22
2.6.3 Vergleich (Feld-) Bussysteme und Ethernet	29
2.7 Entwicklungen und Trends	31
3 Grundlagen der Topologieoptimierung	35
3.1 Graphentheorie	35
3.1.1 Grundbegriffe und Definitionen	35
3.1.2 Speicherung von Graphen	37
3.1.3 Minimal aufspannende Bäume	39

3.1.4	Kürzeste Wege	40
3.2	Kombinatorische Optimierung	42
3.2.1	Einführung	42
3.2.2	Exakte Lösungsverfahren	44
3.2.3	Heuristiken und Metaheuristiken	45
4	Mathematisches Modell	55
4.1	Problembeschreibung	55
4.1.1	Vernetzungskosten	55
4.1.2	Herausforderungen	56
4.2	Berücksichtigung von Eigenschaften	59
4.2.1	Leitungsbaum	59
4.2.2	Positionsabhängige Kosten	62
4.3	Formulierung der Kostenfunktion	63
4.4	Berücksichtigung von Anforderungen	65
4.4.1	Verkehrsanforderungen	65
4.4.2	Ausfallsichere Vernetzung	67
4.5	Komplexitätsabschätzung	67
5	Verfahren zur Kostenoptimierung von Topologien	69
5.1	Verwandte Probleme und Verfahren	69
5.2	Positionierungsverfahren	73
5.2.1	Exaktes Lösungsverfahren	73
5.2.2	Heuristische Lösungsverfahren	77
5.2.3	Ergänzende Verfahren	85
5.3	Vernetzungsverfahren	88
5.3.1	Baumförmige Vernetzungsstruktur	90
5.3.2	Ausfallsichere Vernetzungsstruktur	95
5.4	Verfahren zur Bestimmung einer Kompromisslösung	101
6	Evaluierung der Verfahren	105
6.1	Evaluierungsstrategie	105
6.2	Positionierungsverfahren	106
6.2.1	Eingeschränkte Probleminstanzen	106
6.2.2	Uneingeschränkte Probleminstanzen	110
6.3	Vernetzungsverfahren	114
6.3.1	Baumförmige Vernetzungsstruktur	115
6.3.2	Ausfallsichere Vernetzungsstruktur	118
7	Anwendung der Verfahren	123
7.1	Maximalausstattung	123
7.2	Kostenvergleich Vernetzungsstrukturen	125
7.3	Bestimmung einer Kompromisslösung	128
8	Zusammenfassung und Ausblick	131

Inhaltsverzeichnis	XI
A Softwarebibliothek zur Kostenoptimierung von Topologien	135
A.1 Architektur und Funktionsumfang	135
A.2 Ablauf einer Optimierungsstudie	138
B Ergänzende Ergebnisse	141
C Konfidenzintervalle	145
Literaturverzeichnis	167
Danksagung	169

Abbildungsverzeichnis

2.1	Überblick Netztopologien	15
2.2	Beispiele für hybride Netztopologien	18
2.3	Die Geschichte von Ethernet, dessen Erweiterungen und Anwendungsbereiche	23
2.4	Rahmenformate von Ethernet	24
2.5	Einordnung der IEEE 802-Standards und der Ethernet-Erweiterungen	26
4.1	Zusammenhang zwischen der Anzahl der Vermittlungsknoten und der Gesamtlänge der Leitungen	57
4.2	Einfluss der Positionen der Vermittlungsknoten und der Vernetzungsstruktur auf die Leitungskosten	58
4.3	Modellkomponenten eines Kabelbaumes	59
4.4	Beispiel für eine Kostenreduzierung durch Leitungsbündelung	60
4.5	Kostenfunktion für einen Leitungskanal ohne und mit Bündelungsgewinn . . .	60
4.6	Beispiel zur Berechnung der Leitungskosten	63
5.1	Intensivierung durch Verschieben der Vermittlungsknoten und Bündelungspunkte	83
5.2	Teilnetz mit einem relevanten Bündelungspunkt mit zwei Nachbarn	87
5.3	Verfahren zur abschließenden Positionsoptimierung	87
5.4	Sequentielle Schritte zur Erstellung einer vollständigen Vernetzungsstruktur . .	89
5.5	Erstellung eines Nachbarschaftsbaumes durch Entfernen und Hinzufügen eines Leitungskanals	93
5.6	Kombination zweier minimaler Spannbäume zur Erzeugung einer ausfallsicheren Vernetzungsstruktur	97
5.7	Beispiel zweier disjunkter Pfade mit mehrfach überschrittenem Bündelungspunkt	99
5.8	Modifikation der Leitungskanäle beim Two-Optimal-Cycle Verfahren	100
5.9	Prinzip zur Bestimmung einer Kompromisslösung	102
6.1	Einfluss der Anzahl zu untersuchender Nachbarschaftslösungen auf die Lösungsqualität und den Berechnungsaufwand	116
6.2	Einfluss der Suchtiefe auf die Lösungsqualität und den Berechnungsaufwand .	116
6.3	Einfluss der Anzahl der Ersetzungen pro detektiertem Kreis auf die Lösungsqualität bei der gierigen Auswahlstrategie	120
6.4	Einfluss der Anzahl der Ersetzungen pro detektiertem Kreis auf die Lösungsqualität bei der zufälligen Auswahlstrategie	120
6.5	Einfluss der maximalen Anzahl aufeinander folgender Iterationen ohne Verbesserung auf die Lösungsqualität bei der gierigen Auswahlstrategie	120

6.6	Einfluss der maximalen Anzahl aufeinander folgender Iterationen ohne Verbesserung auf die Lösungsqualität bei der zufälligen Auswahlstrategie	120
6.7	Anzahl tatsächlich durchgeführter Iterationen bei der gierigen Auswahlstrategie	120
6.8	Anzahl tatsächlich durchgeführter Iterationen bei der zufälligen Auswahlstrategie	120
7.1	Leitungskosten, Kosten für die Vermittlungsknoten und Vernetzungskosten für eine Problem Instanz mit 35 Endknoten	129
7.2	Leitungskosten, Kosten für die Vermittlungsknoten und Vernetzungskosten für eine Problem Instanz mit 50 Endknoten	129
7.3	Kompromisslösung für eine Problem Instanz mit 35 Endknoten	130
A.1	Architektur der Softwarebibliothek	137
A.2	Ablauf einer Optimierungsstudie	138
A.3	Aufbau einer Parameterdatei	140

Tabellenverzeichnis

2.1	Überblick bekannter (Feld-) Bussysteme	21
6.1	Vergleich der Verfahren zur Positionierung von Vermittlungsknoten	107
6.2	Vergleich der Positionierungsverfahren mit beliebig positionierbaren Vermittlungsknoten und eingeschränkten Problem instanzen	108
6.3	Vergleich der Positionierungsverfahren mit integrierten Vermittlungsknoten und eingeschränkten Problem instanzen	109
6.4	Vergleich der Positionierungsverfahren mit beliebig positionierbaren Vermittlungsknoten	112
6.5	Vergleich der Positionierungsverfahren mit integrierten Vermittlungsknoten	113
6.6	Vergleich der Verfahren zur Erstellung einer baumförmigen Vernetzungsstruktur	117
6.7	Vergleich der Verfahren zur Erstellung einer ausfallsicheren Vernetzungsstruktur	121
7.1	Anzahl sinnvoller Vermittlungsknoten und Bündelungspunkte	124
7.2	Kostenvergleich baumförmiger und ausfallsicherer Vernetzungsstrukturen	126
7.3	Vergleich der Kostenanteile von Vermittlungsnetz und Zugangsnetz	127
B.1	Vergleich der Verfahren zur Erstellung einer baumförmigen Vernetzungsstruktur unter Berücksichtigung einer Kostenmatrix	141
B.2	Vergleich der Verfahren zur Erstellung einer ausfallsicheren Vernetzungsstruktur unter Berücksichtigung einer Kostenmatrix	142
C.1	Konfidenzintervalle zu den Ergebnissen in Tabelle 6.2	145
C.2	Konfidenzintervalle zu den Ergebnissen in Tabelle 6.3	145
C.3	Konfidenzintervalle zu den Ergebnissen in Tabelle 6.4	146
C.4	Konfidenzintervalle zu den Ergebnissen in Tabelle 6.5	147
C.5	Konfidenzintervalle zu den Ergebnissen in Tabelle 6.6	148
C.6	Konfidenzintervalle zu den Ergebnissen in Tabelle 6.7	149
C.7	Konfidenzintervalle zu den Ergebnissen in Tabelle B.1	150
C.8	Konfidenzintervalle zu den Ergebnissen in Tabelle B.2	151

Algorithmenverzeichnis

3.1	Algorithmus von Prim	40
3.2	Allgemeines Nachbarschaftssuchverfahren	47
3.3	Prinzip des Simulated Annealing	49
3.4	Prinzip des Toleranzschwellwertverfahrens	51
3.5	Prinzip des Tabu Search	53
5.1	SA-basiertes Positionierungsverfahren	78
5.2	Eröffnungsverfahren für SA-basiertes Positionierungsverfahren	79
5.3	Verfahren zur Erzeugung einer Nachbarschaftslösung	80
5.4	TS-basiertes Positionierungsverfahren	84
5.5	Ergänzungsverfahren zur Erstellung einer baumförmigen Vernetzungsstruktur .	91
5.6	Lokales Suchverfahren zur Verfeinerung einer baumförmigen Vernetzungsstruktur	92
5.7	Gieriges Verfahren zur Verfeinerung einer baumförmigen Vernetzungsstruktur .	94
5.8	Eröffnungsverfahren zur Erstellung einer ausfallsicheren Vernetzungsstruktur .	98

Abkürzungen und Symbole

Abkürzungen

AFDX	Avionics Full Duplex Switched Ethernet
ARP	Address Resolution Protocol
AS-I	Aktor-Sensor-Interface
AVB	Audio/Video Bridging
CAN	Controller Area Network
CENELEC	Comité Européen de Normalisation Electrotechnique
cint	Konfidenzintervall
COTS	Commercial off-the-shelf
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DA	Destination Address
DHCP	Dynamic Host Configuration Protocol
DIX	DEC, Intel und Xerox
EAP	Extensible Authentication Protocol
EIB	Europäischer Installationsbus
EMV	Elektromagnetische Verträglichkeit
Ethernet/IP	Ethernet Industrial Protocol
FCS	Frame Check Sequence
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol

ID	Identifier
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IKR	Institut für Kommunikationsnetze und Rechnersysteme
IKR SimLib	IKR Simulation Library
ILP	Integer LP
IP	Internet Protocol
ISO	International Organization for Standardization
KE	Kosteneinheit
LACP	Link Aggregation Control Protocol
LAN	Local Area Network
LIN	Local Interconnect Network
LLC	Logical Link Control
LON	Local Operating Network
LP	Linear Program
MAC	Medium Access Control
max	Maximum (Maximalwert)
mean	Arithmetischer Mittelwert
Mil-Std	Military-Standard
MILP	Mixed ILP
min	Minimum (Minimalwert)
MOST	Media Oriented Systems Transport
MST	Minimal Spanning Tree
MSTP	Multiple Spanning Tree Protocol
NTP	Network Time Protocol
OCST	Optimal Communication Spanning Tree
OSI	Open Systems Interconnection
PAD	Padding

PC	Personal Computer
PoE	Power over Ethernet
Pre	Preamble
Profibus	Process field bus
RSVP	Resource reSerVation Protocol
SA	Simulated Annealing
SCIP	Solving Constraint Integer Programs
SFD	Start of Frame Delimiter
SNMP	Simple Network Management Protocol
SNTP	Simple Network Time Protocol
ST	Steiner Tree
STP	Spanning Tree Protocol
TCI	Tag Control Information
TCP	Transmission Control Protocol
TPID	Tag Protocol ID
TS	Tabu Search
UDP	User Datagram Protocol
VLAN	Virtual LAN
WLAN	Wireless LAN

Symbole

$A(G)$	Adjazenzmatrix (Nachbarschaftsmatrix) des Graphen G
$a_{i,j}$	Element (i,j) der Matrix $A(G)$
B	Endliche Menge von Bündelungspunkten
c	Allgemeine Kosten für eine (Menge von) Kante(n)
c_L	Kosten für eine Längeneinheit einer physikalischen Leitung
c_N	Kosten für einen Endknoten
c_S	Kosten für einen Vermittlungsknoten
$c_{\Sigma L}$	(Gesamt-) Leitungskosten
$c_{i,j}$	Kosten für die Kante oder den kürzesten (günstigsten) Pfad zwischen den Punkten i und j mit den Koordinaten (x_i,y_i) und (x_j,y_j)
c_{Port}	Kosten für einen (Ethernet-) Anschluss
c_{fix}	Fixkostenanteil
$d_{i,j}$	Distanz zwischen den Punkten i und j mit den Koordinaten (x_i,y_i) und (x_j,y_j)
d	Suchtiefe (baumförmige Vernetzungsstruktur) oder Anzahl der Ersetzungen pro detektiertem Kreis (ausfallsichere Vernetzungsstruktur)
$E (E(G))$	Menge von Kanten (des Graphen G)
G	Allgemeiner Graph bestehend aus Knoten und Kanten
G_T	Baumförmiger Graph (Baum) bestehend aus Knoten und Kanten
$I(G)$	Inzidenzmatrix (Beziehungsmatrix) des Graphen G
K	Nicht leere, endliche Menge zulässiger Koordinaten
k_B	Physikalische Boltzmannkonstante
$l_{\Sigma L}$	Gesamtlänge der physikalischen Leitungen
L	Endliche Menge zulässiger Lösungen
N	Nicht leere, endliche Menge invarianter Endknoten
o	Binäre Variable zur Kennzeichnung integrierter Vermittlungsknoten
P	Pfad oder Weg in einem Graph
p	Abkühlungsgröße

$p(\cdot)$	Endliche Folge von Operationen zur Erzeugung einer zulässigen Nachbarschaftslösung
P_A	Akzeptanzwahrscheinlichkeit
\mathbf{R}	Verkehrsanforderungsmatrix
$r_{i,j}$	Element (i,j) der Matrix \mathbf{R}
S	Nicht leere, endliche Menge von Vermittlungsknoten
T	Tabuliste (Menge verbotener Lösungen oder Schritte)
$V(V(G))$	Nicht leere Menge von Knoten (des Graphen G)
W_i	Energieniveau im Zustand i
x_i	x Koordinate des Punktes i
y_i	y Koordinate des Punktes i
$z(\cdot)$	Ziel- oder Kostenfunktion eines Optimierungsproblems
\mathbb{K}	Algebraischer Körper
\mathbb{N}	Menge der ganzen Zahlen (natürliche Zahlen)
$\mathbb{Q}(\mathbb{Q}^+)$	Menge der rationalen (positiven) Zahlen
$\mathbb{R}(\mathbb{R}^+)$	Menge der reellen (positiven) Zahlen
\mathfrak{N}	Problemspezifischer Parameter mit einem Wert $\gg 1$
β	Anzahl erzeugter Nachbarschaftslösungen
$\mathbf{\Gamma}$	Matrix mit den Umgebungskosten (positionsabhängige Kosten)
$\gamma_{i,j}$	Element (i,j) der Matrix $\mathbf{\Gamma}$
Δ	Konnektivitätsmatrix von Leitungssegmenten zwischen Endknoten und Vermittlungsknoten bzw. Bündelungspunkten
$\delta_{i,j}$	Element (i,j) der Matrix Δ
$\epsilon(v)$	(Konnektivitäts-) Grad eines Knotens $v \in V(G)$
$\zeta(u)$	Menge der zum Knoten $u \in V(G)$ inzidierten Kanten
λ	Kostenfaktor für zusätzliche Leitungssegmente
μ	Mittelwert zufällig erzeugter Verkehrsanforderungen
\mathcal{NP}	Klasse der nichtdeterministisch polynomiellen Entscheidungsprobleme

Ξ	Steuerparameter
$\xi_{k,l}^{i,j}$	Binäre Variable zur Kennzeichnung von Leitungen zwischen den Knoten k, l für die Kommunikation zwischen den Knoten i, j
$\theta(\cdot)$	Kostenfunktion für einen Leitungskanal
τ	(Umgebungs-) Temperatur beim Simulated Annealing oder Schwellwert beim Toleranzschwellwertverfahren
Υ	Konnektivitätsmatrix von Leitungskanälen zwischen Vermittlungsknoten und Bündelungspunkten
$v_{i,j}$	Element (i,j) der Matrix Υ
φ	Anzahl gebündelter Leitungssegmente in einem Leitungskanal
Φ	Konnektivitätsmatrix von Leitungssegmenten zwischen den Vermittlungsknoten und Bündelungspunkten
$\phi_{i,j}$	Element (i,j) der Matrix Φ
Ψ	Konnektivitätsmatrix von Leitungskanälen zwischen Endknoten und Vermittlungsknoten bzw. Bündelungspunkten
$\psi_{i,j}$	Element (i,j) der Matrix Ψ

1 Einführung

1.1 Einordnung und Herausforderungen

Zur Steuerung, Regelung und Datenverarbeitung werden elektronische Systeme in technische Systeme, Prozesse und Umgebungen eingebettet. Diese sind häufig mit der physikalischen Umwelt verbunden, sammeln über Sensoren Informationen und nehmen über Aktoren Einfluss auf ihre Umgebung. Aufgrund der zunehmenden Integrationsdichte von Halbleiterbauelementen und sinkenden Herstellungskosten ist die Anzahl eingebetteter Systeme in den letzten Jahrzehnten enorm gestiegen. Anfangs waren sie autonom und interagierten lediglich mit ihren angeschlossenen Sensoren und Aktoren. Um technische Systeme und Prozesse effizienter und sicherer zu gestalten sowie innovative Dienste zu realisieren, wurden eingebettete Systeme direkt mittels Punkt-zu-Punkt-Verbindungen miteinander vernetzt. So entstanden die ersten verteilten eingebetteten Systeme.

Zunehmend erhöhte sich neben der Anzahl eingebetteter Systeme auch der Informationsaustausch, was zu einem größeren Verkabelungsaufwand führte. Jedoch kann aus Platz-, Ressourcen- und Kostengründen nicht jedes eingebettete System mit jedem verbunden werden. Aus diesem Grund wurden Kommunikationsnetze zur Vernetzung der verteilten eingebetteten Systeme eingesetzt. Ein solches *eingebettetes Kommunikationsnetz* ermöglicht den Informationsaustausch und die transparente Nutzung gemeinsamer Ressourcen. Üblicherweise sind eingebettete Kommunikationsnetze nach außen nicht direkt als Kommunikationsnetz erkennbar. Bekannte Anwendungsbereiche sind die Fahrzeug- und Flugzeugtechnik sowie die Fertigungs-, Prozess- und Automatisierungstechnik.

Seit Beginn der Vernetzung wurde aufgrund teilweise differenzierter Anforderungen in den verschiedenen Anwendungsbereichen eine Vielzahl digitaler, serieller (Feld-) Bussysteme entwickelt. Im Verlauf stiegen die Anzahl der in einer Umgebung gleichzeitig eingesetzten Bussysteme und damit die Komplexität und die Kosten. Ende der neunziger Jahre entstand dann die Idee, zur Homogenisierung und zur Kostensenkung die in lokalen Netzen verbreitete Ethernet-Technologie einzusetzen.

Aufgrund der weiten Verbreitung von Ethernet sind kostengünstige Komponenten am Markt verfügbar, es existiert eine breite Produktpalette und eine nachhaltige Entwicklung ist gesichert. Ein weiterer Vorteil von Ethernet gegenüber den klassischen (Feld-) Bussystemen ist die wesentlich größere Übertragungsrate. So wurden zuerst für die Fertigungs-, Prozess- und Automatisierungstechnik geeignete Ethernet-basierte Lösungen entwickelt. Andere Bereiche wie die Flugzeugtechnik folgten einige Zeit später. Die fehlende Echtzeitfähigkeit wurde durch teilwei-

se proprietäre Erweiterungen ergänzt. Derartige Lösungen wurden unter dem Namen *Industrial Ethernet* vermarktet. Aus Wirtschaftlichkeitsgründen und dank standardisierter Erweiterungen werden zukünftig verstärkt Komponenten eingesetzt, die vollständig kompatibel zu den bereits heute in lokalen Netzen eingesetzten Ethernet-Komponenten sind. Durch den Einsatz von *Standard Ethernet* entstehen auch neue Möglichkeiten für die Kommunikation zwischen eingebetteten Kommunikationsnetzen und Netzen der Außenwelt.

Durch den Einsatz von Vermittlungsknoten (*engl. switch*) und Vollduplexleitungen werden bei heutigen Ethernet-Netzen Kollisionen vermieden und die Leistungsfähigkeit gesteigert. Das erhöht allerdings die Kosten und den benötigten Platzbedarf. Viele Produkte mit eingebetteten Kommunikationsnetzen werden in einer großen Stückzahl hergestellt; sie unterliegen einem enormen Wettbewerb und sind somit besonders kostensensitiv. Zudem spielen vor allem bei Transportmitteln und mobilen Geräten die Größe und das Gewicht eine außerordentliche Rolle, weshalb diese minimiert werden müssen. Der Wunsch, Ethernet als Basistechnologie für eingebettete Kommunikationsnetze einzusetzen, und die Anforderungen, die in eingebetteten Bereichen existieren, stehen somit im Widerspruch zueinander. Dieser kann nur aufgelöst werden, wenn die Kosten und der zusätzlich benötigte Platzbedarf minimiert werden. Um dieses Ziel zu erreichen, muss die zugrunde liegende Topologie optimiert werden. Das bedeutet, dass die Vermittlungsknoten optimal positioniert und die Komponenten kostengünstig miteinander vernetzt werden müssen. Dabei müssen sowohl die verteilten eingebetteten Systeme kostengünstig an die Vermittlungsknoten angebunden als auch die Vermittlungsknoten untereinander kostengünstig miteinander vernetzt werden.

Bei der Optimierung müssen die speziellen Eigenschaften und Anforderungen eingebetteter Kommunikationsnetze berücksichtigt werden. Häufig werden Kommunikationsnetze in rauen Umgebungen (erhöhte Temperaturbereiche, Feuchtigkeit, Schmutz, etc.) mit starken Beanspruchungen der Komponenten eingebettet. Deshalb muss beim Entwurf und der Optimierung die Zielumgebung berücksichtigt werden. Zum Schutz der physikalischen Leitungen und aus Wirtschaftlichkeitsgründen werden diese üblicherweise gebündelt in Leitungskanälen installiert. Dies hat zur Folge, dass neben den Vermittlungsknoten auch die Anzahl sinnvoller Bündelungspunkte bestimmt und diese optimal positioniert werden müssen.

Für die Übertragung des aufkommenden Verkehrs müssen innerhalb eines Netzes ausreichend Ressourcen installiert werden. Die Herausforderung ist, eine Vernetzungsstruktur mit minimalen Kosten, die die Verkehrsanforderungen erfüllt, zu bestimmen. Bei sicherheitskritischen Systemen wird das Netz durch strukturelle Redundanz vor Ausfällen gesichert. Das Bestimmen einer kostengünstigen ausfallsicheren Vernetzungsstruktur ist eine weitere Herausforderung bei der Topologieoptimierung.

Die Vernetzungskosten setzen sich aus den Kosten für die Vermittlungsknoten und für die Vollduplexleitungen zusammen. Für eine kostenoptimale Topologie müssen also beide Kostenarten minimiert werden. Die Gesamtlänge der Leitungen ist umso geringer, umso größer die Anzahl installierter Vermittlungsknoten ist und umgekehrt. Die beiden Ziele – Minimierung der Anzahl der Vermittlungsknoten und Minimierung der Leitungskosten – sind also widersprüchlich zueinander. Eine Herausforderung ist, für dieses multikriterielle Optimierungsproblem eine akzeptable Kompromisslösung zu finden.

1.2 Übersicht über die Arbeit

Ziel dieser Arbeit ist die Entwicklung und Bewertung von Verfahren zur Kostenoptimierung von Topologien eingebetteter Kommunikationsnetze. Kapitel 2 befasst sich mit den Grundlagen und Anwendungsbereichen eingebetteter Kommunikationsnetze. Zudem werden die wesentlichen Eigenschaften eingebetteter Kommunikationsnetze sowie deren funktionale und nicht-funktionale Anforderungen beschrieben. Diese grenzen eingebettete Kommunikationsnetze von den klassischen Kommunikationsnetzen ab. Es werden verschiedene Technologien zur Vernetzung verteilter eingebetteter Systeme vorgestellt. Abschließend wird ein Ausblick über die aktuellen Entwicklungen und Trends gegeben.

In Kapitel 3 werden die für diese Arbeit notwendigen Grundlagen der Topologieoptimierung behandelt. Viele Probleme der Topologieoptimierung können mit existierenden Verfahren der Graphentheorie gelöst werden. Deshalb gliedert sich dieses Kapitel in zwei Teile: (i) Grundlagen der Graphentheorie und (ii) Grundlagen der kombinatorischen Optimierung. Im Grundlagenteil der kombinatorischen Optimierung werden nach einer Einführung exakte Lösungsverfahren sowie bekannte Metaheuristiken vorgestellt.

Zur Lösung eines Optimierungsproblems muss dieses geeignet modelliert und eine Zielfunktion formuliert werden. In Kapitel 4 wird eine Ziel- bzw. Kostenfunktion formuliert, welche die Grundlage für die spätere Optimierung bildet. Bei der Modellierung und Formulierung der Zielfunktion werden die Eigenschaften eingebetteter Kommunikationsnetze entsprechend berücksichtigt. Abschließend werden die Komplexität und die Größe des Lösungsraumes abgeschätzt.

In Kapitel 5 werden neue anwendungsreife und leistungsfähige Verfahren zur Kostenoptimierung von Topologien entwickelt. Es werden mehrere Optimierungsverfahren sowohl zur Positionierung von Vermittlungsknoten und Bündelungspunkten als auch zur Erstellung baumförmiger und ausfallsicherer Vernetzungsstrukturen entwickelt. Zudem wird ein Verfahren zur Bestimmung einer Kompromisslösung entwickelt, welches den Zielkonflikt zwischen Minimierung der Anzahl der Vermittlungsknoten und Minimierung der Leitungskosten löst.

In Kapitel 6 werden die entwickelten Verfahren evaluiert. Hierfür werden Probleminstanzen unterschiedlicher Größe zufällig erzeugt und deren Topologie mit den entwickelten Verfahren optimiert. Neben der Qualität der Lösungen wird auch der Rechenaufwand betrachtet. Es werden die Ergebnisse der Verfahren mit gleichem Anwendungszweck verglichen. In Kapitel 7 wird die Anwendungsreife der entwickelten Optimierungsverfahren anhand exemplarischer, praxisrelevanter Fragestellungen gezeigt. Abgeschlossen wird die Arbeit in Kapitel 8 mit einer Zusammenfassung und einem Ausblick.

Teilweise wurden wesentliche Aussagen und Ergebnisse der vorliegenden Arbeit vom Autor in mehreren von Fachleuten begutachteten Artikeln auf internationalen Konferenzen [1–3] und in Fachzeitschriften [4, 5] veröffentlicht. Eine Übersicht über die Herausforderungen und Lösungsansätze bei der Kostenoptimierung von Topologien wurde vom Autor in [6] präsentiert. Der Autor hat an mehreren zu dieser Arbeit methodisch verwandten, veröffentlichten Arbeiten [7–11] mitgewirkt. Teilweise wurden die Verfahren unter führender Anleitung des Autors in studentischen Arbeiten [12–16] implementiert, getestet und bewertet.

2 Eingebettete Kommunikationsnetze

Dieses Kapitel widmet sich den Grundlagen eingebetteter Kommunikationsnetze. In Abschnitt 2.1 werden die wesentlichen Begriffe eingeführt und eingeordnet. Anschließend werden die Bereiche und Umgebungen diskutiert, in die heutzutage Kommunikationsnetze typischerweise eingebettet sind. In den Abschnitten 2.3 und 2.4 werden die Eigenschaften eingebetteter Kommunikationsnetze sowie deren funktionale und nichtfunktionale Anforderungen beschrieben. Dadurch erfolgt auch eine Abgrenzung zu klassischen Kommunikationsnetzen. In Abschnitt 2.5 werden denkbare Topologien mit ihren Vor- und Nachteilen diskutiert. Zur Vernetzung verteilter eingebetteter Systeme existieren verschiedene Technologien, welche in Abschnitt 2.6 beschrieben und miteinander verglichen werden. Abgeschlossen wird dieses Kapitel mit einem Ausblick über die aktuellen Entwicklungen und Trends in Abschnitt 2.7.

2.1 Begriffe und Einordnung

Heutzutage werden in viele technische Systeme, Prozesse und Umgebungen elektronische Rechner oder Computer eingebettet. Diese *eingebetteten Systeme* (engl. *embedded systems*) übernehmen komplexe Steuerungs-, Regelungs- und Datenverarbeitungsaufgaben. Technische Systeme werden dadurch flexibler, leistungsfähiger und es entstehen innovative Dienste mit erheblichem Mehrwert. Durch die Einbettung sind die Systeme für die Umwelt und für den Benutzer weitestgehend unsichtbar bzw. werden nicht direkt von außen wahrgenommen. Sie werden speziell für eine Aufgabe entworfen und führen dedizierte Funktionen innerhalb des Gesamtsystems aus [17].

Häufig sind eingebettete Systeme mit der physikalischen Umwelt verbunden. Über Sensoren sammeln sie (Umgebungs-) Informationen, verarbeiten diese und nehmen über Aktoren Einfluss auf ihre Umwelt [18]. Eingebettete Systeme, die nach diesem Prinzip arbeiten, werden auch *Steuergeräte* genannt. Der Einfluss auf die Umwelt erfolgt oft automatisch und ohne menschliche Mitwirkung. Aus diesem Grund müssen eingebettete Systeme, insbesondere in sicherheitskritischen Bereichen, verlässlich funktionieren.

Bis vor einigen Jahren wurden für eingebettete Systeme ausschließlich speziell angepasste Hard- und Software eingesetzt. Mittlerweile werden zunehmend auch *Massenmarkt*-Hardware und Standardkomponenten, beispielsweise aus dem Bereich der Personal Computer (PC), eingesetzt. Solche Systeme werden als *Embedded-PCs* bezeichnet. Durch die breite Produktpalette und die Wiederverwendung von Standardkomponenten können die Kosten gesenkt werden. Häufig unterliegt die Hardware durch die raue Einsatzumgebung besonderen Anforderungen

hinsichtlich Robustheit sowie Temperatur- und Schmutzverträglichkeit. In diesem Fall werden robuste *Industrie-PCs* eingesetzt, die weitestgehend ohne bewegliche, mechanisch angetriebene Komponenten ausgestattet sind und Steckverbindungen möglichst vermeiden.

Aufgrund der Mikrominiaturisierung von Schaltkreisen und Komponenten, der zunehmenden Integrationsdichte von Halbleiterbauelementen auf Chips [19] und sinkender Herstellungskosten in den letzten Jahrzehnten ist die Anzahl der in einem komplexen technischen System eingebetteten Systeme erheblich gestiegen. Anfangs waren die eingebetteten Systeme autonom (*engl. stand-alone*) und interagierten lediglich mit ihren angeschlossenen Sensoren und Aktoren. Es bestand keine Vernetzung zwischen den eingebetteten Systemen. Um technische Systeme und Prozesse effizienter zu gestalten sowie mehr Sicherheit und Komfort zu ermöglichen, mussten die eingebetteten Systeme miteinander kommunizieren. Dazu wurden sie mit den notwendigen Schnittstellen und Fähigkeiten ausgestattet und direkt mittels Punkt-zu-Punkt-Verbindungen miteinander vernetzt [20]. Durch die Zunahme des Informationsaustausches erhöhte sich der Verkabelungsaufwand, die Zahl der Steckverbindungen und damit die Komplexität erheblich [21].

Aus Platz-, Ressourcen- und Kostengründen kann nicht jedes eingebettete System mit jedem verbunden werden. Deshalb wurden Kommunikationsnetze in technische Systeme eingebettet, die die eingebetteten Systeme miteinander vernetzen und damit die Kommunikation, den Informationsaustausch und die Nutzung gemeinsamer Ressourcen transparent ermöglichen. Transparent bedeutet, dass die Systeme sich nicht darum kümmern müssen, mit welchen Verfahren, Netzelementen und Medien die Informationen übermittelt werden [22]. Die eingebetteten Systeme, die miteinander über ein Netz kommunizieren, werden in dieser Arbeit *Endsysteme* oder in Anlehnung an die Terminologie von Kommunikationsnetzen *Endknoten* genannt.

Dank der Vernetzung wurden in den letzten Jahren neuartige Anwendungen und Systeme realisiert. Beispielhaft sind komplexe mechatronische Systeme wie Fahrerassistenzsysteme, bei denen Mechanik, Elektronik und Informationsverarbeitung kombiniert werden. Auch zukünftig sind eingebettete Netze in vielen Bereichen die technologische Voraussetzung (*engl. enabling technology*) für innovative, verteilte Anwendungen, Dienste und Systeme.

Abschließend kann ein eingebettetes Kommunikationsnetz wie folgt definiert werden:

“Ein eingebettetes Kommunikationsnetz ist ein Kommunikationsnetz, das in ein anderes technisches System oder in einen Prozess eingebettet ist, nach außen nicht direkt als Kommunikationsnetz erkennbar ist, verteilte eingebettete Systeme vernetzt und Kommunikationsbasisdienste bereitstellt.”

2.2 Anwendungsbereiche

Heutzutage werden Kommunikationsnetze in technische Systeme und Umgebungen unterschiedlichster Bereiche eingebettet. Im Folgenden werden die wesentlichen Anwendungsbereiche und deren Eigenschaften in Anlehnung an [18] beschrieben.

Fahrzeuge (Automobilbereich) Moderne Fahrzeuge verfügen über mehr als 70 Steuergeräte [11], die miteinander vernetzt sind. Aufgrund der stufenweisen Einführung von

Kommunikationsnetzen sind Domänen entstanden, die jeweils spezifische Bereiche bzw. funktionale Gruppen umspannen. Solche Gruppen sind beispielsweise der Antriebsstrang, das Chassis oder das Infotainmentsystem im Fahrgastraum. Die einzelnen Domänen werden über Koppellemente miteinander verbunden. Die fahrzeuginternen Kommunikationsnetze tragen wesentlich zur Sicherheit und zum Komfort bei. Aufgrund der Massenproduktion spielen die Kosten eine außerordentliche Rolle. Einsparungen in der Produktion oder in der Wartung von nur wenigen Prozent können Kosteneinsparungen in Milliardenhöhe bewirken [23].

Flugzeuge und Avionik Im Hinblick auf Komplexität und Anzahl der Anschlüsse zählt das Flugzeug zu den anspruchsvollsten Anwendungsbereichen der Elektronik. Ein Flugzeug verfügt über mehr als 5000 Steuergeräte [24]. Die Verlässlichkeit der Systeme hat höchste Priorität. Einen Großteil des Gesamtwertes eines Flugzeuges machen heute die elektronischen Systeme aus [18]. Das Kommunikationsnetz des Airbus A380 ist ebenfalls in Domänen, ähnlich wie im Fahrzeug, untergliedert. Beispielsweise existiert eine Domäne für das Cockpit, eine für die Kabine und eine für das Fahrwerk.

Eisenbahntechnik Bei Lokomotiven, Zügen und stationären Sicherheitssystemen sind die Bedingungen und Anforderungen ähnlich wie in Fahrzeugen und Flugzeugen [18]. Die eingebetteten Kommunikationsnetze tragen ebenfalls maßgeblich zum Komfort und zur Sicherheit bei. Genauso spielt auch in diesem Bereich die Verlässlichkeit eine außerordentliche Rolle.

Industrielles Umfeld (Fertigungs-, Prozess- und Automatisierungstechnik) In diesem Bereich werden bereits seit Jahrzehnten Kommunikationsnetze eingesetzt. Anfangs kommunizierten nur einfache Mess- und Stelleinrichtungen einer Anlage miteinander. Später wurden diese Einrichtungen leistungsfähiger und an übergeordnete Leiteinrichtungen angeschlossen. Heutzutage sind in einer modernen Fabrik Zellen-, Fertigungs- und Prozessrechner miteinander verbunden und steuern und regeln komplexe Anlagen und Fertigungsstraßen [25]. Die Anforderungen hinsichtlich Leistungsfähigkeit, Einsatzumgebung und Kosten sind jedoch sehr differenziert.

Gebäudeautomatisierung und Heimvernetzung In intelligenten Gebäuden (*engl. smart buildings*) werden Kommunikationsnetze eingesetzt, um die Wirtschaftlichkeit und Sicherheit zu erhöhen, dem Nutzer einen höheren Komfort zu ermöglichen und die Energiekosten zu senken [26, 27]. Hierfür werden Verbrauchserfassungs- und Sicherheitssysteme, Klimaanlage und Lichtsteuerungen miteinander vernetzt. Die Anzahl vernetzter Endsysteme kann von wenigen bis zu mehreren Tausend reichen [28].

Medizintechnik Heutige medizinische Geräte sind mit einer Vielzahl an Sensoren, Aktoren und Steuergeräten ausgestattet. Beispielsweise muss bei einem Computertomographen das eingebettete Kommunikationsnetz hohe Übertragungsraten unterstützen, um die verschiedenen sensorisch erfassten Daten verlust- bzw. fehlerfrei an ein bildverarbeitendes System zu übertragen.

Militärische Systeme und Satelliten Eingebettete Kommunikationsnetze finden sich auch in komplexen militärischen Ausrüstungsgegenständen und Satelliten wieder. So werden in einem Kampffahrzeug Bilder von einer Kamera erfasst und zur Auswertung an einen leistungsfähigen, datenverarbeitenden Rechner übertragen. Unbemannte Kampffahrzeuge verfügen zur selbst-

ständigen Steuerung über leistungsfähige Rechner, die mit einer Vielzahl an Sensoren und Aktoren vernetzt sind.

Robotik Seit Anfang der Robotik werden Sensoren, Aktoren und Steuergeräte miteinander vernetzt und bilden so ein eingebettetes Kommunikationsnetz. Die Anforderungen, die Eigenschaften und die Anzahl der Komponenten hängen vom Einsatzgebiet und der Größe des Roboters ab.

2.3 Eigenschaften

Eingebettete Kommunikationsnetze ermöglichen eine transparente Kommunikation zwischen Endsystemen. Typischerweise sind diese Netze für die Umwelt und für den Benutzer weitestgehend unsichtbar. Durch die feste Einbettung ist das Kommunikationsnetz in seiner Struktur größtenteils statisch. Insbesondere bei Serienprodukten wird die ursprüngliche Struktur des Systems während der Lebensdauer nicht verändert. Modifikationen nach der Installation bzw. der Auslieferung verursachen erhebliche Kosten. Aus diesem Grund sind der Entwurf und die Dimensionierung von außerordentlicher Bedeutung. Eingebettete Kommunikationsnetze haben gegenüber anderen technischen Produkten, beispielsweise aus dem Bereich der Konsumer- und Unterhaltungselektronik, eine wesentlich längere Lebensdauer von typischerweise zwei Jahrzehnten und mehr [23]. Die langfristige Verfügbarkeit der Komponenten muss also gewährleistet sein.

Eingebettete Kommunikationsnetze sind in der Regel abgeschlossen und nicht *öffentlich* zugänglich. Die Anwendungen und Dienste sowie das von ihnen erzeugte Verkehrsaufkommen sind meistens bekannt und definiert. Falls Schnittstellen nach außen existieren, wird der externe Verkehr mittels geeigneter Mechanismen geregelt. An den Zugangspunkten können die Datenraten und ihre Spitzenwerte begrenzt werden. Dieser Ansatz wird als *Traffic-Shaping* bezeichnet [29]. Auf diese Weise werden unerwünscht hohe Verkehrslasten vermieden und das Netz kann die definierten Dienstgüteanforderungen erfüllen.

Im Vergleich zu Unternehmens- und Weitverkehrsnetzen werden eingebettete Netze weitestgehend isoliert und autonom betrieben. Es existiert keine administrative Instanz, die den operativen Betrieb organisiert, administriert und überwacht. Die Diagnose (Inspektion) und Wartung erfolgt zu definierten Zeitpunkten durch einen (externen) Instandhalter, jedoch nicht im operativen Betrieb. Im Fehlerfall kann das Netz vollständig ausfallen bzw. bei sicherheitskritischen Diensten automatisch in einen Notbetrieb wechseln. In diesem Zustand sind nur eingeschränkte Funktionen verfügbar. Die Wiederherstellung der vollen Funktionsfähigkeit erfolgt durch einen Instandhalter. Dabei werden das Kommunikationsnetz bzw. das technische System gegebenenfalls vollständig abgeschaltet und die defekten Komponenten in Stand gesetzt. Im Anschluss an die Instandsetzung ist das technische System wieder voll funktionsfähig. Unternehmens- und Weitverkehrsnetze hingegen werden normalerweise nicht vollständig abgeschaltet. Bei Ausfall einer Komponente werden die Verkehrsströme geeignet umgeleitet. Ein weiterer Unterschied ist die Abrechenbarkeit. Diese spielt in eingebetteten Netzen keine Rolle; hingegen ist dies in Weitverkehrsnetzen eine zentrale Aufgabe des Netzmanagements.

Verglichen mit herkömmlichen, lokalen Netzen weist der Datenverkehr in einem eingebetteten Netz eine andere Charakteristik auf. Ein eingebettetes System versendet häufig nur wenige Datenwörter zyklisch mit einer kurzen Periode (teilweise < 1 ms). Demzufolge ergeben sich Nachrichten mit kurzen Nutzdatenlängen. Es wird zwischen zeit- (*engl. time*) und ereignisbasierten (*engl. event triggered*) Nachrichten unterschieden. Kontinuierliche (Sensor-) Daten wie Druck und Geschwindigkeit werden unter Beachtung des Abtasttheorems zyklisch (zeitbasiert) erfasst und übertragen. Hingegen werden ereignisbasierte Nachrichten nur bei Änderung eines Zustandes oder bei Überschreitung eines Schwellwertes übertragen [30].

In lokalen Netzen kommuniziert ein Dienstanutzer (*engl. client*), beispielsweise ein Arbeitsplatzrechner, mit einem Diensterbringer (*engl. server*). Es existiert eine eindeutige Trennung zwischen diesen beiden. Der Server selbst kommuniziert wiederum in der Rolle eines Client mit einem anderen Server. So ergibt sich eine hierarchische Struktur, in der die Informationsströme auf den einzelnen Ebenen aggregiert werden. In eingebetteten Kommunikationsnetzen ist die Struktur eher flach und die vernetzten eingebetteten Systeme tauschen Informationen direkt miteinander aus. Dieses Prinzip der Kommunikation wird als *Peer-to-Peer-Kommunikation* (Kommunikation unter gleichgestellten Einheiten) bezeichnet [29]. Jedes System kann gleichzeitig Diensterbringer und Dienstanutzer sein. Es existiert keine feste Rollenzuteilung [31]. Häufig erfolgt die Kommunikation nach dem *Publish-Subscribe* Prinzip [32]. Eine Quelle, der Verteiler (*engl. publisher*), verteilt selektiv Informationen als *Multicast*-Nachrichten an mehrere Empfänger, die Abonnenten (*engl. subscribers*). Systemweite Informationen, beispielsweise zur Prozesssynchronisation, werden nach dem Erzeuger-Verbraucher- (*engl. producer-consumer*) Prinzip verteilt. Dabei versendet eine Quelle, der Erzeuger, Informationen als *Broadcast*-Nachrichten an alle Systeme innerhalb des Netzes. Diese entscheiden mittels (Filter-) Kriterien, ob eine Nachricht relevant ist. Verbraucher sind also Anwendungen höherer Ebene und keine physikalischen Systeme [31].

Die Kosten eingebetteter Kommunikationsnetze berechnen sich aus den Beschaffungs-, Installations- und Betriebskosten. Bei Serienprodukten sind wegen hoher Stückzahlen die Beschaffungs- und Installationskosten von Bedeutung. Hingegen sind bei Individualösungen wie der Gebäudeautomatisierung und bei Fabriksteuerungen die Betriebskosten wichtiger.

Aus Wirtschaftlichkeitsgründen werden einzelne physikalische Leitungen eines eingebetteten Kommunikationsnetzes in einem Leitungsstrang gebündelt. Auch können mehrere Bündel zusammengefasst werden. Die Bündel werden in Leitungs- bzw. Kabelkanälen verlegt. Die Bündelung erfolgt häufig durch Schläuche, Kabelbinder oder Bindegarn. Zusammenhängende Leitungsbündel werden als *Kabelbaum* bezeichnet. An den Endpunkten des Kabelbaumes befinden sich die anzuschließenden Komponenten bzw. die entsprechenden Stecker. Die Vorteile von Kabelbäumen sind [33]:

- Bessere Verbaubarkeit (während der Produktion) und Kostenreduzierung durch standardisierte Installationsprozesse
- Geringere Fehlerrate beim Anschluss von Komponenten
- Gemeinsame Isolierung bietet einen gemeinsamen Schutz vor Störeinflüssen und Abrieb, reduziert das Gewicht und den benötigten Bauraum und verringert potentielle Störaussendungen

- Einfachere Wartbarkeit, da Leitungen ausschließlich innerhalb von (definierten) Kanälen verlegt sind

Der Entwurf wirtschaftlicher Kabelbäume ist äußerst komplex. So ist der Kabelbaum eines Automobils der Oberklasse ungefähr 3,2 km lang und umfasst ungefähr 1500 Leitungen [33]. Aufgrund von Problemen mit den Kabelsträngen verzögerte sich sogar die Auslieferung des Airbus A380 um mehrere Monate [34].

Typischerweise unterliegen eingebettete Kommunikationsnetze starken Beanspruchungen und sind in rauen Umgebungen mit Temperaturen von -40 bis über 100°C, hoher Luftfeuchtigkeit, Schadgasen und mechanischen Störeinflüssen installiert [35]. Dies hat Auswirkungen auf die Beschaffungs- und Installationskosten. Innerhalb der Umgebung existieren Bereiche, in denen es unter Umständen aufwändig und folglich teuer ist, Komponenten und Leitungen zu installieren. In Bereichen mit extremer Umgebungstemperatur werden Komponenten mit einem entsprechenden (Schutz-) Gehäuse benötigt. Wegen starker elektromagnetischer Felder müssen gegebenenfalls Leitungen mit einer besseren Abschirmung installiert werden. Es existieren auch Bereiche, in denen es unmöglich ist, Komponenten zu verbauen oder Leitungen zu installieren. In anderen Bereichen hingegen kann die Installation wesentlich geringere Kosten verursachen. Die Kosten sind damit positionsabhängig.

Die Kosten hängen neben dem Verbauort der Komponenten und der Leitungen auch von deren Größe und Gewicht ab. In den meisten technischen Systemen ist der verfügbare Platz stark limitiert. Deswegen werden häufig die Komponenten speziell für die Zielumgebung gefertigt und passgenau installiert. Zunehmend wird auch versucht, die Anzahl physikalischer Komponenten durch Virtualisierung zu verringern. Bei diesem Ansatz teilen sich mehrere unabhängige Anwendungen die Ressourcen einer leistungsfähigen Komponente.

2.4 Anforderungen

Anforderungen legen die zu erfüllenden qualitativen und quantitativen Eigenschaften oder die zu erbringenden Leistungen eines Systems oder Prozesses fest [36]. In der Regel werden Anforderungen nach funktionalen und nichtfunktionalen Anforderungen klassifiziert. Eine funktionale Anforderung beschreibt eine Funktion oder Tätigkeit, die das System oder der Prozess ausführen soll. Nichtfunktionale Anforderungen repräsentieren keine Funktion [37] und legen die Eigenschaften eines Systems oder Prozesses fest.

2.4.1 Funktionale Anforderungen

Ein bedarfsgerechter Entwurf eines eingebetteten Kommunikationsnetzes erfordert die Identifikation aller funktionalen Anforderungen. In diesem Abschnitt werden die wesentlichen Anforderungen, die im Einzelfall unterschiedlich behandelt werden können, beschrieben. Nicht in allen Anwendungsbereichen existieren identische Anforderungen. Eine grundlegende funktionale Anforderung an ein eingebettetes Kommunikationsnetz ist das Ermöglichen von Kommunikation, d. h. [38]

- Aufbau einer Verbindung zwischen zwei Kommunikationsteilnehmern
- Übertragen (Transport) von Informationen
- Adressierungs- oder Nummerierungsschema zur eindeutigen Identifizierung der Kommunikationsteilnehmer

Für das Übertragen von Informationen muss das eingebettete Kommunikationsnetz über ausreichend Transportleistung verfügen. Das Maximum an Transportleistung, das erreicht werden kann, wird Datenübertragungsrate genannt [39].

Des Weiteren sollte ein eingebettetes Kommunikationsnetz eine Menge von Basisdiensten bereitstellen. Diese umfassen Funktionen zur Kommunikationssteuerung und zum Datentransport sowie Dienstgütemechanismen [38]. Wegen der meist rauen Umgebung sollte das Kommunikationsprotokoll einfach und robust sein sowie Mechanismen zur Fehlererkennung unterstützen. Geeignet hierfür sind Prüfsummenverfahren wie die zyklische Redundanzprüfung.

Eingebettete Kommunikationsnetze übertragen unter anderem zeitsensitive Informationen und müssen folglich Echtzeitanforderungen erfüllen. Häufig wird eine definierte Antwortzeit innerhalb eines vernetzten Systems gefordert. Jasperneite [40] definiert den Begriff *Echtzeitfähigkeit* folgendermaßen:

“Ein System ist dann echtzeitfähig, wenn eine zu erbringende Funktion innerhalb einer durch die Umgebungsbedingungen vorgegebene Zeitspanne rechtzeitig ausgeführt wird.”

Die Anforderungen sind also vom Kontext abhängig und bedeuten nur, dass die Übertragung und Verarbeitung von Informationen lediglich schneller als die Dynamik des technischen Prozesses sein muss. Echtzeitfähigkeit ist somit eine Frage der Definition [41]. Jedoch kann die Nichterfüllung von Echtzeitanforderungen in sicherheitskritischen Bereichen zu verheerenden, sogar lebensbedrohlichen Situationen führen. Eine elektronisch gesteuerte Bremse in einem Fahrzeug muss nahezu unverzögert im Millisekundenbereich reagieren. Eine Überschreitung der definierten Zeit ist nicht zu tolerieren. Neben der Übertragungsverzögerung ist zudem die Abweichung der Verzögerung (Jitter) von Bedeutung. Anforderungen an die Übertragungsrate, an die maximale Übertragungsverzögerung und an die Abweichung der Verzögerung werden häufig unter dem Begriff der Dienstgüteanforderungen zusammengefasst. Allgemein bezeichnet die Dienstgüte “quantifizierbare Eigenschaften eines Dienstes; die einzelnen Eigenschaften werden als Dienstgütemerkmal oder -parameter bezeichnet” [39].

Funktionen müssen häufig zyklisch in einer festgelegten Reihenfolge abgearbeitet und die Funktionswerte über das Kommunikationsnetz übertragen werden. Voraussetzung dazu ist zunächst eine systemweit einheitliche und gültige Zeit, auf deren Basis die Funktionen nach globalen Ablaufplänen (*engl. schedules*) abgearbeitet werden können [25]. Aus Kostengründen können nicht alle Teilnehmer mit einer zentralen Uhr verbunden werden. Auch das Installieren einer *genauen* (Atom-) Uhr in jedem Teilnehmer wäre zu teuer. Deshalb wird durch regelmäßiges Synchronisieren der lokalen Uhren eine systemweit einheitliche Zeit geschaffen. Die größte Herausforderung ist, die lokalen Uhren auch ohne eine deterministische Übertragung zu synchronisieren. In der Vergangenheit wurden unterschiedliche Standards zur genauen Zeitsynchronisation vorgeschlagen. Die Bekanntesten sind das Network Time Protocol (NTP) [42], das

Simple Network Time Protocol (SNTP) [43, 44] und der Standard [45] mit dem Namen *Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*.

In Abschnitt 2.3 wurde erwähnt, dass häufig $1:n$ Kommunikationsverbindungen bestehen. Dies bedeutet, dass eine Quelle dieselbe Informationen an n Empfänger versendet (Multicast). Für eine solche Kommunikation müssen geeignete Protokolle unterstützt werden. Ein Publish-Subscribe-Verfahren muss ermöglichen, dass Abonnenten ihr Interesse an einer Information anmelden bzw. abmelden können. Zur Kennzeichnung von Multicast- und Broadcast-Nachrichten müssen ausreichend Adressen reserviert sein. Auch mögliche Vermittlungseinheiten müssen ggf. diese Mechanismen unterstützen.

2.4.2 Nichtfunktionale Anforderungen

Nichtfunktionale Anforderungen haben eine wachsende Bedeutung bei der Entwicklung und dem Entwurf eingebetteter Kommunikationsnetze. Heutzutage sind die nichtfunktionalen Eigenschaften eines Systems oder Prozesses häufig wettbewerbsentscheidend [37]. In diesem Abschnitt werden die wesentlichen nichtfunktionalen Anforderungen an ein eingebettetes Kommunikationsnetz beschrieben.

Eine wesentliche nichtfunktionale Anforderung ist die Verlässlichkeit und Robustheit. Diese gibt an, wie verlässlich die grundlegenden Funktionen erfüllt werden, und umfasst nach [18, 46] folgende Eigenschaften:

- Zuverlässigkeit: Eigenschaft, dass ein System seine Funktion in einem gegebenen Einsatzzeitraum korrekt und ohne Unterbrechung ausführt (kontinuierliche Erbringung der korrekten Funktion) und sich robust gegen Fehler und Störungen verhält [47]. Systeme können durch redundante Auslegung der Teilkomponenten zuverlässig gestaltet werden. Somit kann auch beim Auftreten von Fehlern und bei Ausfällen von Teilkomponenten die Funktion korrekt ausgeführt werden.
- Wartbarkeit: Wahrscheinlichkeit, dass ein ausgefallenes System innerhalb einer definierten Zeitspanne repariert werden kann. Zudem beschreibt es die Fähigkeit, dass Veränderungen am System vorgenommen werden können.
- Verfügbarkeit: Wahrscheinlichkeit, dass ein System korrekt funktioniert. Die Verfügbarkeit gibt den relativen Zeitanteil, in dem ein System betriebsfähig ist bzw. korrekt arbeitet, an [47]. Um eine hohe Verfügbarkeit zu erreichen, müssen sowohl die Zuverlässigkeit als auch die Wartbarkeit hoch sein. Es ist ein Maß für das Bereitsein zum Gebrauch.
- (Funktions-) Sicherheit und Toleranz: Eigenschaft, dass ein ausfallendes System keinen Schaden verursacht bzw. dass katastrophale Folgen für die Umgebung vermieden werden.
- Integrität: Eigenschaft, dass vertrauliche Daten geheim bleiben und dass die Authentizität der Kommunikation gewährleistet ist.

Verlässlich und robust bedeutet auch, dass das eingebettete Kommunikationsnetz eine hohe elektromagnetische Verträglichkeit (EMV) und geringe Störempfindlichkeit besitzt. Externe Einflüsse sollten nicht störend auf das Übertragungsmedium wirken und keine Fehlfunktionen verursachen. Umgekehrt sollte das Kommunikationsnetz selbst auch keine elektromagnetischen Störungen erzeugen und andere elektrische Komponenten damit störend beeinflussen [48].

Für die Wartbarkeit, Diagnose und Fehlerbehebung müssen standardisierte Schnittstellen nach außen bereitstehen. Im Fehlerfall müssen die fehlerhaften Komponenten einzeln austauschbar sein. Aus diesem Grund sollte das Netz modular aufgebaut sein. Standardisierte Schnittstellen sind auch für die Netzkopplung mit Hilfe von Netzkoppelementen notwendig. Schnittstellen zur Kopplung haben unter Umständen eine besondere Schutzfunktion, um das Kommunikationsnetz vor Angriffen, vor unbefugtem Zugriff und vor Überlast zu schützen.

Standardisierte Schnittstellen sind auch zur Integration von Komponenten in das vorhandene Kommunikationsnetz nützlich. Zur Integration und Inbetriebnahme sollten entsprechende Dienste zur Verfügung stehen [25]. Hierzu zählen Funktionen zur (automatischen) Zuweisung von Adressen, symbolischen Bezeichnern und Systemrollen. Die zugrunde liegende Vernetzungstechnologie muss skalierbar hinsichtlich zukünftiger Entwicklungen und Anpassungen sein. Auch die Übertragungsleistung muss mit der Netzgröße wachsen können.

Die Zuverlässigkeit ist insbesondere bei sicherheitskritischen Systemen von Bedeutung. Eine hohe Zuverlässigkeit lässt sich durch eine hohe Ausfall- bzw. Fehlersicherheit aller Einzelkomponenten erreichen. Die Zuverlässigkeit kann aber auch durch den Einsatz redundanter Komponenten erhöht werden. Beispielsweise verfügt der Airbus A380 für die *Fly-by-Wire*-Steuerung über primäre und sekundäre Steuerungssysteme [34]. Im Fehlerfall kann zwischen diesen Systemen umgeschaltet werden. Damit ein Ausfall eines Kommunikationspfades zwischen zwei Knoten kompensiert werden kann, müssen zwischen den Knoten mindestens zwei knoten- und kantendisjunkte Pfade existieren. Hierfür können verschiedene Schutzgrade unterschieden werden [49–51]:

- Bei der **1+1 Absicherung** werden die Informationen zweifach über getrennte Pfade gesendet. Bei Ausfall eines Pfades erreicht die Information den Empfänger über den Ersatzpfad. Dieses Verfahren ist am schnellsten, da im Fehlerfall nicht umgeschaltet und/oder ein neuer Pfad berechnet werden muss. Allerdings wird mindestens die doppelte Übertragungsrate benötigt.
- Bei der **1:1 Absicherung** wird für jeden Pfad ein Ersatzpfad reserviert. Auf diesen wird im Fehlerfall umgeschaltet. Vorteilhaft ist, dass der Ersatzpfad im Normalbetrieb für niedrig priorisierte, sicherheitsunkritische Informationen verwendet werden kann.
- Bei der **1:N Absicherung** sichert ein Ersatzpfad mehrere Primärpfade ab. Somit kann der Ersatzpfad lediglich den Ausfall eines einzelnen Primärpfades absichern.
- Bei der **M:N Absicherung** werden zur Absicherung von N Pfaden insgesamt $M < N$ Ersatzpfade reserviert. Somit kann ein Ausfall von M Pfaden kompensiert werden.

Durch redundante Auslegung von Komponenten kann ein Ausfall kompensiert und somit die Zuverlässigkeit erhöht werden. Häufig jedoch wird Ausfallsicherheit ausdrücklich gefordert und

nicht als Teil der Zuverlässigkeit beschrieben. In diesem Fall kann die Ausfallsicherheit als eine funktionale Anforderung angesehen werden. Eine eindeutige Unterscheidung zwischen funktionalen und nichtfunktionalen Anforderungen ist also nicht immer möglich.

Eingebettete Kommunikationsnetze unterliegen neben der Verlässlichkeit und Robustheit weiteren einschränkenden, nichtfunktionalen Anforderungen wie Wirtschaftlichkeit (minimale Kosten), Performanz sowie geringem Platz- und Ressourcenverbrauch. Um die Kosten, das Gewicht und den benötigten Bauraum zu minimieren, sollte die Menge der benötigten Komponenten minimal sein. Insbesondere in mobilen Produkten muss das Gewicht minimiert werden. Immerhin bedeutet weniger Gewicht auch weniger Energieverbrauch (Kraftstoffverbrauch) und weniger CO₂-Ausstoß. Durch zusätzliches Gewicht entstehen meistens zusätzliche Kosten und die Attraktivität des Produktes sinkt. Im Airbus A380 werden Aluminiumkabel als elektrische Leitungen verlegt. Das soll gegenüber einer herkömmlichen Verkabelung zu einer Gewichtseinsparung von bis zu 300 Kilogramm führen [34].

Auch der Energieverbrauch eingebetteter Systeme selbst und für deren Kommunikation ist in den letzten Jahren zunehmend wichtiger geworden [18, 19]. In mobilen Produkten wird häufig die für den Betrieb notwendige Energie von Batterien geliefert, die aufgrund gleichzeitiger Gewichts- und Platzanforderungen möglichst leicht und klein sein müssen. In manchen Produkten, beispielsweise in Flugzeugen, müssen Ersatzgeneratoren bei einem Ausfall der primären Energieerzeuger die Netzkomponenten mit Energie versorgen [34]. Aus diesem Grund müssen sämtliche Netzkomponenten eine möglichst geringe Leistung aufnehmen, was eine bessere Energieeffizienz bedeutet. Für die Vernetzung muss eine geeignete energiesparende Übertragungstechnik verwendet werden. Mögliche Vermittlungseinheiten müssen geringe Anforderungen hinsichtlich Speicherbedarf und Rechenleistung haben.

In Bereichen der Massen- und Serienproduktion sind die Stückkosten von wichtiger Bedeutung. Der Stückpreis hängt von den Entwicklungs- und Herstellungskosten ab. Je höher die Stückzahl, desto geringer ist der Anteil der Entwicklungskosten je Stück. Bei großen Stückzahlen wird während der Entwicklung viel Aufwand in die Optimierung der herzustellenden Komponenten investiert, um dadurch die Herstellungskosten zu minimieren. Womöglich werden zur Kostenminimierung für die Verkabelung kostengünstige Materialien wie nicht geschirmte, verdrehte Zweidrahtleitungen und geringe Längen eingesetzt [48]. Ebenso wichtig ist die einfache, d. h. kostengünstige Verbaubarkeit bzw. Installation der Komponenten während der Produktion. Die Installationskosten können durch eine Reduzierung fehler- und störungsträchtiger Steckverbindungen minimiert werden [48].

2.5 Netztopologien

Die Topologie in einem Kommunikationsnetz beschreibt die Struktur der Verbindungen zwischen den Netzknoten. Netzknoten können Verbindungs- oder Endknoten sein. Bei den Verbindungsknoten wird zwischen solchen mit Verarbeitungsfunktion und solchen ohne unterschieden. Normalerweise verfügen Verbindungsknoten, die auf Schicht 1 (Bitübertragungsschicht) des Open Systems Interconnection (OSI)-Referenzmodells [52] arbeiten, über keine Verarbeitungsfunktion. Ein Beispiel ist ein Verstärker (*engl. repeater*). Hingegen verfügen Verbindungsknoten, die auf OSI-Schicht 2 (Sicherheitsschicht) und höher arbeiten, über Verarbeitungsfunk-

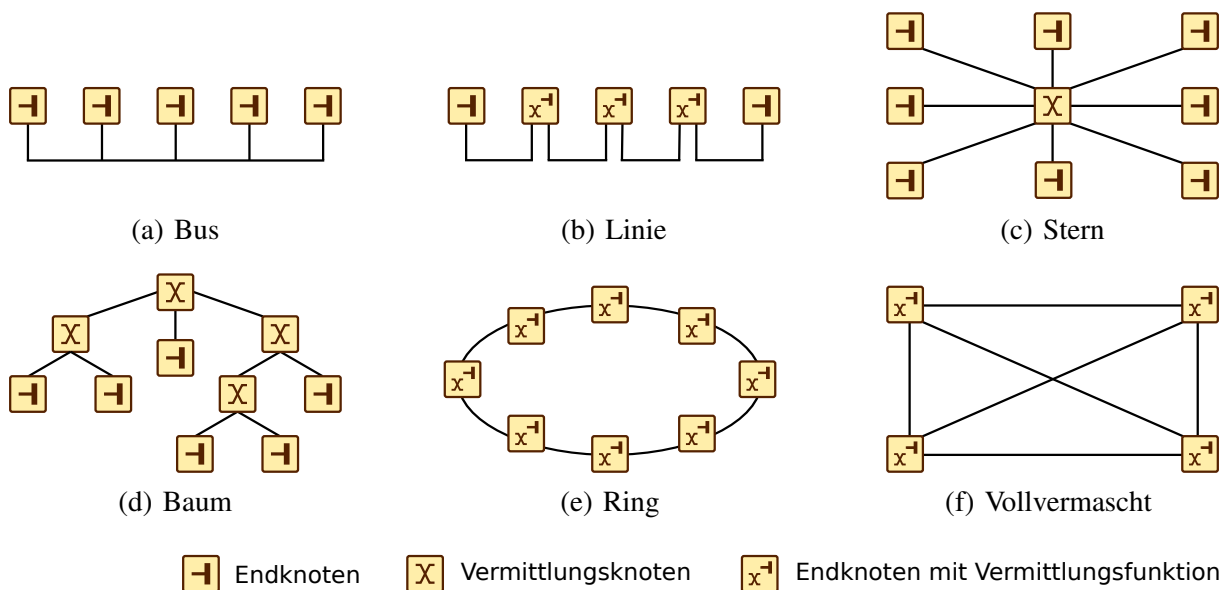


Abbildung 2.1: Überblick Netztopologien

tion. Sie verarbeiten ankommende Informationseinheiten (Nachrichten) und leiten diese nach einem definierten Vorgehen weiter. Solche Verbindungsknoten werden auch Vermittlungsknoten genannt. Ein Vermittlungsknoten ist im Prinzip eine sequentielle Anordnung eines Multiplexers und Demultiplexers mit einer dazwischen geschalteten Weiterleitungseinheit.

Die Verbindungen können durch verschiedene Übertragungsmedien (Kupfer, Glasfaser, Funk) realisiert werden [53]. Topologien werden in Form von Graphen (siehe Kapitel 3.1) beschrieben. Die Topologie ist entscheidend für die Eigenschaften eines Netzes wie Kosten, Leistungsfähigkeit (Ende-zu-Ende Verzögerung, usw.) und Ausfallsicherheit. Grundsätzlich wird zwischen physikalischer und logischer Topologie unterschieden. Die physikalische Topologie beschreibt die Struktur der Verkabelung. Die logische Topologie hingegen beschreibt die Kommunikationsverbindungen und -pfade zwischen den Knoten. Dabei kann die logische Topologie von der physikalischen abweichen. In herkömmlichen Kommunikationsnetzen werden diese beiden Topologien häufig getrennt voneinander betrachtet. In eingebetteten Kommunikationsnetzen unterscheiden sich physikalische und logische Topologie in der Regel nicht.

Im Bereich der eingebetteten Netze, was hinsichtlich der Größe und der Ausdehnung weitestgehend einem lokalen Netz entspricht, existieren grundsätzlich sechs verschiedene Basistopologien: (a) Bus, (b) Linie bzw. Reihe, (c) Stern, (d) Baum, (e) Ring und (f) teil- oder vollvermaschtes Netz. Neben den Basistopologien existieren im Grunde genommen auch Punkt-zu-Punkt-Verbindungen. Eine Punkt-zu-Punkt-Verbindung ist die einfachste Möglichkeit, zwei Knoten miteinander zu verbinden, indem eine Leitung zwischen diesen installiert wird. Diese Verbindungsart wird auch Zweipunktverbindung genannt. In eingebetteten Kommunikationsnetzen wird diese Art typischerweise zur Verbindung zwischen einem Steuergerät und einem Sensor oder Aktor verwendet. Der Nachteil ist allerdings, dass, sobald viele Endknoten miteinander kommunizieren möchten, der Verkabelungsaufwand und die Kosten stark ansteigen.

In Abbildung 2.1 sind die genannten Basistopologien dargestellt. Im Folgenden werden diese mit ihren Vor- und Nachteilen diskutiert.

Bus Bei dieser Topologie werden alle Teilnehmer passiv über sogenannte Stichleitungen an ein gemeinsames Medium angeschlossen [25]. Es existieren keine Vermittlungsknoten zwischen den Endknoten. Da alle Knoten über einen gemeinsamen Bus bzw. über eine gemeinsame Leitung kommunizieren, wird dies auch Mehrpunkt-Verbindung (*engl. multipoint*) genannt. Es werden alle übertragenen Nachrichten von allen Knoten empfangen. Dadurch können alle Knoten direkt mit allen anderen im Netz kommunizieren. Allerdings wird ein geeignetes Medienzugriffsverfahren, das den Zugriff auf das gemeinsame Medium regelt, benötigt, welches auch gleichzeitige Medienzugriffe erkennt und Konflikte auflöst. Der Vorteil einer Bustopologie liegt im geringen Verkabelungsaufwand und in den geringen Kosten. Das Hinzufügen von Endknoten ist in der Regel einfach realisierbar. Des Weiteren hat der Ausfall eines Knotens keine Auswirkungen auf die Funktion des Netzes.

Der Nachteil ist, dass eine Störung des Mediums das gesamte Netz beeinflussen oder blockieren kann. Zudem kann zu einem Zeitpunkt stets nur ein Knoten senden, was eine Blockierung der anderen zur Folge hat. Bei einer großen Anzahl von Knoten kann das gemeinsame Medium stark ausgelastet sein, was zu hohen Sendeverzögerungen führt. Zudem kann bei einer großen Anzahl von Knoten und bei einem Buszugriffsverfahren, das Kollisionen zulässt und auf eine nachträgliche Behebung setzt, durch überproportional viele Kollisionen die effektive Übertragungsrate niedrig sein. Drahtlose Netze können auf der Bitübertragungsschicht ebenfalls als Busstruktur betrachtet werden, da alle Knoten auf dasselbe Medium zugreifen und die Nachrichten von allen anderen empfangen werden.

Linie Bei einer Linientopologie werden bidirektionale Zweipunktverbindungen zwischen zwei Knoten installiert. Dabei wird jeder Knoten mit einem Vorgänger und einem Nachfolger verbunden, mit Ausnahme der beiden Knoten am Anfang und am Ende der Linie. Die zweifach angebotenen Knoten verfügen über Vermittlungsfunktionen und folglich über einen Zwischenpuffer, um empfangene Nachrichten gegebenenfalls an ihre Nachfolger weiter zu leiten. Der Verkabelungsaufwand ist genauso gering wie bei einem Bus, wobei bei einer Linientopologie Kollisionen vermieden werden. Jedoch muss jeder Knoten über zwei Sende- und Empfängermodule verfügen, was höhere Kosten zur Folge hat. Zudem wird durch das Verarbeiten und Weiterleiten der Nachrichten in den Knoten die Latenz erhöht. Ebenso wie bei einem Bus ist das Hinzufügen von Knoten einfach möglich.

Stern Bei der Sterntopologie wird jeder Endknoten über Zweipunktverbindungen mit einem zentralen Vermittlungsknoten verbunden. Da sämtliche Kommunikation zwischen zwei Endknoten über den zentralen Vermittlungsknoten erfolgt, stellt dieser einen Engpass sowie einen *Single-Point-of-Failure* dar. Ein Ausfall des zentralen Vermittlungsknotens bewirkt den Ausfall des gesamten Netzes. Häufig wird der zentrale Vermittlungsknoten redundant ausgelegt und so die Ausfallsicherheit erhöht. Der zentrale Vermittlungsknoten kann mit zusätzlichen Funktionen wie Erkennen fehlerhafter Nachrichten oder automatischem Zuordnen von Adressen zu Anschlüssen ausgestattet sein. Durch geeignete Dimensionierung des zentralen Vermittlungsknotens können Nachrichtenverluste vermieden und ein hoher Durchsatz erreicht werden. Der Verkabelungsaufwand bei einem Stern ist höher als bei einem Bus oder einer Linie. Demgegenüber steht eine höhere Verfügbarkeit bei Verbindungsausfall oder bei Ausfall eines einzelnen Endknotens. In diesen Fällen ist die Kommunikation zwischen den verbleibenden Endknoten nicht betroffen. Das wiederum ermöglicht eine einfache Fehlersuche.

Baum Die Baumtopologie ist eine Fortführung der Sterntopologie. Sie kann als die hierarchische Anordnung mehrerer Sterne gesehen und einfach erweitert werden. Bäume sind dadurch gekennzeichnet, dass sie eine Wurzel haben, von der eine oder mehrere Verbindungen ausgehen und der Weg einer Nachricht stets eindeutig ist, da immer genau ein Weg zwischen zwei Endknoten existiert. Die Zuverlässigkeit des Netzes hängt von der Zuverlässigkeit der Vermittlungsknoten ab. Der Ausfall einer Verbindung nahe der Baumwurzel beeinflusst einen Großteil des Netzes. Hingegen hat der Ausfall eines Endknotens keine Konsequenzen auf das Netz. Die Vermittlungsknoten und die Übertragungsrate von Verbindungen nahe der Wurzel sollten ausreichend dimensioniert werden, da die Kommunikation von einer Baumhälfte zur anderen immer über die Wurzel geht. Bei zunehmender Baumtiefe, d. h. Anzahl der Verbindungen zwischen der Wurzel und einem Endknoten, kann es zu hohen Latenzen kommen.

Ring In einer Ringtopologie ist jeder Knoten mit genau zwei anderen Knoten verbunden, so dass ein geschlossener Ring entsteht. Dies entspricht einer Linie, bei der der erste und letzte Knoten mit einer zusätzlichen Zweipunktverbindung miteinander verbunden werden. Jeder Knoten innerhalb des Ringes verfügt über Vermittlungsfunktionen und folglich über einen entsprechenden Zwischenpuffer. Ringtopologien erfordern Zugriffsprotokolle, um den Zugriff auf den Ring zu steuern und Kollisionen zu vermeiden. Es können sowohl uni- als auch bidirektionale Zweipunktverbindungen eingesetzt werden. Der Vorteil von bidirektionalen Verbindungen ist, dass bei einer Ringunterbrechung die Knoten vor und nach der Unterbrechung den Ring überbrücken können und eine Kommunikation weiterhin möglich ist. Im Falle von unidirektionalen Verbindungen kann ein Verbindungsbruch nicht kompensiert werden. Die meisten Ringimplementierungen garantieren die Einhaltung einer maximalen Zugriffszeit und eine maximale Übertragungsverzögerung. Im Vergleich zu anderen Topologien erfordert ein Ring einen höheren Verkabelungsaufwand.

Teil- und Vollvermaschung Eine allgemeine Verbindungsstruktur stellt die Teil- bzw. Vollvermaschung dar. Dabei sind entweder alle Knoten mit jeweils allen anderen direkt verbunden (Vollvermaschung), wie in Abbildung 2.1(f) dargestellt, oder jeder Knoten ist mit einer ausgewählten Menge anderer Knoten verbunden (Teilvermaschung). Jeder Knoten muss die Nachrichten durch die Wahl der richtigen Verbindung direkt oder indirekt zu einem oder mehreren Empfängern senden. Aus diesem Grund müssen ausgewählte Knoten über Vermittlungsfunktionen verfügen. Bei der Vollvermaschung ist jede Verbindung genau einem Knoten zugeordnet, was zu einem hohen Verkabelungsaufwand führt. Die Zahl der benötigten Verbindungen steigt näherungsweise quadratisch mit der Zahl der Knoten und beträgt bei n Knoten $n(n-1)/2$. Dafür hat ein vollvermaschtes Netz die optimale Verfügbarkeit, da bei Ausfall einer Verbindung durch Umlenkung immer ein alternativer Weg gefunden werden kann. Falls bei einer Teilvermaschung die Wege zwischen den Knoten nicht eindeutig sind, sind Protokolle zur Wegfindung notwendig. Typischerweise haben vollvermaschte Netze wegen des hohen Verkabelungsaufwandes und der hohen Kosten für eingebettete Kommunikationsnetze eine geringe Bedeutung. Betrachtet man die in Abschnitt 2.1 beschriebene Evolution der Vernetzung, so wurde die gewachsene vollvermaschte Struktur durch andere Strukturen ersetzt.

Hybride Strukturen In eingebetteten Netzen findet man häufig komplexe hybride Topologien vor. Diese sind eine Kombination aus den vorgestellten Basistopologien. Oft sind diese komplexen Netze hierarchisch strukturiert, wobei auf jeder anderen Ebene eine andere Topologie verwendet wird. Die einzelnen Ebenen werden durch Netzkoppelemente miteinander ver-

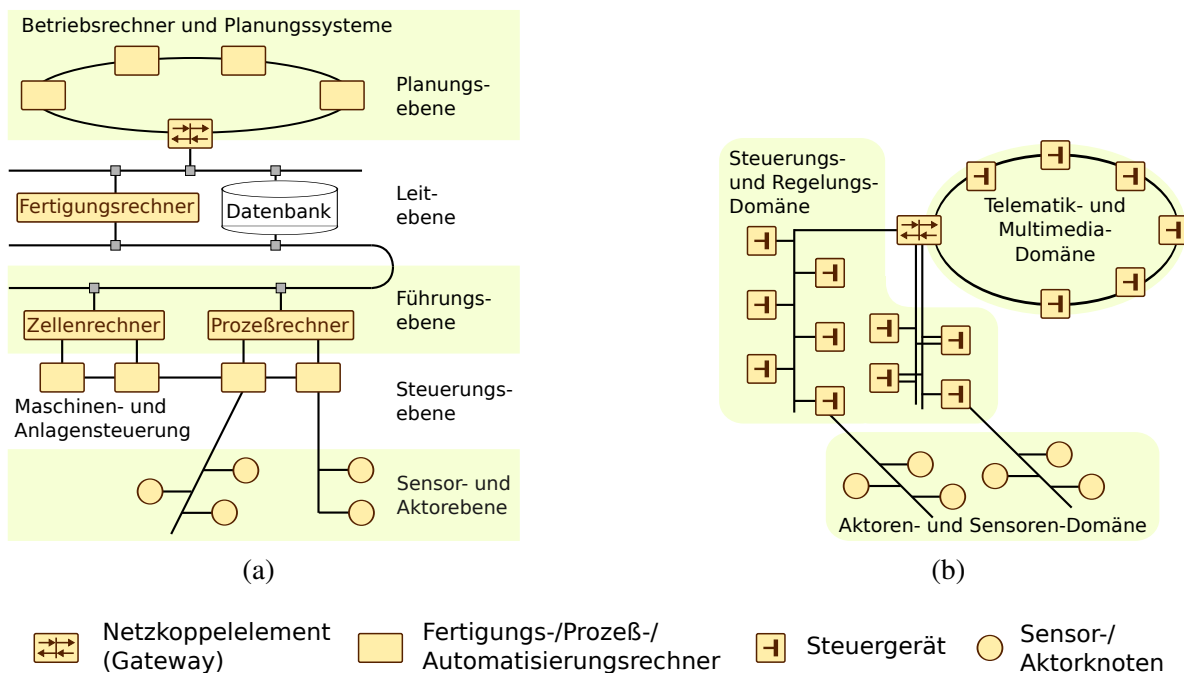


Abbildung 2.2: Beispiele für hybride Netztopologien aus den Bereichen a.) Fertigungs-, Prozess- und Automatisierungstechnik [25, 55] und b.) Fahrzeugvernetzung

bunden. Ein Netzwerkelement muss neben der direkten Umsetzung der Daten auch effiziente Mechanismen zur Adressumsetzung und Protokolladaptation sowie zur Sicherstellung der Dienstgüte und Zuverlässigkeit bereitstellen. Unter Protokolladaptation wird eine Protokollumsetzung bzw. -transformation verstanden, bei der die Dienstprimitiven der Transport- und Verarbeitungsschicht umgesetzt werden. Die Veröffentlichung von Biersack [54] enthält eine umfangreiche und strukturierte Literaturliste mit 380 Veröffentlichungen, Büchern und Dokumenten, die sich mit Kopplungsvarianten und Ausprägungen von Netzwerkelementen beschäftigen.

Ein Netzwerkelement entscheidet zudem, welche Informationen von einer Ebene in eine andere weitergeleitet werden. Es findet damit eine Form der Verkehrslenkung (*engl. routing*) statt [48]. Bei Mehrpunktverbindungen kann durch den gezielten Einsatz von Netzwerkelementen die Komplexität durch Segmentierung verringert und eine Vergrößerung der Zuverlässigkeit erreicht werden. Durch die Segmentierung wird die Zahl der Endknoten und die Netzauslastung reduziert. Ein segmentiertes Netz bietet eine höhere Verfügbarkeit, da bei Ausfall oder Überlast nur die Endknoten eines Segmentes betroffen sind. Ein Netzwerkelement, das auf der Transportschicht oder einer höheren OSI-Schicht arbeitet, wird *Gateway* genannt. Netzwerkelemente, die auf darunter liegenden Schichten arbeiten, werden auf OSI-Schicht 1 *Repeater*, auf OSI-Schicht 2 *Switch* und auf OSI-Schicht 3 *Router* genannt.

Beispiele für hybride Topologien eingebetteter Kommunikationsnetze sind in Abbildung 2.2 dargestellt: a.) ein hierarchisches Netz aus der Fertigungs-, Prozess- und Automatisierungstechnik und b.) ein fahrzeuginternes Netz. Fahrzeuginterne Netze sind in mehrere Domänen aufgeteilt, die über ein oder mehrere Netzwerkelemente miteinander verbunden sind. Wie in Abbildung 2.2(b) ersichtlich, ist ein Teil der Endknoten in der Steuerungs- und Regelungsdomäne redundant angebunden, um eine höhere Ausfallsicherheit zu erzielen.

2.6 Vernetzungstechnologien

Zur Vernetzung verteilter eingebetteter Systeme existieren verschiedene Technologien. Im ersten Teil wird ein Überblick über bekannte (Feld-) Bussysteme gegeben. Dabei werden einige bekannte Vertreter kurz erläutert. Im zweiten Teil werden die Grundlagen, die Historie und die wesentlichen Erweiterungen der Ethernet-Technologie beschrieben. In diesem Teil werden zudem Industrial Ethernet-Lösungen vorgestellt. Abschließend werden (Feld-) Bussysteme und Ethernet miteinander verglichen.

2.6.1 Überblick bekannter (Feld-) Bussysteme

Anfang der achtziger Jahre wurden zur Kommunikation zwischen Steuergeräten in Fahrzeugen und Flugzeugen sowie zwischen (Feld-) Geräten in industriellen Anlagen (*Feld-*) *Bussysteme* entwickelt. Nach der Norm IEC 61158 [56] ist ein Feldbus ein digitaler, serieller Datenbus für die Kommunikation zwischen Systemen der Automatisierungstechnik. Häufig werden diese Systeme auch *industrielle Kommunikationsnetze* genannt. Am Anfang wurden linienförmige Bussysteme konzipiert, an die direkt Endsysteme angeschlossen werden konnten [25]. Später wurden andere Strukturen zur Erhöhung der Zuverlässigkeit standardisiert. Aufgrund teilweise differenzierter Anforderungen in den unterschiedlichen Bereichen wurde in den letzten Jahrzehnten eine Vielzahl serieller Bussysteme entwickelt. In diesem Abschnitt werden exemplarisch einige weit verbreitete Bussysteme vorgestellt. Moderne Bussysteme bieten mehr als reine Kommunikationsfunktionalität und erfüllen größtenteils die in Abschnitt 2.4 genannten Anforderungen. Sie verfügen meistens über standardisierte Applikationen und Funktionsblöcke, branchenspezifische Profildefinitionen und Gerätebeschreibungen [25].

Controller Area Network (CAN) Das serielle Bussystem CAN wurde 1985 in einer Kooperation der Firmen Bosch und Intel entwickelt [57] und ist in der internationalen Norm ISO 11898 [58] standardisiert. CAN hat eine weite Verbreitung und wird überwiegend in Fahrzeugdomänen mit Echtzeitanforderungen eingesetzt. CAN verwendet als Zugriffsverfahren Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) und erkennt somit das Auftreten von Kollisionen während der Arbitrierungsphase. Zur Kollisionserkennung vergleicht jeder Teilnehmer den lokalen mit dem am Bus anliegenden Pegel. Falls ein anderer Teilnehmer gleichzeitig sendet, weichen die beiden Pegel voneinander ab. Die äquivalente Signallänge muss also größer als die Länge des Busses sein. Die Dauer eines Bitintervalls verlängert sich mit zunehmender Buslänge, da die Signallaufzeit proportional zur Buslänge ist. Mit zunehmender Buslänge verringert sich die maximale Übertragungsrate. Je nach Anforderung werden Übertragungsraten von 125, 250, 500 kbits/s oder 1 Mbit/s eingesetzt.

Bei CAN werden nicht die einzelnen Busteilnehmer eines Netzes adressiert, sondern jede Nachricht erhält eine eindeutige Kennzeichnung, den *Message-Identifier*. Dieser wird für zwei wesentliche Mechanismen verwendet: Erstens entspricht der Identifier der Übertragungspriorität: Je kleiner der numerische Wert, desto größer die Priorität. Zweitens unterstützt die eindeutige Kennzeichnung das effiziente Filtern von Nachrichten auf Empfängerseite. Eine Standard-Nachricht im CAN 2.0A Format hat, abhängig von den übertragenen Nutzdaten, eine Nachrichtenlänge von minimal 47 Bit bis maximal 111 Bit, inklusive 11 Bit langen Identifiern. Neben diesem Format existiert ein erweitertes Format CAN 2.0B mit 18 Bit Identifier. Der CAN-

Standard definiert lediglich die Bitübertragungsschicht und die Sicherungsschicht. Für höhere Schichten sind unterschiedliche Mechanismen zur Unterstützung von Nachrichtensegmentierungen u. ä. verfügbar [59].

Process field bus (Profibus) Das offene und herstellerübergreifende Feldbusssystem Profibus wird überwiegend in der Prozess- und Automatisierungstechnik eingesetzt. Der Standard liegt als internationale Norm IEC 61158 [56] vor. Diese Norm umfasst neben dem Profibus-Standard noch weitere Feldbusstandards. Mit Profibus können Systeme auf unterschiedlichen Feldebene(n) (siehe Abbildung 2.2(a)) miteinander vernetzt werden. So wird die Anbindung von Sensoren und Aktoren an Netze der darüber liegenden Steuerungsebene unterstützt. Profibus existiert in drei miteinander kompatiblen Varianten [25]: (a) Profibus-DP (Dezentrale Peripherie) ist auf die Anbindung von Sensoren und Aktoren an ein zentrales Steuerungssystem ausgerichtet. (b) Profibus-PA (Prozess-Automation) wird in explosionsgefährdeten Bereichen eingesetzt. (c) Profibus-FMS (Fieldbus Message Specification) wird zur Vernetzung auf der Führungsebene (siehe Abbildung 2.2(a)) verwendet.

Profibus unterstützt eine Übertragungsrate von bis zu 12 Mbit/s und eine Kabellänge von maximal 1200 Meter. Die maximale Systemanzahl in einem Bussegment beträgt 32. Bei Verwendung von verdrehten Zweidrahtleitungen wird Profibus als Linienstruktur realisiert. Bei optischer Übertragung über Lichtwellenleiter können Stern-Topologien realisiert werden. Zur Datenübertragung unterstützt Profibus drei azyklische Dienste (Sendung ohne Quittung, mit Quittung sowie mit Anforderung und Rückantwort) sowie einen zyklischen Dienst. Das Profibus-Kommunikationsmodell definiert die unteren beiden OSI-Schichten und die oberste OSI-Schicht [25].

Military-Standard-1553 (Mil-Std-1553) Das serielle Bussystem Mil-Std-1553 wird überwiegend in Flugzeugen und Satelliten eingesetzt und ist vergleichbar mit CAN. Der erste Standard Mil-Std-1553A wurde 1975, der Nachfolger Mil-Std-1553B [60] 1978 vom Verteidigungsministerium der Vereinigten Staaten von Amerika veröffentlicht. Erstmals wurde das Bussystem im Kampfjet F-16 und im Kampfhubschrauber AH-64 Apache implementiert [61]. Der Standard definiert die Bitübertragungsschicht und die Sicherungsschicht. Das Bussystem unterstützt eine Übertragungsrate von maximal 1 Mbit/s. Es existieren drei Nachrichtentypen: Daten, Kommando und Status. Eine Nachricht hat eine Länge von 20 Bit, wovon 16 Bit Nutzdaten sind. In den Nutzdaten sind 5 Bit für die Adressierung reserviert, wobei nur die Nachrichtentypen Kommando und Status adressiert werden. Eine Adresse (11111) ist für Broadcast-Nachrichten reserviert [60]. Ein Segment besteht aus einem Bus-Controller und bis zu 31 sogenannten Remote-Terminals. Der Bus-Controller fungiert als zentrale Einheit und steuert die gesamte Kommunikation. Diese erfolgt im Halbduplexbetrieb. Zur Erhöhung der Zuverlässigkeit können redundante Busse optional installiert werden.

KNX Der KNX-Standard [62] ist ein weltweit anerkannter Standard im Bereich der Haus- und Gebäudesystemtechnik. Technisch gesehen ist der KNX-Standard eine Weiterentwicklung des europäischen Installationsbusses (EIB) [63]. KNX ist in der europäischen Norm CENELEC EN 50090 [64] und in der internationalen Norm ISO/IEC 14543 [62] standardisiert. Der Standard definiert unterschiedliche Übertragungsmedien. Dazu zählt auch eine drahtlose Variante. Die Systeme werden über einen Bus miteinander verbunden. Eine KNX-Anlage ist hierarchisch aufgebaut [65]. Ein Gerät wird einer Linie zugeordnet, mehrere Linien werden an eine Hauptlinie angeschlossen und bilden einen Bereich. Unterstützt wird eine Übertragungsra-

Tabelle 2.1: Überblick bekannter (Feld-) Bussysteme

Eigenschaft	CAN	Profibus	Mil-Std-1553	KNX	MOST
Anwendungsbereich	Fahrzeuge	Industrielles Umfeld	Flugzeuge	Gebäudeautomatisierung	Fahrzeuge
Topologie	Bus	Linie oder Stern	Bus	Bus	Ring
max. Übertragungsrate	1 Mbit/s	12 Mbit/s	1 Mbit/s	9,6 kbit/s	150 Mbit/s
max. Teilnehmeranzahl	beliebig, techn. ca. 100	32	31 (+ Bus-Controller)	256	64
Standard	[58]	[56]	[60]	[62]	[66]

te von 9,6 kbit/s. Genauso wie bei CAN wird als Zugriffsverfahren CSMA/CA verwendet. Zur Auflösung einer Kollision wird jedoch neben der Nachrichtenpriorität auch die Geräteadresse berücksichtigt. Jedem Gerät wird eine eindeutige physikalische Adresse zugewiesen. Diese orientiert sich meistens am Einbauort im Gebäude [65]. Im Standard sind drei Nachrichtenarten definiert: Datentelegramm (*engl. data frame*), Abfragetelegramm (*engl. poll data frame*) und Bestätigungstelegramm (*engl. acknowledge frame*). Eine Übersicht über Bussysteme für die Gebäudesystemtechnik ist zu finden in [28, 65].

Media Oriented Systems Transport (MOST) Die MOST-Spezifikation [66] definiert ein serielles Kommunikationssystem zur Übertragung von Audio-, Video-, Sprach- und Datensignalen. MOST ist kein klassischer Feldbus. Aufgrund seiner Bedeutung in der Automobilindustrie, seinem Anwendungsbereich und seinen Anforderungen kann MOST jedoch durchaus zu den (Feld-) Bussystemen gezählt werden. MOST-Netze werden überwiegend in Fahrzeugen zur Vernetzung von Multimedia-, Telematik- und Bordinfotainmentsystemen eingesetzt. MOST150 [66] unterstützt eine Datenübertragungsrate von bis zu 150 Mbit/s und verfügt über drei Übertragungskanäle. Der Steuerkanal (*engl. control channel*) wird für Steuerbefehle sowie Status-, Diagnose- und Gateway-Nachrichten verwendet. Es handelt sich um ereignisbasierte Nachrichten mit kurzer Paketlänge und geringer Datenübertragungsrate. Der synchrone Kanal (*engl. synchronous channel*) wird für den Transport von multimedialen Datenströmen mit hohen Übertragungsraten genutzt. Der asynchrone Kanal (*engl. asynchronous channel*) wird für die Übertragung von Daten paketbasierter Dienste mit großer Paketlänge und hohem Datenaufkommen genutzt. Auf der Bitübertragungsschicht werden die Daten seriell in einer Ringtopologie über einen Lichtwellenleiter übertragen. Die Spezifikation definiert nicht nur die unteren beiden OSI-Schichten. So bieten die MOST Network Services einheitliche Schnittstellen und alle notwendigen Funktionalitäten an, die für den Betrieb eines MOST-Systems notwendig sind [67].

In Tabelle 2.1 sind die soeben behandelten (Feld-) Bussysteme mit ihren wesentlichen Eigenschaften übersichtlich zusammengefasst. Wie bereits erwähnt wurde in den letzten Jahren eine Vielzahl serieller (Feld-) Bussysteme für unterschiedliche Anwendungsbereiche entwickelt. Weitere wesentliche, auf dem Markt verfügbare Systeme sind FlexRay, K-Line, Bitbus, Inter-

bus, SimaticNet, Local Operating Network (LON), WorldFIP, ARINC 429, ControlNet, DeviceNet und Sercos. Darüber hinaus gibt es eine Vielzahl proprietärer Bussysteme und Speziallösungen. Überblicke über bekannte serielle Bussysteme und deren Beschreibung sind zu finden in [25, 59, 68–70].

Neben den Feldbussystemen existieren auch sogenannte Sensor-Aktuator-Bussysteme, die eine Kommunikation zwischen einfachen Geräten wie etwa Schaltern und Motoren und den datenverarbeitenden Einheiten ermöglichen. Diese Bussysteme sind einfach konzipiert und unterstützen nur eine geringe Übertragungsrate von wenigen kbit/s. Die Anzahl angesteuerter Geräte kann groß sein, weshalb die Kosten besonders gering sein sollten. Bekannte Vertreter sind Local Interconnect Network (LIN) [71] aus dem Bereich der Fahrzeugtechnik und Aktor-Sensor-Interface (AS-I) [72] aus dem Bereich der Fertigungs-, Prozess- und Automatisierungstechnik. In den letzten Jahren hat die Leistungsfähigkeit von Sensoren und Aktoren zugenommen. Oft besteht ein logischer Sensor bzw. Aktor aus einer Menge physikalischer Sensoren bzw. Aktoren. Zunehmend werden diese leistungsfähigen Komponenten direkt an ein leistungsfähiges Kommunikationsnetz angeschlossen und kommunizieren so nicht mit einem, sondern mit mehreren datenverarbeitenden, übergeordneten Systemen. Solche Netze werden auch Embedded Sensor Networks [73] genannt.

2.6.2 Ethernet

2.6.2.1 Grundlagen

Im Jahre 1972 entwickelten Metcalfe und seine Kollegen am Forschungszentrum von Xerox in Palo Alto die erste Version von Ethernet. Das Ziel war eine Technologie zur lokalen Vernetzung von Arbeitsplatzrechnern, Servern und Druckern innerhalb eines Gebäudes [74]. Die Kommunikation erfolgte über einen gemeinsamen Bus. Später wurde das ursprüngliche Zugriffsverfahren namens Aloha [75, 76] durch die Ergänzung eines Verfahrens zur Kollisionserkennung verbessert. Das verbesserte Verfahren mit dem Namen Carrier Sense Multiple Access with Collision Detection (CSMA/CD) [77] wurde 1977 patentiert [78]. In Kooperation mit DEC und Intel veröffentlichte Xerox im Jahre 1980 eine modifizierte Version von Ethernet namens *DIX Ethernet II* mit einer maximalen Übertragungsrate von 10 Mbit/s. DIX steht dabei für die an der Kooperation beteiligten Firmen. Mitte der achtziger Jahre standardisierte das Institute of Electrical and Electronics Engineers (IEEE) die unterschiedlichen Versionen von Ethernet. Da IEEE grundsätzlich Markenbezeichnungen in den Namen ihrer Standards vermeidet, wurde die *Technology 802.3 CSMA/CD* genannt [74]. Der dazugehörige IEEE 802.3 Standard [79] definiert sowohl die OSI-Schicht 1 als auch die OSI-Unterschicht 2a und legt damit das Verfahren für den Medienzugriff (*engl. Medium Access Control, MAC*) fest. Im IEEE 802.2 Standard [80] wird für die OSI-Schicht 2b das Logical Link Control (LLC) Protokoll zur Datensicherung mittels eines Verbindungskonzeptes beschrieben. Zusammen definieren die beiden IEEE Standards [79] und [80], wie in Abbildung 2.5 dargestellt, also die beiden unteren Schichten des OSI-Referenzmodelles.

In den letzten Jahren entwickelte sich Ethernet zur weltweit vorherrschenden Vernetzungstechnologie für lokale Netze und ist inzwischen allgegenwärtig. Die Erfolgsfaktoren sind bis heute die Flexibilität, die ständige Weiterentwicklung unter Beibehaltung der Rückwärtskompatibili-

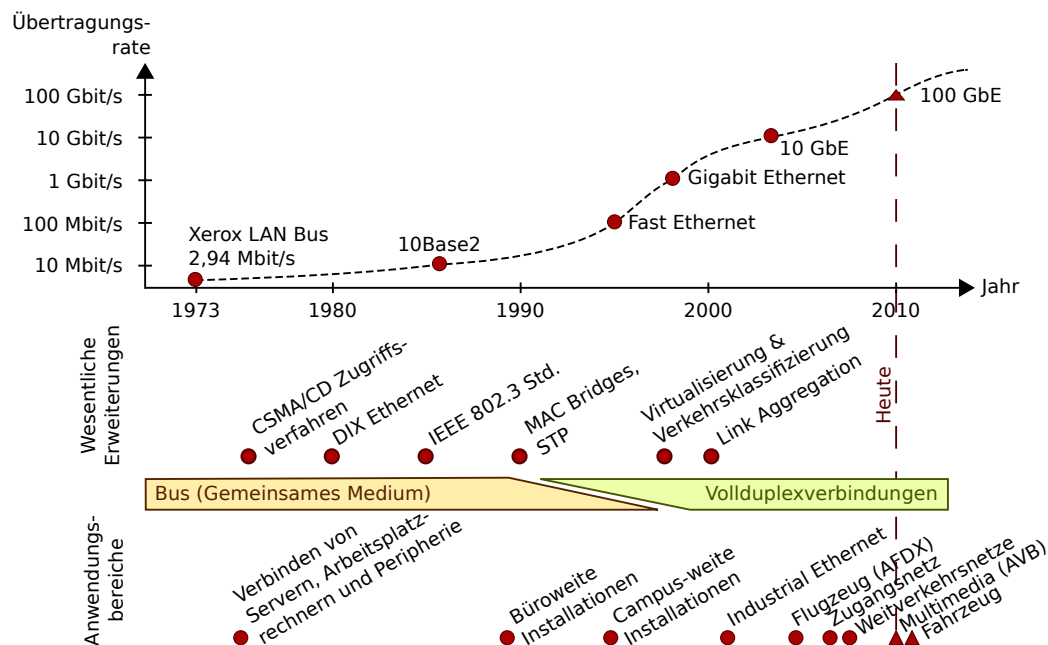


Abbildung 2.3: Die Geschichte von Ethernet, dessen Erweiterungen und Anwendungsbereiche [4]

tät (Abwärtskompatibilität), die Skalierbarkeit, die geringen Kosten und ein einfaches Management [4, 29]. In Abbildung 2.3 ist die Erfolgsgeschichte von Ethernet und dessen Erweiterungen dargestellt. Ursprünglich war eine Übertragungsrate von 2,94 Mbit/s möglich. Mitte der neunziger Jahre wurden 100 Mbit/s (IEEE 802.3u, *Fast Ethernet*) unterstützt; und bereits 2002 10 Gbit/s (IEEE 802.3ae, *10 GbE*). Momentan standardisiert die IEEE Arbeitsgruppe P802.3ba eine Übertragungsrate von 40 Gbit/s (*40 GbE*) und 100 Gbit/s (*100 GbE*) [81].

In Abbildung 2.3 sind zudem die Anwendungsbereiche von Ethernet aufgeführt. Heutzutage wird Ethernet auch als Vernetzungstechnologie im industriellen Umfeld und in Flugzeugen eingesetzt. Seit einigen Jahren werden zudem Zugangs- und Weitverkehrsnetze auf der Basis von Ethernet realisiert. Ethernet wird sich in nächster Zeit weiter verbreiten und weitere Anwendungsbereiche erschließen. Die IEEE Arbeitsgruppe Audio/Video Bridging (AVB) [82] spezifiziert momentan Erweiterungen zur Unterstützung verteilter Multimediaanwendungen (IEEE 802.1Qat [83] und IEEE 802.1Qav [84]). Derartige Anwendungen haben hohe Dienstgütereanforderungen hinsichtlich der maximalen Verzögerung und des Jitters.

Als weiterer Anwendungsbereich für Ethernet wird voraussichtlich die fahrzeuginterne Vernetzung hinzukommen. Der Trend deutet darauf hin, dass Ethernet auch in diesem Bereich zunehmend relevanter wird. Mehrere Arbeiten [85–89] haben die Einsatzfähigkeit von Ethernet im Fahrzeug bereits nachgewiesen. Auf längere Sicht ist auch der Einsatz in sicherheitskritischen Domänen vorstellbar [85].

Die in Abbildung 2.3 aufgeführten Anwendungsbereiche werden vom Autor et al. in [4] umfassend diskutiert und verglichen. Die Entwicklung der Ethernet-Standardisierung wird in [90] beschrieben. Um die Anforderungen in den einzelnen Anwendungsbereichen zu erfüllen, wurden die IEEE 802-Standards um spezifische Erweiterungen angepasst. Ebenfalls in [4] werden

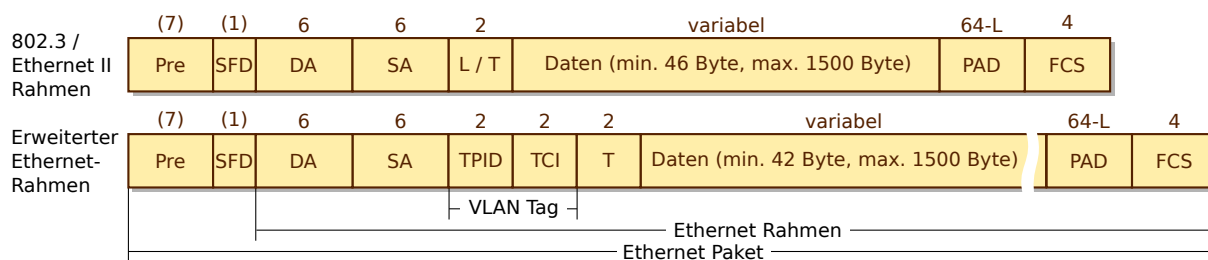


Abbildung 2.4: Rahmenformate von Ethernet

diese Erweiterungen aufgeführt und es wird diskutiert, was unter dem Begriff *Ethernet* heutzutage überhaupt zu verstehen ist.

Was bei Ethernet bis heute unverändert geblieben ist, ist die paketvermittelnde Kommunikation, wobei die Pakete als Rahmen (*engl. frames*) bezeichnet werden. Der Aufbau eines Ethernet-Rahmens ist in Abbildung 2.4 dargestellt. Die Übertragung eines Rahmens wird mit einer sieben Byte langen Präambel (*engl. preamble, Pre*) und dem Rahmenanfangsbegrenzer (*engl. start of frame delimiter, SFD*) gekennzeichnet. Die Präambel wird von jedem (Vermittlungs-) Knoten zur Sicherstellung der korrekten Übertragung neu aufgebaut [22]. Nach dem SFD beginnt der eigentliche Rahmen. Dieser startet mit einer 6 Byte langen Zieladresse (*engl. destination address, DA*) und eine 6 Byte langen Quelladresse (*engl. source address, SA*¹). Diese Adressen werden MAC-Adressen genannt, sind weltweit eindeutig und werden bereits durch den Hersteller festgelegt. Die nachfolgenden 2 Byte werden abhängig vom verwendeten Standard unterschiedlich interpretiert. In den IEEE 802-Standards geben diese Bytes Auskunft über die Länge L des Dateninhaltes (Längenfeld). Bei Ethernet II dienen diese Bytes zur Kennzeichnung des verwendeten Protokolls der nächsthöheren Schicht innerhalb der Nutzdaten (Typenfeld T)². Als nächstes folgen die Nutzdaten mit einer maximalen Größe von 1500 Byte. Aus Gründen der Kollisionserkennung beträgt die minimale Nutzdatengröße 46 Byte. Falls die Nutzdaten weniger als 46 Byte sind, füllt das Feld PAD, d. h. Padding (*dt. Füllung*), den Rahmen bis zur Minimalgröße auf $(64 - L)$ Byte). Da für das Typenfeld nur Werte ab $0x600$ (1536_{10}) definiert sind [91], kann jeder Wert kleiner oder gleich 1500 als Länge und jeder Wert über 1536 als Typ interpretiert werden. Heutzutage werden überwiegend Ethernet II Rahmen versendet [4]. Das letzte Feld im Rahmen ist die Prüfsummensequenz FCS (*engl. frame check sequence*). Damit können Übertragungsfehler mit einer gewissen Wahrscheinlichkeit erkannt werden. Im Verlauf der letzten Jahre wurde das Rahmenformat um domänenspezifische Anforderungen erweitert. Ein bekanntes Beispiel sind die Erweiterungen des Carrier Grade Ethernet für die Vermittlung von Rahmen in Weitverkehrsnetzen [4].

¹Die Abkürzung SA für die Quelladresse ist in der Literatur üblich, beispielsweise in den IEEE 802-Standards. Allerdings wird in anderen Teilen der Arbeit mit dieser Abkürzung die Metaheuristik *Simulated Annealing* bezeichnet.

²Die Abkürzungen L und T sind in der Literatur üblich. Allerdings bezeichnen in anderen Teilen der Arbeit L eine endliche Menge zulässiger Lösungen und T eine Tabuliste.

2.6.2.2 Erweiterungen

Ein Erfolgsfaktor von Ethernet ist dessen ständige Weiterentwicklung. Erweiterungen ermöglichen den Einsatz in neuen Anwendungsbereichen mit spezifischen Anforderungen. In Abbildung 2.5 sind neben den bereits erwähnten IEEE 802-Standards die Ethernet-Erweiterungen, die für eingebettete Kommunikationsnetze von Bedeutung sind, in das OSI-Referenzmodell eingeordnet. Die Erweiterungen werden im Folgenden näher beschrieben. Die Abbildung 2.5 zeigt zudem eine medienunabhängige und eine medienabhängige Schnittstelle sowie die Reconciliation- (*dt. Abgleich*) Schicht. Diese Schicht bildet zusammen mit der medienunabhängigen Schnittstelle bei Ethernet-Varianten mit Übertragungsraten von größer 100 Mbit/s den Zugang zur Bitübertragungsschicht. Bei Ethernet ist die Bitübertragungsschicht unterteilt in drei weitere Schichten, welche in [4] ausführlich erklärt werden.

MAC Bridge Um eine Kollision auf einem Bus zu erkennen, darf eine maximale Kabellänge in Abhängigkeit von der Übertragungsrate nicht überschritten werden. Bei steigender Anzahl von Stationen, die versuchen zu senden, steigt die Busauslastung und die Leistungsfähigkeit sinkt. Irgendwann ist die Kapazität erschöpft [29]. Zur Verbesserung der Leistungsfähigkeit und zur Vergrößerung der möglichen räumlichen Ausdehnung wurden sogenannte *MAC Bridges* (Netzkoppelemente auf OSI-Schicht 2) [92] eingeführt. Eine solche Bridge verbindet zwei Teilnetze miteinander. Dabei können sich die beiden Teilnetze in ihren Übertragungsraten und/oder Medientypen unterscheiden. Eine Bridge empfängt einen Rahmen aus einem Teilnetz und leitet diesen selektiv in das andere weiter. Eine MAC Bridge mit mehr als zwei Anschlüssen wird Multiport-Bridge oder Switch (*dt. Vermittlungsknoten*) genannt. Heutige, vollständig segmentierte Ethernet-Netze bestehen aus einem oder mehreren Vermittlungsknoten und Vollduplexverbindungen. Ein solches Ethernet-Netz unterscheidet sich deutlich vom ursprünglichen Ethernet. Es wandelte sich von einem kollisionsbehafteten zu einem kollisionsfreien Netz mit einem (nahezu) deterministischen Verhalten und damit zu einem Netz mit höherer Leistungs- und Echtzeitfähigkeit. Bei einem vollständig segmentierten Netz mit Vollduplexleitungen steht im Vergleich zu einem gemeinsamen Medium eine doppelt so hohe Übertragungsrate zur Verfügung. Es können zwei Endknoten theoretisch gleichzeitig mit der jeweils maximal verfügbaren Übertragungsrate in beide Richtungen kommunizieren.

Die Entscheidung, ob ein Rahmen weitergeleitet oder verworfen wird, erfolgt anhand der Zieladresse mit Hilfe einer Look-up Tabelle [92]. In dieser Tabelle werden MAC-Adressen und der dazugehörige Anschluss (dazugehöriges Netzsegment) gespeichert. Wenn der Vermittlungsknoten einen Rahmen empfängt, trägt er die Quelladresse und den dazugehörigen Anschluss in die Tabelle ein. So lernt der Vermittlungsknoten, welche Adressen zu welchem Netzsegment gehören. Falls ein Rahmen empfangen wird, dessen Adresse nicht in der Tabelle enthalten ist, wird dieser Rahmen an alle Segmente weitergeleitet, mit Ausnahme des Eingangssegmentes. Dieser Vorgang wird Fluten (*engl. flooding*) genannt. Die Netzstruktur kann sich allerdings ändern. Aus diesem Grund wird die Tabelle bei jedem empfangenen Rahmen aktualisiert und in periodischen Abständen werden veraltete Einträge aus der Tabelle entfernt [29].

Virtualisierung und Verkehrsklassifizierung Zur Optimierung der Verkehrslenkung und für mehr Flexibilität beim Netzentwurf wurde 1998 eine Rahmenerweiterung standardisiert [93]. Diese Erweiterung ist in Abbildung 2.4 (unten) dargestellt und ermöglicht die logische Separierung (Virtualisierung) eines physikalischen Netzes. Die Rahmenerweiterung besteht aus vier

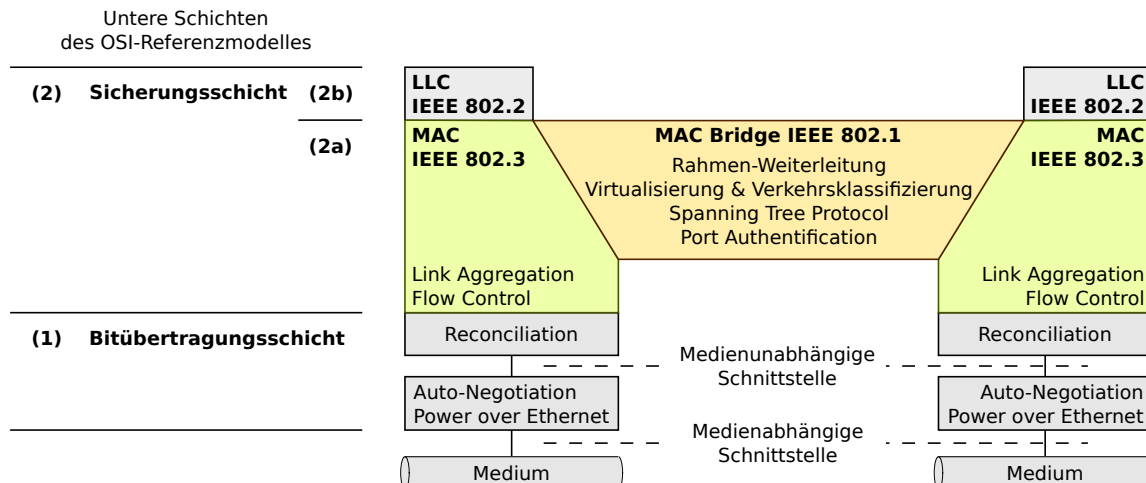


Abbildung 2.5: Einordnung der IEEE 802-Standards und der Ethernet-Erweiterungen in das OSI-Referenzmodell

Bytes. Die ersten beiden Bytes TPID (*engl. Tag Protocol Identifier*) entsprechen der Typinformation eines normalen Rahmens und besitzen den Wert 0x8100 (*VLAN-Tagged Frame*) [91]. In den nächsten beiden Bytes TCI (*engl. Tag Control Information*) sind drei Bit zur Kennzeichnung der Priorität, ein Bit für den Indikator des kanonischen Formats (Wertigkeitsreihenfolge) und 12 Bit für die Kennnummer des virtuellen Netzes (VLAN-ID) reserviert. An diese Kennnummer schließt sich die Typeninformation T an, die ursprünglich an der Stelle des TPID stand. Durch die Möglichkeit der Priorisierung können Datenströme unterschiedlich behandelt und Dienstgüteeanforderungen unterstützt werden.

Spanning Tree Protocol Heutige Kommunikationsnetze haben häufig eine teilvermaschte Struktur, in denen Schleifen auftreten können. In diesem Fall zirkulieren Nachrichten endlos und können gegebenenfalls das Netz überlasten. Aus diesem Grund müssen Schleifen entsprechend behandelt bzw. aufgelöst werden. Die bekannteste Lösung ist das Spanning Tree Protocol (STP) [92]. Dieses Protokoll entfernt logisch einige Verbindungen, indem Anschlüsse in den Vermittlungsknoten deaktiviert werden. Bei Ausfall einer Verbindung können zuvor deaktivierte Anschlüsse wieder aktiviert und eine zuvor entfernte Verbindung wieder genutzt werden. Zur Auflösung von Schleifen in virtualisierten Netzen wurde das Multiple Spanning Tree Protocol (MSTP) [93] spezifiziert.

Port Authentication Bei Ethernet stehen Methoden zur Authentifizierung und Autorisierung am Netzzugang zur Verfügung. Das im IEEE 802.1X Standard [94] definierte Protokoll ermöglicht eine Authentifizierung der an einem Vermittlungsknoten angeschlossenen Endknoten. Nach erfolgreicher Authentifikation wird der entsprechende Anschluss freigegeben. Die Verfahren zur Authentifikation werden im Standard nicht beschrieben. Für diese Aufgabe wird das Extensible Authentication Protocol (EAP) [95] eingesetzt. Mehr Informationen zu Port Authentication sind zu finden in [96, 97].

Link Aggregation Eine weitere wesentliche Erweiterung ist die Link Aggregation Funktion (IEEE 802.3 [79], Abschnitt 43). Diese Funktion ermöglicht das Bündeln bzw. Zusammenschalten mehrerer paralleler physikalischer Leitungen zu einer logischen Leitung. Durch das Zusammenschalten kann die maximale Übertragungsrate zwischen zwei Systemen erhöht werden. Ein

weiterer Vorteil ist eine erhöhte Ausfallsicherheit. Bei Ausfall einer physikalischen Leitung verringert sich lediglich die maximale Datenübertragungsrate. Zur automatischen Erkennung paralleler Leitungen wurde das Link Aggregation Control Protocol (LACP) [98] spezifiziert.

Flow Control Aufgrund unterschiedlicher Übertragungsraten auf Teilstrecken in einem Netz kann es in den Vermittlungsknoten zu Pufferüberläufen und zu Rahmenverlusten kommen. Die Ethernet Flusskontrolle (*engl. flow control*) [79] ermöglicht, die Übertragung temporär zu unterbrechen, indem bei Überlast ein Vermittlungsknoten oder Endknoten eine Überlast signalisiert. Hierzu schickt der überlastete Knoten eine entsprechende Nachricht mit der gewünschten Unterbrechungsdauer an seinen Nachbarn [4]. Dadurch können Rahmenverluste vermieden werden.

Power over Ethernet (PoE) Ethernet unterstützt eine Energieversorgung von Komponenten über elektrische Datenleitungen. Genannt wird diese Funktionalität Power over Ethernet und ist in den Standards IEEE 802.3at [99] und 802.3af [100] spezifiziert. Momentan können maximal 30 W Leistung in eine Leitung eingespeist werden. In der Diskussion ist eine Erhöhung der Maximalleistung auf 60 W [101]. Auch ein Mischbetrieb mit PoE-fähigen Systemen und herkömmlichen Systemen ist möglich.

Auto-Negotiation Ethernet ermöglicht unterschiedliche Übertragungsraten. Dadurch können Systeme mit unterschiedlichen Ethernet-Schnittstellen miteinander verbunden werden. Auto-Negotiation handelt automatisch die maximal mögliche Übertragungsrate zwischen zwei Endpunkten aus und konfiguriert diese. Das Verfahren arbeitet auf OSI-Schicht 1 und ist im IEEE 802.3u Standard [79] definiert. Allerdings kann das Verfahren nur bei verdrehten Adernpaaren (*engl. twisted-pair*) angewandt werden [4]. Diese Erweiterung ist ein gutes Beispiel für die Rückwärtskompatibilität und die Einfachheit des Managements.

Medientypen Ein Erfolgsfaktor von Ethernet ist, dass eine Vielzahl an Medientypen eingesetzt werden kann. Die ursprüngliche Version von Ethernet verwendete Koaxialkabel. Es folgten Leitungen mit verdrehten Adernpaaren unterschiedlicher Ausprägung [102]. Bei solchen Leitungen werden die Sende- und Empfangssignale auf getrennten Leitungspaaren übertragen. Dies ermöglicht einen Vollduplexbetrieb. Zunehmend werden auch Lichtwellenleiter eingesetzt. Diese haben den Vorteil, dass sie unempfindlich gegenüber elektrischen oder elektromagnetischen Störfeldern sind. Rech diskutiert in seinem Buch [102] die unterschiedlichen Medientypen sowie deren Vor- und Nachteile.

Die beschriebenen Erweiterungen sowie weitere werden vom Autor et al. in [4] und von Rech [102] detaillierter beschrieben.

2.6.2.3 Industrial Ethernet

Nachdem Ethernet im Bereich der lokalen Netze weltweit einen Marktanteil von mehr als 90% erreicht hatte, entstand Ende der neunziger Jahre die Idee, Ethernet auch in der Fertigungs-, Prozess- und Automatisierungstechnik einzusetzen [41]. Das Ziel war, eine flexible, einheitliche, leistungsfähige und kostengünstige Technologie zu etablieren und so die Anzahl unterschiedlicher (Feld-) Bussysteme einzuschränken. Im Verlauf der letzten Jahre wurden zunehmend Lösungen basierend auf Ethernet entwickelt. Diese werden unter dem Namen *Industrial*

Ethernet vermarktet. Allerdings gibt es bis heute keinen einheitlichen, wohldefinierten Standard für Industrial Ethernet. So sind in der internationalen Norm IEC 61784-2 [103] 23 Typen von Industrial Ethernet registriert [104]. Beschreibungen unterschiedlicher Industrial Ethernet-Lösungen sind zu finden in [25, 31, 40, 41, 70, 105, 106].

Die am Markt existierenden Lösungen können danach klassifiziert werden, inwieweit sie kompatibel zu den IEEE 802-Standards sind. Heutige Industrial Ethernet-Lösungen definieren häufig weit mehr als bloße Kommunikationsfunktionalität auf den beiden unteren OSI-Schichten. In [4] wird zwischen *vollständig kompatiblen*, *kompatiblen* und *inkompatiblen* Lösungen unterschieden.

Vollständig kompatible Lösungen Lösungen, die vollständig kompatibel sind, nutzen Ethernet als Vernetzungstechnologie und definieren darüber hinaus eine Architektur oberhalb von OSI-Schicht 2. Somit können kostengünstige Standardkomponenten eingesetzt werden. Solche Lösungen können problemlos mit einem *reinen* Ethernet-Netz verbunden werden und es ist keine Protokollumsetzung durch ein Gateway erforderlich. Ein bekanntes Beispiel ist das Ethernet Industrial Protocol (Ethernet/IP) [107]. Zur Sicherstellung der Echtzeitfähigkeit verwenden solche Lösungen erweiterte Ethernet-Rahmen. Zeitkritische Informationen erhalten dabei eine hohe Priorität, zeitunkritische Informationen eine niedrige. Manche Lösungen nutzen spezielle Protokollstapel für die Kommunikation oberhalb von Schicht 2 und andere wiederum verwenden die Protokolle TCP/UDP/IP [106].

Bis heute wird heftig diskutiert, inwieweit Ethernet überhaupt deterministisch und echtzeitfähig ist. In vielen Anwendungsbereichen sind die Zyklus- und Reaktionszeiten im Millisekundenbereich und daher völlig ausreichend [25]. In einem vollständig segmentierten Ethernet-Netz treten keine Kollisionen auf. Aber Ethernet wird durch die Segmentierung und die Verwendung von Verkehrsklassen noch nicht echtzeitfähig. Beispielsweise können durch eine zu geringe Pufferdimensionierung in einem Vermittlungsknoten bei gleichzeitig ankommenden Rahmen Verluste auftreten [105]. Um dies zu vermeiden, wurden angepasste, erweiterte Lösungen entwickelt. Die Implementierung eines zusätzlichen Schedulers zur Verkehrsverwaltung zwischen OSI-Schicht 2 und 3 stellt eine mögliche Lösung dar [108]. In der Vergangenheit wurden mehrere Arbeiten [40, 109–122], die die Echtzeitfähigkeit von Ethernet bewertet und nachgewiesen haben, veröffentlicht.

Kompatible Lösungen Kompatible Lösungen unterstützen vollständig die Funktionen und Eigenschaften, die in den IEEE 802-Standards beschrieben sind. Netze dieser Klasse können somit mit *reinen* Ethernet-Netzen gekoppelt werden. Jedoch kann die Echtzeitfähigkeit nur mit speziell angepassten Vermittlungsknoten garantiert werden. Somit endet die Echtzeitfähigkeit an den Grenzen eines solchen (Teil-) Netzes [105]. Ein bekanntes Beispiel ist *EtheReal* [123]. Zur Bereitstellung von Ratengarantien mit maximaler Verzögerung wird ein verbindungsorientiertes Dienstgüteverfahren eingesetzt. Falls ein Endknoten zeitkritische Informationen übertragen möchte, versucht er, mittels einer Signalisiernachricht eine Verbindung zum Ziel aufzubauen. Falls genügend Ressourcen auf dem Pfad verfügbar sind, leitet jeder Vermittlungsknoten von der Quelle zum Ziel die Anfrage an den nächsten Vermittlungsknoten weiter. Der letzte Vermittlungsknoten schickt eine Reservierungsbestätigung entlang des protokollierten Pfades zurück. Sämtliche Vermittlungsknoten auf dem Pfad reservieren die angefragten Ressourcen (Übertragungsraten). Voraussetzung für eine erfolgreiche Reservierung ist, dass jeder

Knoten das Reservierungsprotokoll unterstützt. Dieses Verfahren ähnelt dem bekannten Signalisierungsprotokoll Resource reSerVation Protocol (RSVP) [124].

Inkompatible Lösungen Inkompatible Lösungen werden überwiegend in Bereichen mit hohen Anforderungen an die Zykluszeit, die Verzögerung, die Synchronität und den Jitter eingesetzt. Diese Anforderungen können ohne eine Protokollmodifikation nicht realisiert werden [25]. Inkompatible Lösungen basieren in der Regel auf einem deterministischen Medienzugriffsverfahren. Decotignie [105] gibt einen Überblick über geeignete Zugriffsverfahren. Kommunikationsnetze, in denen ein modifiziertes Zugriffsverfahren verwendet wird, werden quasi isoliert betrieben und über spezielle Koppellemente mit normalen Ethernet-Netzen verbunden. Ein bekanntes Beispiel für eine solche Lösung ist EtherCAT [125]. Hier wird von einem Master ein Ethernet-Standardrahmen erzeugt und an den ersten Teilnehmer geschickt. Dieser liest und beschreibt die für ihn reservierten Datenfelder und leitet den Rahmen an seinen Nachfolger weiter. Der letzte Teilnehmer schickt den Rahmen zurück an den Master. Da eine normale, vollduplexfähige Verkabelung zwischen den Teilnehmern verwendet wird, ergibt sich physikalisch eine Linientopologie, logisch jedoch ein offener Ring. Mit EtherCAT können Zykluszeiten von $30 \mu s$ erreicht werden [70].

2.6.3 Vergleich (Feld-) Bussysteme und Ethernet

Ethernet umfasst eine Menge offener Standards für die beiden unteren OSI-Schichten (siehe Abbildung 2.5) und definiert im Wesentlichen die Vernetzung von Systemen und die Übertragung von Informationen. Heutige (Feld-) Bussysteme verfügen über mehr als reine Kommunikationsfunktionalität. Sie spezifizieren Kommunikations- und Geräteprofile und bieten Funktionen zum Management, zur Konfiguration, zur Inbetriebnahme und zur Diagnose [25, 30]. Ein Vergleich, wie er auch von Schnell und Wiedemann in ihrem Buch [25] gemacht wird, ist somit nur begrenzt möglich. Allerdings ergänzen zunehmend die bekannten Protokolle Internet Protocol (IP) [126], Transmission Control Protocol (TCP) [127] und User Datagram Protocol (UDP) [128] Ethernet um Funktionalitäten höherer Schichten. IP wird zur Wegfindung über heterogene Netze hinweg eingesetzt. Zum Transport großer Datenmengen eignet sich TCP und UDP. Da Feldbusprotokolle häufig für die Übertragung von Informationen mit geringem Umfang konzipiert wurden, sind bei größeren Datenmengen TCP und UDP eindeutig im Vorteil [25].

Oberhalb von TCP und UDP gibt es eine Vielzahl etablierter Anwendungsprotokolle, die zwar nicht für den Einsatz im industriellen Umfeld entwickelt wurden, dort aber zunehmend eingesetzt werden. Beispiele hierfür sind das Hypertext Transfer Protocol (HTTP) [129], das File Transfer Protocol (FTP) [130, 131] oder das Simple Network Management Protocol (SNMP) [132]. Für die Zeitsynchronisation kann das bereits erwähnte SNTP verwendet werden. In vielen Bereichen ist dessen Genauigkeit völlig ausreichend.

Ethernet erlaubt zudem ein einfaches Management und eine automatische Systemintegration. Das Address Resolution Protocol (ARP) [133] ermöglicht die Zuordnung einer IP-Adresse zu einer MAC-Adresse. Es existiert also eine Verbindung zwischen der Vermittlungsschicht und der Sicherungsschicht. Damit entfällt die bei (Feld-) Bussystemen notwendige statische Adresskonfiguration, was für größere Netze unabdingbar ist [25]. Die Netzeinstellungen für einen Endknoten können mit dem Dynamic Host Configuration Protocol (DHCP) [134] automatisch kon-

figuriert werden. Die Konfiguration der Vermittlungsknoten nach der Systemintegration entfällt ebenfalls, da diese selbstlernend sind.

Betrachtet man die Kombination aus Ethernet, IP/TCP/UDP und den genannten Anwendungsprotokollen, dann ist ein Vergleich mit den (Feld-) Bussystemen durchaus möglich. Da die Protokolle IP, TCP und UDP quasi zu allen Internet-Anwendungen kompatibel sind [135], entstehen durch deren Einsatz neue Möglichkeiten für die Kommunikation zwischen eingebetteten Kommunikationsnetzen und Netzen der Außenwelt. Inzwischen wurden in vielen Bereichen dank integrierter Webserver das Management und die Konfiguration von Systemen vereinfacht. So kann mit Hilfe eines Internet-Browsers jederzeit (*engl. anytime*) und von überall (*engl. anywhere*) auf ein System zugegriffen werden.

In der Massenproduktion wie der Fahrzeugproduktion spielen die Kosten eine außerordentliche Rolle. Ethernet ist eine weit verbreitete (Massenmarkt-) Technologie und wird kontinuierlich weiterentwickelt. Die derzeitigen Aktivitäten zur Standardisierung von 40 Gbit/s und 100 Gbit/s demonstrieren das weltweit große Interesse hinsichtlich Weiterentwicklung. Auch das Wissen und Verständnis für diese Technologie ist weltweit verfügbar und nicht auf spezielle Bereiche wie beispielsweise die Fertigungs-, Prozess- und Automatisierungstechnik beschränkt [40]. Ethernet wird nicht durch einen Marktführer dominiert, wie es häufig bei (Feld-) Bussystemen der Fall ist. Die weite Verbreitung führt zu einer nachhaltigen und hohen Verfügbarkeit von Komponenten. Folglich ist Ethernet als Vernetzungstechnologie auch für Netze, die eine lange Lebensdauer besitzen, geeignet. Ein Netzausbau ist bei Ethernet einfach, da es traditionell zu allen Vorgängerstandards kompatibel ist.

Ein weiterer Vorteil ist, dass heutige handelsübliche Rechner bereits über eine Ethernet-Schnittstelle verfügen. So kann die Instandhaltung bequem mit herkömmlichen Rechnern erfolgen, indem sich diese einfach, unter Berücksichtigung von Sicherheitsaspekten, mit dem eingebetteten Kommunikationsnetz verbinden. Im Vergleich dazu ist für die Instandhaltung von (Feld-) Bussystemen spezielle Hardware notwendig, die hohe Kosten verursacht.

Ein häufig genanntes Argument gegen den Einsatz von Ethernet ist die fehlende Echtzeitfähigkeit und das nicht deterministische Verhalten. Dieses Argument gilt für vollständig segmentierte Netze anerkannterweise nicht mehr. In den meisten Fällen erfüllen kollisionsfreie Ethernet-Netze Echtzeitanforderungen und Determinismus. Für streng isochrone Anwendungen wie im Bereich der Antriebssteuerung müssen jedoch die in Abschnitt 2.6.2.3 beschriebenen Erweiterungen eingesetzt werden.

Das Versenden von Nachrichten mit kleinen Nutzdatenlängen ist bei Ethernet im Vergleich zu (Feld-) Bussystemen ineffizient, dies wird aber durch die um den Faktor 100 höhere Übertragungsrate kompensiert. Ethernet ist mit Übertragungsraten von 1 Gbit/s (*Gigabit Ethernet*) für eingebettete Kommunikationsnetze derzeit ausreichend. Ethernet unterstützt mit 10 Gbit/s und demnächst mit 40 und 100 Gbit/s sogar noch höhere Übertragungsraten. Inwiefern diese für eingebettete Kommunikationsnetze technisch und wirtschaftlich interessant sind, ist fraglich. Hinsichtlich der Übertragungsraten wird Ethernet garantiert auch zukünftige Anforderungen erfüllen. Ergänzend kommt hinzu, dass sich dank der hohen Übertragungsraten Kommunikationszyklen verkürzen.

Ein Wechsel der Vernetzungstechnologie von (Feld-) Bussystemen hin zu Ethernet ermöglicht eine vereinfachte Maschinen- und Geräteinstallation. So kann mit PoE die Verkabelung durch Wegfall der separaten Energieversorgung reduziert werden. Insbesondere in eingebetteten Bereichen kann dies zu erheblichen Kosten- und Gewichtseinsparungen führen. Nach Breuer und Kartes [101] lassen sich durch den Einsatz von PoE bis zu 80% der Installationskosten einsparen bei gleichzeitiger Steigerung der Verfügbarkeit. In Zukunft werden intelligente Sensoren und Aktoren lediglich über eine Leitung angebunden sein, über die sowohl kommuniziert als auch die benötigte Energie übertragen wird.

Im Vergleich zu (Feld-) Bussystemen bietet Ethernet mehr Möglichkeiten hinsichtlich der Topologie. (Feld-) Bussysteme haben diesbezüglich oft Einschränkungen. Mit Ethernet können alle in Abschnitt 2.5 beschriebenen Topologien realisiert werden, wobei ein Bus aufgrund von Kollisionen nicht interessant ist. Durch die Integration eines *kleinen* Vermittlungsknotens mit drei Anschlüssen in jedem System kann eine Linienstruktur realisiert werden. Sogar ein unidirektionaler Ring ist denkbar. Eine solche Struktur vergleichen Rahmani et al. [88] mit einer kaskadierten Sternstruktur und bewerten deren Leistungsfähigkeit für den Einsatz im Fahrzeug. Rüping et al. [136] analysieren Ethernet-basierte Topologien für den Automatisierungsbereich hinsichtlich der Übertragungsverzögerung.

Zusammengefasst bietet Ethernet einige Vorteile gegenüber heutigen (Feld-) Bussystemen. Durch die Kombination aus Ethernet, IP/TCP/UDP und bekannten Anwendungsprotokollen werden auch Funktionalitäten höherer Schichten abgedeckt. Die vollständige Segmentierung eines Ethernet-Netzes ermöglicht zudem Echtzeitfähigkeit und Determinismus.

2.7 Entwicklungen und Trends

Mit dem Mil-Std-1553 System wurde in den siebziger Jahren eines der ersten Bussysteme entwickelt. Es folgte später die Entwicklung von CAN für die Vernetzung von Steuergeräten im Fahrzeug. Ebenfalls in diesem Zeitraum wurden die ersten offenen, proprietären Systeme für die Fertigungs-, Prozess- und Automatisierungstechnik entwickelt. Unter *offen* ist die Offenlegung der Protokollspezifikationen zu verstehen. Dies ermöglichte unterschiedlichen Herstellern, Systeme für einen bestimmten Bus zu entwickeln. In den neunziger Jahren wurden dann verschiedenste internationale Standards veröffentlicht und somit der Schritt von den proprietären Bussystemen zu herstellerneutralen Bussystemen vollzogen [30]. Sauter [137] gibt eine Übersicht über die Evolution der (Feld-) Bussysteme.

Oft bestehen eingebettete Kommunikationsnetze aus nur einem logischen Segment ohne eine Möglichkeit des Netzüberganges und bilden damit eine Kommunikationsinsel. Zukünftig werden diese Netze in komplexe (unternehmensweite) Netze stärker integriert sein und werden zunehmend mit externen, IP-basierten Netzen der Außenwelt kommunizieren. So kann beispielsweise ein Produktionsplaner innerhalb einer Firma die Maschinenauslastung beobachten oder ein Automobilhersteller kann über ein Mobilfunknetz Diagnoseinformationen von einem Fahrzeug abfragen und Serviceinformationen übermitteln. Natürlich müssen bei der Öffnung der Netze Sicherheitsaspekte beachtet und es dürfen keine Echtzeitanwendungen beeinflusst werden. Für eine einfache Kopplung mit Netzen der Außenwelt sind zudem standardisierte

Schnittstellen und eine einheitliche Kommunikationsbasis notwendig, wie sie beispielsweise Ethernet liefert.

In den letzten Jahren sind die Anforderungen an eingebettete Kommunikationsnetze erheblich gestiegen. Anfangs wurden nur kurze Nachrichten mit geringen Datenmengen übertragen. Unter anderem aufgrund der Kopplung mit externen, IP-basierten Netzen werden neben den Steuerungs- und Regelungsinformationen auch Management-, Prozess-, und Konfigurationsinformationen, die einen größeren Umfang haben können, übertragen. Hinzu kommen verteilte Multimediaanwendungen mit einer Unvorhersehbarkeit der Verkehrscharakteristik. Aus diesen Gründen werden zunehmend höhere Datenübertragungsraten unter Beibehaltung der Echtzeitfähigkeit gefordert [137]. Grundsätzlich bestehen zwei Optionen, diese Anforderungen zu erfüllen [40]:

1. Die maximale Übertragungsrate von (Feld-) Bussystemen wird erhöht und diese werden in die Lage versetzt, Nachrichtenverkehr auf der Basis von IP zu übertragen, ohne die Echtzeitfähigkeit negativ zu beeinflussen. Gemessen an den auf dem Markt verfügbaren (Feld-) Bussystemen müssten deren Übertragungsraten um mindestens eine Zehnerpotenz wachsen.
2. Ethernet wird dahin gehend erweitert bzw. konfiguriert, dass neben den hohen Übertragungsraten auch zeitkritische Informationen in geforderter Dienstgüte übertragen werden können.

Aus Wirtschaftlichkeitsgründen wird sich aller Voraussicht nach die zweite Option etablieren. Ethernet bietet die Möglichkeit einer durchgängigen, nachhaltigen und einheitlichen Vernetzung auf der Basis günstiger, in Serie gefertigter Standardkomponenten mit großer Stückzahl (*engl. commercial off-the-shelf*). Dies wird zu einer Kostensenkung sowohl in der Anschaffung von Komponenten als auch im operativen Netzbetrieb führen.

Bisher werden als Übertragungsmedien überwiegend Koaxialkabel und verdrehte Zweidrahtleitungen eingesetzt. Zunehmend werden alternative Übertragungsmedien eingesetzt. Der Einsatz optischer Medien wird aufgrund sinkender Preise und deren Störunempfindlichkeit zunehmen. Obwohl die schlechten Eigenschaften einer drahtlosen Kommunikation in rauen Umgebungen nur geringe Datenraten zulassen, werden sich auch drahtlose Lösungen weiter verbreiten. Die entfallende Verkabelung und die Mobilität von Systemen ermöglichen flexible Kommunikationsnetze. Aufgrund der Vielzahl am Markt verfügbaren Wireless LAN (WLAN) [138]-Ethernet-Bridges ist eine einfache und kostengünstige Kopplung zwischen drahtlosen Netzen und Ethernet-Netzen möglich.

Durch die weiter sinkenden Preise von Sensoren und Aktoren bei gleichzeitiger Zunahme ihrer Leistungsfähigkeit wird in vielen Anwendungsbereichen die Komplexität und Größe der Netze weiter wachsen. Waren 1985 noch ungefähr 3 bis 6 Systeme über ein (Feld-) Bussystem vernetzt, werden 2015 in großen Gebäuden mehr als 1 Million Sensoren, Aktoren und Systeme miteinander vernetzt sein [137]. Sicherlich wird dieser Zuwachs nicht in allen Anwendungsbereichen stattfinden. So kann in einem Fahrzeug die Anzahl der Steuergeräte aus Platzgründen nicht weiter zunehmen [139]. In diesem Bereich wird die Leistungsfähigkeit der einzelnen Steuergeräte ansteigen und die Virtualisierung Einzug halten. Jedoch werden durch

innovative mechatronische *x-by-wire* Lösungen der Vernetzungsgrad und die Anforderungen an die Zuverlässigkeit zunehmen.

Das Integrieren und Konfigurieren neu hinzukommender Systeme stellt noch immer eine Herausforderung dar. Es werden Dienste zur selbstständigen Systemintegration ohne menschliches Zutun benötigt [137]. Diese Dienste werden ähnlich zu dem bereits heute bekannten *Plug-n-Play* Konzept sein. Seit langem können die Netzeinstellungen für einen Rechner via DHCP automatisch konfiguriert werden. Allerdings werden neue Dienste weit über den Funktionsumfang von DHCP hinausgehen. Das Ziel ist, während der Integration verteilte Ressourcen und Dienste innerhalb des Netzes automatisch zu finden. Umgekehrt müssen hinzukommende Systeme ihre Dienste und Ressourcen im Netz bekannt machen. Dazu werden Formalismen und Protokolle notwendig sein, mit denen ein System Dienste und Ressourcen auffinden, anfordern und veröffentlichen kann.

Heutzutage kommen in einem technischen System mehrere Vernetzungstechnologien gleichzeitig zum Einsatz. Die Gründe sind divergierende Anforderungen und die stufenweise Einführung der Technologien. Jede dieser Technologien benötigt spezielle Software und Hardware, die nicht direkt zueinander kompatibel sind. Zur Verringerung der Komplexität und der Kosten wird die Anzahl gleichzeitig verwendeter Vernetzungstechnologien stagnieren oder sogar reduziert werden müssen. Hinzu kommt, dass sich in eingebetteten Kommunikationsnetzen ein Paradigmenwechsel vollzogen hat. Statt einer eigenen Entwicklung dedizierter Technologien setzen die Hersteller und Ausrüster zunehmend auf Standardkomponenten aus dem Konsumentenbereich. Diese sind aufgrund der höheren Stückzahl kostengünstiger und teilweise deutlich leistungsfähiger. Die Verwendung standardisierter Technologien fördert unter anderem die Vereinheitlichung technischer Systeme. Der Einsatz von Standardkomponenten bietet weitere Vorteile. Eine parallele Entwicklung proprietärer Technologien mit teilweisen langwierigen Harmonisierungsprozessen wird dadurch vermieden [139].

Bisher hat sich in Anwendungsbereichen wie dem Automobil- und Flugzeugbau bei dem Übergang von einer Produktgeneration zur nächsten die Funktionalität um etwa 20% geändert. Jedoch hat sich die Software um bis zu 80% geändert, da sich die Technologien, z. B. für die Steuergeräte und Bussysteme, geändert haben [23]. Eine zunehmende Herausforderung bildet die Unabhängigkeit und Asynchronität der Technologieentwicklung zur Produktentwicklung [139]. Deshalb muss die Auswahl einer Vernetzungstechnologie nachhaltig erfolgen. Eine Technologie muss über viele Jahre bestehen und für zu erwartende technologische Veränderungen möglichst robust sein. Die ausgewählte Technologie muss erweiterbar sein, um zu erwartenden Innovationen Rechnung zu tragen. Eine Technologie, die bereits heute an den Grenzen ihrer Leistungsfähigkeit ist, erfüllt nicht die Anforderung der Technologierobustheit [139]. In Zukunft werden verstärkt Technologien eingesetzt, die als *evolutionsfähig* einzuschätzen sind und die kompatibel zu nachfolgenden Standards sein werden. Dabei werden sicherlich der Verbreitungsgrad und die Historie einer Technologie zur Abschätzung herangezogen. Ethernet löst weitestgehend das Problem der Asynchronität der Technologieentwicklung zum Produktlebenszyklus, da es bereits seit Jahrzehnten erfolgreich eingesetzt wird, somit die Technologierobustheit bewiesen hat und sämtliche Standards quasi zueinander kompatibel sind bzw. kompatibel sein werden.

Das für den Bereich der lokalen Netze entwickelte, paketorientierte Ethernet wurde inzwischen dahin gehend erweitert, dass dieses auch in eingebetteten Bereichen mit Echtzeitanforderun-

gen eingesetzt werden kann. Wie bereits erwähnt haben sich im industriellen Bereich bereits mehrere Lösungen auf der Basis von Ethernet etabliert. In Flugzeugen wird seit Längerem eine Ethernet-basierte Lösung namens Avionics Full Duplex Switched Ethernet (AFDX) [140] eingesetzt. Im neuen Airbus A380 wird das Flattern der Tragflächen von einem Sensorsystem, das Teil des AFDX-Netzes ist, überwacht [34]. Derartige Lösungen werden die bisherigen (Feld-) Bussysteme ablösen [141]. Nachdem Ethernet-basierte Lösungen bereits erfolgreich eingesetzt werden, wird nun verstärkt auf *Standard Ethernet* gesetzt. Der Begriff *Standard Ethernet* ist dahin gehend zu verstehen, dass keine angepasste Software oder Hardware nötig und es vollständig kompatibel zu dem bereits heute in lokalen Netzen eingesetzten Ethernet sein wird.

Durch die weite Verbreitung werden fehlende Funktionalitäten ergänzt und Ethernet wird weitere Anwendungsbereiche erschließen. Das Interesse der Automobilhersteller, Ethernet für die fahrzeuginterne Vernetzung einzusetzen, wurde bereits erwähnt. Das Forschungsprojekt *ESANA* [142] beschäftigte sich mit dem Einsatz von Echtzeit-Ethernet für die Vernetzung von Aktoren und Sensoren.

Die Historie, die zukünftigen Anforderungen und die aktuelle Entwicklung lassen den Schluss zu, dass die nächste Generation der eingebetteten Kommunikationsnetze nach der Einführung der (Feld-) Bussysteme auf Ethernet basieren wird. Ethernet ist die technologische Voraussetzung (*engl. enabling technology*) für innovative und verteilte Anwendungen und Dienste. Auch in Zukunft werden Ethernet-Netze vollständig segmentiert sein, d. h. ausschließlicher Einsatz von Vermittlungsknoten und Vollduplexleitungen. Das erhöht allerdings die Kosten und den Platzbedarf. Damit trotzdem die Vorteile von Ethernet auch in kostensensitiven (Massenmarkt-) Produkten genutzt werden können, muss die zugrunde liegende Topologie hinsichtlich der Kosten optimal sein. Um dies zu erreichen, werden in dieser Arbeit neue Verfahren zur Kostenoptimierung von Topologien eingebetteter, Ethernet-basierter Kommunikationsnetze entwickelt. Für eine kostenoptimale Topologie müssen eine optimale Anzahl von Vermittlungsknoten bestimmt, diese optimal positioniert und sämtliche Komponenten optimal miteinander vernetzt werden. Dabei müssen bei der Optimierung zwingend sowohl die Eigenschaften wie Leitungsbündelung und positionsabhängige Kosten als auch die Anforderungen hinsichtlich Ressourcendimensionierung und Ausfallsicherheit berücksichtigt werden.

3 Grundlagen der Topologieoptimierung

Dieses Kapitel beschäftigt sich mit den für diese Arbeit notwendigen Grundlagen der Topologieoptimierung. Häufig werden Probleme der Topologieoptimierung auf kombinatorische Optimierungsprobleme, die eng mit graphentheoretischen Fragestellungen verknüpft sind, abgebildet. Aus diesem Grund gliedert sich dieses Kapitel in zwei Teile. Im ersten Teil werden die Grundlagen der Graphentheorie, im zweiten Teil die Grundlagen der kombinatorischen Optimierung behandelt.

Die beiden Teile sind nicht als vollständige Einführung in die Themengebiete der Graphentheorie und der kombinatorischen Optimierung zu verstehen, sondern als Einführung und Erklärung aller im Laufe dieser Arbeit notwendigen Begriffe und Algorithmen.

3.1 Graphentheorie

Die Topologie eines Kommunikationsnetzes kann in Form eines Graphen beschrieben werden. Dies hat den Vorteil, dass viele Probleme der Topologieoptimierung mit existierenden Verfahren der Graphentheorie gelöst werden können. Aus diesem Grund befasst sich dieses Kapitel mit der Graphentheorie. Im ersten Teil werden die in der Graphentheorie grundlegenden Begriffe sowie die üblichen Datenstrukturen eingeführt und definiert.

Graphen werden in unterschiedlichen Bereichen verwendet. Beispiele sind neben der Beschreibung von Kommunikationsnetzen auch die Untersuchung von elektrischen Schaltungen, Aufgaben des Projektmanagements und der Routenplanung. Daher existiert eine Vielzahl von Algorithmen zur Erzeugung sowie zur Durch- und Untersuchung von Graphen. Im zweiten Teil dieses Abschnittes werden ausgewählte, für diese Arbeit relevante Graphenalgorithmen vorgestellt.

Für eine ausführliche Diskussion der Graphentheorie sei auf die umfassende Literatur [143–148] verwiesen.

3.1.1 Grundbegriffe und Definitionen

Ein allgemeiner Graph G wird beschrieben als ein Tupel (V, E) disjunkter Mengen mit $E \subset [V]^2$. Die Elemente der Menge E sind also eine 2-elementige Teilmenge von V . Die Elemente der Menge V werden Knoten, Ecken oder Punkte (*engl. vertices*) und die Elemente der Menge

E werden Kanten genannt. Häufig wird die Knotenmenge V des Graphen G mit $V(G)$ und die Kantenmenge mit $E(G)$ referenziert [147]. Die Anzahl der Elemente von V wird mit $|V|$ (Kardinalität) und die Anzahl der Elemente von E mit $|E|$ bezeichnet.

Eine Kante $e \in E$ stellt eine Verbindung zwischen zwei Knoten u und v dar, wobei $u, v \in V$. Die beiden Knoten u und v werden als Endknoten der Kante e und die Kante e wird als inzident mit u und v bezeichnet. Die Inzidenz beschreibt also eine Beziehung zwischen Knoten und Kanten. Mit $\zeta(u)$ wird die Menge der zum Knoten u inzidenten Kanten beschrieben. Sind je zwei Knoten von G benachbart, so heißt G vollständig. Zwei Kanten werden als inzident bezeichnet, wenn diese einen gemeinsamen Endknoten haben [146]. Die Anzahl der Knoten $|V|$ eines Graphen heißt Ordnung des Graphen. Ein Graph heißt endlich, wenn V und E endliche Mengen sind, anderenfalls heißt G unendlich [148]. Ein Graph G' wird als Teilgraph (oder Untergraph) eines Graphen G bezeichnet ($G' \sqsubseteq G$), wenn gilt $G' = (V', E')$ mit $V' \subseteq V$ und $E' \subseteq E$.

In gerichteten Graphen (oder Digraphen) sind den Kanten Richtungen zugeordnet. Demgegenüber sind in ungerichteten Graphen den Kanten keine Richtungen zugeordnet; sie sind also bidirektional. Bei einem gerichteten Graphen führt eine gerichtete Kante $e \in E$ zwischen zwei Knoten u und v von u nach v , aber nicht von v nach u . Der Knoten u wird als Anfangsknoten von e und der Knoten v als Endknoten von e bezeichnet. Der Knoten u heißt Vorgänger von Knoten v und v Nachfolger von u . In einem gerichteten Graphen können auch parallele Kanten existieren. Zwei Kanten sind parallel, wenn deren Anfangs- und Endknoten identisch sind. Besteht ein Graph sowohl aus ungerichteten als auch aus gerichteten Kanten, so wird dieser als gemischter Graph bezeichnet.

In einem Graph wird eine Kante als Schlinge bezeichnet, wenn deren beiden Endknoten (ungerichteter Graph) bzw. deren Anfangs- und Endknoten (gerichteter Graph) identisch sind. Ein gerichteter Graph kann zwischen zwei Knoten durchaus mehrere Kanten haben [146]. Solche Kanten werden Mehrfachkanten genannt. Graphen mit Mehrfachkanten werden Multigraphen genannt [149].

Der (Konnektivitäts-) Grad $\epsilon_G(v)$ bzw. $\epsilon(v)$ eines Knotens $v \in V$ ist die Anzahl der mit v inzidenten Kanten, wobei Schlingen doppelt gezählt werden [148]. Ein Knoten v mit dem Grad $\epsilon(v) = 0$ ist isoliert [146]. Bei einem gerichteten Graphen wird die Anzahl gerichteter Kanten, die von einem Knoten weg führen, Ausgangsgrad und die Anzahl der Kanten, die zu einem Knoten hin führen, Eingangsgrad genannt [149].

Wird in einem Graph G jedem Element e der Kantenmenge E ein Gewicht bzw. Kosten c mittels einer Kostenfunktion

$$c : e \in E(G) \rightarrow \mathbb{R} \quad (3.1)$$

zugeordnet bzw. bewertet, so wird G als gewichteter oder bewerteter Graph bezeichnet. Überwiegend werden den Kanten positive Kosten $c \in \mathbb{R}^+$ zugeordnet. Die Kosten bezeichnen beispielsweise die geographische Lage, den Verkehrsfluss über eine Leitung oder die maximale Übertragungsrate einer Leitung [150]. Ist G ein gerichteter Graph, so kann die Zahl der parallelen Kanten auch als Gewicht dargestellt werden.

In einem Graphen wird ein Pfad oder Weg P als eine Folge von Knoten v_1, \dots, v_n bezeichnet, so dass für $1 \leq i \leq n - 1$ gilt: $(v_i, v_{i+1}) \in E$. In einem ungewichteten Graphen wird die Länge eines Weges definiert durch die Zahl der Kanten auf dem Pfad, also $n - 1$ [145]. In einem gewichteten Graphen sind die Kosten eines Weges definiert als die Summe der Kantengewichte aller zugehörigen Kanten:

$$c(P) = \sum_{i=1}^{n-1} c_{v_i, v_{i+1}}, \quad (3.2)$$

wobei $c_{v_i, v_{i+1}}$ die Kosten für die Kante zwischen den Knoten v_i und v_{i+1} sind. Der Weg von einem Knoten u zu einem Knoten v in einem Graphen mit den minimalen Kosten wird kürzester Weg genannt. Ein Weg wird als einfach bezeichnet, wenn alle Knoten auf dem Weg paarweise verschieden sind. Ein einfacher Weg, bei dem der erste und letzte Knoten identisch sind, wird Zyklus genannt. Falls zwischen jedem beliebigen Knotenpaar mindestens ein Weg existiert, so wird der Graph als zusammenhängend bezeichnet. Weiter wird ein zusammenhängender Graph ohne Zyklen (Kreise) als Baum bezeichnet. Algorithmen zur Feststellung, ob ein Graph zusammenhängend ist, werden in [147] vorgestellt. Eine Knotenmenge $U \subseteq V$ eines Graphen G heißt trennend, falls $G - U$ unzusammenhängend ist [148]. Ein Baum, der alle Knoten eines Graphen enthält, wird als aufspannender Baum bezeichnet.

Ein Graph heißt n -fach zusammenhängend, wenn dieser nach dem Entfernen von $n - 1$ beliebigen Kanten immer noch zusammenhängend ist. Zur Feststellung, ob ein Graph 2-fach zusammenhängend ist, kann die bekannte Tiefensuche angewandt werden. Weitere, allgemeine Algorithmen zu dieser Fragestellung werden in [147, 151] diskutiert.

Normalerweise verbindet eine Kante eines Graphen zwei Knoten miteinander. Graphen, bei denen eine Kante beliebig viele Knoten miteinander verbinden kann, werden Hypergraphen genannt [152]. Dementsprechend werden diese Kanten Hyperkanten genannt. Eine solche Hyperkante ist also die Verallgemeinerung einer Kante in einem gewöhnlichen Graphen.

3.1.2 Speicherung von Graphen

Zur Problemlösung bzw. Anwendung von Algorithmen müssen Graphen geeignet abgebildet und gespeichert werden. Hierfür sind geeignete Datenstrukturen notwendig, die häufig auftretende Operationen effizient unterstützen. Die elementaren Operationen für Graphen sind [151]:

- Feststellen, ob ein Paar von Knoten benachbart ist
- Bestimmen der Nachbarn eines Knoten
- Bestimmen der Nachfolger und Vorgänger eines Knoten in einem gerichteten Graphen

Neben der Effizienz der Operationen ist häufig auch der Speicherverbrauch ein wichtiges Kriterium für die Wahl einer geeigneten Datenstruktur. Es besteht eine Wechselwirkung zwischen

Effizienz der Operationen und des Speicherverbrauchs. Eine effiziente Ausführung einer Operation geht oft zu Lasten des Speicherverbrauchs [151].

Die einfachste Art, einen Graphen zu speichern, ist die Kantenliste. Die Kantenliste für einen Graphen G speichert alle Kanten in einer Liste. Eine solche Kantenliste sieht wie folgt aus [148]:

$$|V|, |E|, a_1, e_1, a_2, e_2, \dots, a_{|E|}, e_{|E|}$$

wobei a_i der Anfangs- und e_i der Endknoten der Kante i sind. Bei gewichteten Graphen kann das jeweilige Kantengewicht entweder zwischen dem Anfangs- und Endknoten gespeichert werden oder es wird noch eine Liste mit den Kantengewichten angehängt.

Die am häufigsten verwendete Datenstruktur zur Speicherung eines Graphen G ist eine Adjazenzmatrix (Nachbarschaftsmatrix). Dies ist eine $|V| \times |V|$ Matrix¹ $A(G)$ mit den Elementen $a_{i,j}$. Bei ungewichteten Graphen ist $a_{i,j} = 1$, falls eine Kante von Knoten i nach Knoten j existiert, ansonsten 0. Da in diesem Fall die Matrix nur die Werte 0 und 1 annimmt, kann diese als binäre Matrix dargestellt werden.

Verfügt ein Graph G über Kantengewichte, so repräsentiert das Element $a_{i,j}$ das Gewicht $c_{i,j}$ der Kante von Knoten i nach Knoten j . Existiert zwischen den Knoten i und j keine Kante, so wird, je nachdem welche Gewichtszuordnung für das Problem sinnvoll ist, $a_{i,j} = 0$, $a_{i,j} = -\infty$ oder $a_{i,j} = +\infty$ gesetzt. Falls ein Graph G ungerichtet ist, gilt $a_{i,j} = a_{j,i}$. Die Adjazenzmatrix $A(G)$ eines ungerichteten Graphen ist somit stets symmetrisch und deren Speicherverbrauch kann um etwa die Hälfte reduziert werden, indem nur die obere Dreiecksmatrix, also die Variablen $a_{i,j}$ mit $i \leq j$, gespeichert werden [148].

Um festzustellen, ob eine Kante zwischen den Knoten i und j existiert, muss lediglich das Element $a_{i,j}$ ausgewertet werden. Diese Operation ist also unabhängig von der Größe des Graphen [151]. Um sämtliche Nachbarn eines Knoten i zu bestimmen, genügt die Betrachtung der i -ten Zeile (oder der i -ten Spalte). Für das Auffinden von Vorgängern und Nachfolgern in gerichteten Graphen gilt Ähnliches.

Neben der Adjazenzmatrix wird häufig auch eine Adjazenzliste (Nachbarschaftsliste) verwendet. Bei der Adjazenzliste wird für jeden Knoten die Liste der Nachbarn gespeichert. Die Liste der Nachbarknoten eines Knoten i heißt Nachbarliste von i . Zur Repräsentation von Gewichten existieren wiederum zwei Möglichkeiten. Entweder wird hinter jedem Knoten j auf der Nachbarliste von i das Gewicht $c_{i,j}$ der Kante von Knoten i nach j gespeichert oder es wird eine kompatible Liste mit den Kantengewichten angelegt [148]. Der Speicherverbrauch bei einer Adjazenzliste ist gering, jedoch sind die Operationen teilweise aufwändig. Um festzustellen, ob eine Kante von Knoten i nach j existiert, muss im schlechtesten Fall die gesamte Nachbarliste durchlaufen werden.

¹Bei einer $m \times n$ Matrix entspricht m der Zeilenanzahl (Zeilendimension) und n der Spaltenanzahl (Spaltendimension).

Eine weitere Möglichkeit, einen Graphen zu speichern, ist die Inzidenzmatrix (Beziehungsmatrix). Bei dieser wird ein Graph G durch eine $|V| \times |E|$ Matrix $I(G)$ beschrieben. In dieser werden die Beziehungen der Knoten zu den Kanten eingetragen.

Die oben vorgestellten und weitere Datenstrukturen werden in [147, 153] hinsichtlich ihrer Effizienz und ihres Speicherverbrauchs verglichen.

3.1.3 Minimal aufspannende Bäume

Ein Problem bei der Planung und Optimierung von Kommunikationsnetzen ist das Problem des minimal aufspannenden Baumes oder minimalen Spannbaumes (*engl. Minimal Spanning Tree, MST*). Ein minimal aufspannender Baum verbindet mit minimalen Kosten alle Knoten eines gewichteten, zusammenhängenden Graphen $G = (V, E)$ baumförmig miteinander. Es gilt also:

$$\min \left\{ \sum_{e \in E(G_T)} c_e \right\} \quad (3.3)$$

wobei G_T ein Baum und ein Teilgraph von G mit $V(G_T) = V(G)$, $E(G_T) \subseteq E(G)$ ist.

Häufig sind bei Fragestellungen zur Topologieoptimierung lediglich die Knoten V sowie für alle Knotenpaare $(u, v) \in V$ die Verbindungskosten gegeben. Des Weiteren sind alle Verbindungen zulässig, d. h. es sind Kanten zwischen allen Knotenpaaren möglich. Gesucht ist dann ein resultierender Baum G_T mit minimalen Kosten, der alle Knoten V miteinander verbindet, wobei die Verzweigungspunkte in den Knoten sind. Dies stellt eine Verallgemeinerung des Problems des minimal aufspannenden Baumes dar.

Die bekanntesten Verfahren zur Bestimmung eines minimal aufspannenden Baumes stammen von Kruskal [154] und Prim [155]. Der Algorithmus von Kruskal ordnet die Kanten aufsteigend nach deren Gewicht $c_{e_1} \leq c_{e_2} \leq \dots \leq c_{e_{|E|}}$ und startet mit einer leeren Kantenmenge. Danach wird iterativ die Kante mit dem geringsten Gewicht hinzugefügt, die keinen Zyklus induziert. Dies wird solange wiederholt, bis alle Kanten hinzugefügt wurden.

Der Algorithmus von Prim, dargestellt in Algorithmus 3.1, wählt einen beliebigen Knoten $s \in V$ als Startknoten aus. Anschließend wird der Baum iterativ durch Hinzufügen von Kanten aufgebaut. In jedem Schritt wird die Kante $\{u, v\}$ zum Baum hinzugefügt, die die geringsten Kosten unter allen Kanten besitzt, die noch nicht Teil des Baumes ist und von der jeweils ein Knoten zum Baum gehört. Dies wird solange wiederholt, bis der Baum vollständig ist, d. h. alle Knoten verbindet. Im Gegensatz zum Algorithmus von Kruskal bilden beim Algorithmus von Prim die ausgewählten Kanten zu jedem Zeitpunkt einen Baum. Zwar wird beim Algorithmus von Prim das Sortieren der Kanten vermieden, jedoch ist die Auswahl der Kanten aufwändiger.

Das MST Problem lässt sich verallgemeinern. Beim Steinerbaumproblem muss eine beliebige Menge von Knoten $V'(G_T) \subseteq V(G)$ mit minimalen Kosten baumförmig verbunden werden, wobei G ein gewichteter, zusammenhängender, ungerichteter Graph ist. Dieser minimal aufspannende Baum G_T wird Steinerbaum (*engl. Steiner Tree, ST*) genannt. Zum Erstellen von G_T

Algorithmus 3.1 Prim's Algorithmus zur Bestimmung eines minimal aufspannenden Baumes [147]

Input: Zusammenhängender, ungerichteter Graph $G = (V, E, c)$

Output: Kantenmenge $E_T \subseteq E$ eines minimal aufspannenden Baumes

- 1: Wähle $s \in V$ beliebig
 - 2: Setze $E_T \leftarrow \emptyset$ und $U \leftarrow \{s\}$
 - 3: **while** $U \neq V$ **do**
 - 4: Wähle eine Kante $\{u,v\} \in \zeta(U)$ mit geringsten Kosten. Sei $u \in U$ und $v \in V \setminus U$.
 - 5: Setze $E_T \leftarrow E_T \cup \{u,v\}$ und $U \leftarrow U \cup \{v\}$
 - 6: **return** E_T
-

können Knoten aus $V \setminus V'$ verwendet werden. Diese Knoten werden *Steinerknoten* oder *Steinerpunkte* genannt. Jeder Steinerbaum G_T ist also ein Teilgraph von G . Enthält $V'(G_T)$ alle Knoten V , so ist jeder minimal aufspannende Baum ein minimaler Steinerbaum. Enthält die Menge V' genau zwei Elemente $|V'| = 2$, so ist jeder kürzeste Weg zwischen diesen beiden Knoten ein minimaler Steinerbaum [147]. Die maximale Zahl an Steinerpunkten in einem Graph G beträgt $|V| - 2$. Ein Steinerbaum mit der maximalen Anzahl von Steinerpunkten wird vollständiger Steiner Baum (*engl. full Steiner Tree*) genannt [156].

Auch das Steinerbaumproblem kann verallgemeinert werden. Dabei wird eine unendliche Menge von Knoten V angenommen. Lediglich die Menge der zu verbindenden Knoten $V' \subseteq V$ muss endlich sein und es muss eine Kostenfunktion $c : E(G) \rightarrow \mathbb{R}^+$ gegeben sein. In praktischen Problemen dienen häufig die Entfernungen als Kosten. Bei Fragestellungen zur Topologieoptimierung müssen in der Regel die Vernetzungskosten minimiert werden. Dabei ist die Position der Endknoten (Endsysteme) vorgegeben, wohingegen die Vermittlungsknoten, hier die Steinerknoten, beliebig in der Ebene positioniert werden können. Zur kostenoptimalen Vernetzung müssen dann die Endknoten durch Bestimmen der Steinerknoten baumförmig miteinander vernetzt werden. Zur Lösung spezieller, teilweise eingeschränkter Steinerbaumprobleme existiert in der Literatur [147, 156–159] eine Vielzahl von (Näherungs-) Verfahren.

3.1.4 Kürzeste Wege

Das Problem des kürzesten Weges oder Pfades (*engl. shortest path problem*) beschäftigt sich mit dem Bestimmen des kürzesten Weges innerhalb eines Graphen von einem Startknoten (Quelle) zu einem Zielknoten (Senke). Dabei ist der kürzeste Weg der Weg mit den geringsten Kosten (siehe Abschnitt 3.1.1). Das Problem des kürzesten Weges tritt in mehreren Varianten auf [147, 151]:

- Kürzester Weg zwischen zwei gegebenen Knoten (*engl. single pair shortest path problem*)
- Kürzeste Wege zwischen einem Knoten und allen anderen Knoten (*engl. single source shortest path problem*)
- Kürzeste Wege zwischen allen Knotenpaaren (*engl. all pairs shortest path problem*)

Für alle Varianten existieren Algorithmen, die teilweise auf un-/gerichteten und un-/gewichteten Graphen angewandt werden können. Weiter kann zwischen negativen und nicht negativen Gewichten unterschieden werden. Sämtliche Verfahren funktionieren nach einem ähnlichen Prinzip. Sie starten beim Startknoten und gehen Schritt für Schritt den kürzesten Weg zu jedem Knoten in aufsteigender Ordnung [160]. Sobald der Zielknoten erreicht ist, ist eine Lösung gefunden und die Suche wird beendet. Diesem Prinzip liegt die Tatsache zugrunde, dass Teilwege kürzester Wege wiederum kürzeste Wege sind. Ein kürzester Weg, der die Knotenfolge u, v, w aufweist, besteht aus der Folge der kürzesten Pfade (u, v) und (v, w) [150].

Das Bestimmen eines kürzesten Weges spielt in vielen praktischen Problemen eine wichtige Rolle; beispielsweise bei der Verkehrslenkung in Kommunikationsnetzen, Routenplanung in Verkehrsnetzen oder in der kombinatorischen Optimierung. Aufgrund der Signifikanz des Problems und der weiten Verbreitung existieren viele Algorithmen, die für spezielle Probleme optimiert wurden. Eine ausführliche Diskussion ist zu finden in [147, 151, 161]. Im Folgenden werden nur die für diese Arbeit wesentlichen Algorithmen vorgestellt.

Algorithmus von Dijkstra Zur Bestimmung des kürzesten Weges zwischen zwei gegebenen Knoten s und d in einem zusammenhängenden Graphen mit positiven Kantengewichten kann der Algorithmus von Dijkstra [162] angewandt werden. Dieser ist ähnlich zu Prim's Algorithmus (siehe Abschnitt 3.1.3) und baut ebenfalls schrittweise einen Baum auf; in diesem Fall mit den kürzesten Wegen. Zu Beginn besteht die Menge E' aus allen Knoten des Graphen G , also ($E' = E$). Wobei die vorläufigen Kosten zum Startknoten s auf 0 und zu allen anderen auf unendlich ∞ gesetzt werden. Danach wird die Menge E' sukzessiv abgebaut. Es wird jeweils der Knoten v , der mit geringsten vorläufigen Kosten erreicht werden kann, aus E' entfernt (im ersten Schritt also der Startknoten s). Dies sind auch die geringsten Kosten zu diesem Knoten. Als nächstes werden die vorläufigen Kosten zu den in E' verbliebenen Knoten aktualisiert, indem festgestellt wird, ob das Verwenden des soeben aus E' entfernten Knotens v zu einem kürzeren Weg führt. Dieses Vorgehen wird solange wiederholt, bis der Zielknoten d erreicht ist.

Falls auch negative Kantengewichte im Graphen G vorkommen, arbeitet der Algorithmus von Dijkstra nicht korrekt. In diesem Fall kann der Algorithmus von Bellman und Ford [163, 164] angewandt werden. Dieser kann auch, sofern keine Zyklen im Graphen enthalten sind, negative Kantengewichte berücksichtigen.

Algorithmus von Suurballe Gelegentlich muss nicht nur ein kürzester Weg zwischen einem Knotenpaar, sondern es müssen mehrere disjunkte kürzeste Wege P_1, \dots, P_n berechnet werden. Ein bekannter Anwendungsfall ist das Bestimmen einer kostengünstigen ausfallsicheren Topologie. Es wird zwischen kanten- und knotendisjunkten Wegen unterschieden. Der Algorithmus von Suurballe [165, 166] bestimmt zwei knotendisjunkte Wege ($P_1 \cap P_2 = \emptyset$) mit minimalen Gesamtkosten [167]. Hierfür bestimmt der Algorithmus einen kürzesten Weg mit einem einfachen Algorithmus und versucht anschließend basierend auf einem modifizierten Graphen einen knotendisjunkten, *zweit-kürzesten* Weg zu finden. Negative Kantengewichte können durch eine Transformation des Graphen eliminiert werden. Suurballe [165] stellt eine geeignete Transformation vor, so dass alle Kanten nach der Transformation über ein positives Gewicht verfügen. Durch Erweiterungen des Algorithmus von Suurballe können k -kürzeste kanten- oder knotendisjunkte Wege bestimmt werden [165, 168]. Jedoch sind derartige Verfahren sehr komplex und rechenintensiv.

Algorithmus von Floyd und Warshall Gelegentlich müssen die kürzesten Wege zwischen allen Knotenpaaren in einem Graphen bestimmt werden. Ein naiver Ansatz wäre, durch mehrmaliges Anwenden des Algorithmus von Dijkstra oder von Bellman und Ford alle kürzesten Wege zu berechnen, indem jeder Knoten einmal als Startknoten s ausgewählt wird. Allerdings existiert ein effizienteres Verfahren für dieses Problem – der Algorithmus von Floyd und Warshall [169, 170]. Dieser Algorithmus setzt einen gewichteten, zusammenhängenden Graphen G ohne Zyklen mit negativen Kosten voraus [147]. Die Idee des Algorithmus basiert wiederum auf der Tatsache, dass wenn der kürzeste Weg von Knoten u nach Knoten w durch Knoten v geht, bereits die enthaltenen Teilwege von u zum Knoten v und von Knoten v zum Knoten w minimal sind. Der Algorithmus berechnet aus bereits bekannten Wegen schrittweise alle kürzesten Wege, die über einen bestimmten Knoten gehen. Dieses Vorgehen wird für alle Knoten wiederholt. Die meisten Implementierungen bestimmen die kürzesten (Teil-) Wege rekursiv. Das Ergebnis ist eine $|V| \times |V|$ Kostenmatrix, oder auch Distanzmatrix genannt, die die Kosten der kürzesten Wege (u,v) und die Kosten der kürzesten Zyklen (u,u) , falls solche existieren, enthält.

Mehrere Algorithmen zur Bestimmung kürzester Weg werden hinsichtlich ihrer Komplexität und der Einschränkungen in [147] verglichen. Dort und in [143, 151, 161] werden weitere Algorithmen zur Bestimmung kürzester Wege ausführlich diskutiert.

3.2 Kombinatorische Optimierung

Probleme der Topologieoptimierung können auf diskrete kombinatorische Optimierungsprobleme abgebildet werden. Aus diesem Grund werden in diesem Abschnitt die Grundlagen der kombinatorischen Optimierung behandelt. Der Abschnitt gliedert sich in drei Teile. Zuerst erfolgt eine Einführung in den Bereich der kombinatorischen Optimierung. Im zweiten und dritten Teil werden exakte Lösungsverfahren und wesentliche Metaheuristiken vorgestellt.

3.2.1 Einführung

Das Ziel der Optimierung ist, eine optimale Lösung (Optimum) für ein formuliertes Problem oder optimale Parameter für ein bestehendes System (Parameteroptimierung [48]) unter gleichzeitiger Berücksichtigung von Nebenbedingungen zu bestimmen. Angewandt wird die Optimierung zur Analyse und Lösung komplexer Problemstellungen in technischen und wirtschaftlichen Bereichen. Die Betriebswirtschaft versucht, Prozesse und den Einsatz von Betriebsmitteln und Ressourcen möglichst effektiv und optimal, d. h. gewinn-maximierend zu gestalten. Dieses Gebiet wird auch als *Operations Research* bezeichnet. In der Logistik wird die Optimierung zur Touren- und Standortplanung eingesetzt. In der Telekommunikation wird versucht, durch optimale Wegfindung die Verzögerungen von Nachrichten in einem Netz zu minimieren.

Ein Optimierungsprozess vollzieht sich nach [171, 172] in sechs Schritten:

1. **Erkennen eines Problems:** Ausgangspunkt des Optimierungsprozesses ist das Auftreten von Entscheidungs- und Handlungsbedarf.

2. **Problemanalyse und -beschreibung:** Bestimmen von Zielen und Handlungsmöglichkeiten sowie Ermittlung von Vorgaben und Nebenbedingungen. Beschreibung des Problems und der Zielsetzung.
3. **Modellierung:** Ausgehend vom deskriptiven Modell wird ein mathematisches Modell formuliert. Es werden die Einfluss- und Zielgrößen identifiziert und ggf. Annahmen getroffen. Das mathematische Modell ist gegenüber der Problembeschreibung prägnanter. Die Gesamtstruktur und (Ursache-Wirkungs-) Zusammenhänge werden offensichtlich.
4. **Datensammlung und -beschaffung:** Für das mathematische Modell sind Daten zu sammeln und zu beschaffen.
5. **Lösungsfindung:** Mit Hilfe eines Lösungsverfahrens (Algorithmus) wird das mathematische Modell unter Verwendung der Daten gelöst. In der Regel erfolgt die Lösung mit geeigneter Software.
6. **Bewertung der Lösung:** Die erhaltene Lösung ist zu analysieren und danach als akzeptabel, modifizierungsbedürftig oder unbrauchbar zu bewerten. Ausgehend vom mathematischem Modell und der Lösung erfolgt ein Planungsvorschlag für das reale Problem.

Die Zahl der Optimierungsansätze und -verfahren ist aufgrund der unterschiedlichen Problemstellungen groß und es besteht kein allgemeingültiges Lösungsverfahren. Allerdings entwickelte sich die lineare Optimierung bzw. lineare Programmierung in den letzten Jahren zu einem beliebten Lösungsverfahren. Voraussetzung für den Einsatz der linearen Programmierung ist, dass sich das Optimierungsproblem in Form einer linearen Zielfunktion $z(\mathbf{x})$ beschreiben lässt. Ziel ist, diese lineare Zielfunktion mittels eines geeigneten Lösungsverfahrens zu optimieren. Die Nebenbedingungen (Restriktionen) werden als lineare Gleichungen und Ungleichungen angegeben. Alle Verfahren der linearen Programmierung gehen von folgendem Grundproblem (Normalform der linearen Programmierung) aus [173]: *Optimiere, d. h. minimiere oder maximiere, die Zielfunktion z*

$$\min \left(z(\mathbf{x}) = \sum_{j=1}^m c_j x_j + c_0 \right) \quad \text{bzw.} \quad \max \left(z(\mathbf{x}) = \sum_{j=1}^m c_j x_j + c_0 \right) \quad (3.4)$$

unter den Nebenbedingungen

$$\sum_{j=1}^m a_{i,j} x_j \leq b_i \quad \text{für } i = 1, \dots, n \quad (3.5)$$

$$x_j \geq 0 \quad \text{für } j = 1, \dots, m \quad (\text{Nichtnegativitätsbedingung}) \quad (3.6)$$

Dabei ist \mathbf{x} der Variablenvektor mit den Entscheidungsvariablen x_j , die reellwertig ($x_j \in \mathbb{R}$) sein müssen. Die Koeffizienten $a_{i,j}$ der Nebenbedingungen sowie die Variablen b_i und c_j der Spaltenvektoren müssen ebenfalls reellwertig ($a_{i,j}, b_i, c_j \in \mathbb{R}$) sein [174].² In der Zielfunktion $z(\mathbf{x})$

²Die oben dargestellte Normalform der linearen Programmierung ist in der Literatur so üblich. Allerdings werden in anderen Teilen der Arbeit mit $a_{i,j}$ die Elemente der Adjazenzmatrix $A(G)$ und mit x_j die x Koordinate des Punktes j bezeichnet.

kann zusätzlich ein absoluter Koeffizient c_0 berücksichtigt werden. Die Schwierigkeit besteht darin, das Optimierungsproblem in linearen Gleichungen und Ungleichungen zu formulieren. Jede Lösung des Problems, die gültig ist, d. h. sie erfüllt die Nebenbedingungen (3.5), wird *gültige Lösung* genannt. Eine *zulässige Lösung* erfüllt zudem die Nichtnegativitätsbedingung (3.6) [171]. Die *optimale Lösung* ist die Lösung, bei der die Zielfunktion ihr Maximum (Maximierungsproblem) oder Minimum (Minimierungsproblem) annimmt, d. h. der Zielfunktionswert maximal oder minimal ist [160].

Viele Optimierungsprobleme aus dem Bereich Operations Research werden als ganzzahliges lineares Programm (*engl. Integer Linear Programm, ILP*) formuliert. Im Vergleich zu einem linearen Programm (*engl. Linear Program, LP*) sind hier nur ganzzahlige ($x_j \in \mathbb{N}$) und binäre ($x_j \in \{0,1\}$) Variablen zugelassen. Ein ILP ist wesentlich schwieriger zu lösen als ein LP [171]. Bei einem (gemischt-) ganzzahligen linearen Programm (*engl. Mixed Integer Linear Program, MILP*) dürfen einige Variablen ganzzahlige und andere kontinuierliche Werte annehmen.

Kombinatorische Optimierungsprobleme können als eine Untermenge der ganzzahlig linearen Probleme betrachtet werden. Bei diesen Problemen werden unterschiedliche Anordnungen, Konfigurationen, Gruppierungen und Strukturierungen von endlich vielen Objekten untersucht [175]. Die meisten kombinatorischen Optimierungsprobleme bestehen aus Reihenfolge-, Auswahl- oder Zuordnungsproblemen und zeichnen sich durch eine hohe Komplexität aus [171, 176]. Bei kombinatorischen Optimierungsproblemen kann auch das Ziel sein, die Anzahl möglicher Konfigurationen zu bestimmen. In Anlehnung an [177] kann ein kombinatorisches Optimierungsproblem allgemein wie folgt definiert werden:

“Gegeben sei eine endliche Menge mit den zulässigen Lösungen L und eine Kostenfunktion $z : L \rightarrow \mathbb{K}$ (algebraischer Körper), die jedem Element $l \in L$ einen Wert zuordnet. Gesucht ist ein Element $l^ \in L$, so dass $z(l^*)$ optimal ist, d. h. minimal oder maximal.”*

3.2.2 Exakte Lösungsverfahren

Zum Lösen kombinatorischer Optimierungsprobleme in endlich vielen Schritten können in Einzelfällen exakte Lösungsverfahren angewandt werden. Domschke und Drexl [171] klassifizieren Verfahren zum exakten Lösen ganzzahliger (linearer) und kombinatorischer Optimierungsprobleme wie folgt:

1. Entscheidungsbaumverfahren
 - Vollständige Enumeration
 - Unvollständige (begrenzte) Enumeration
 - Verfahren der dynamischen Optimierung
2. Schnittebenenverfahren
3. Kombinationen aus (1) und (2)

Bei der vollständigen Enumeration wird für alle Elemente (Anordnungen) $l \in L$ der Wert der Zielfunktion berechnet und somit das optimale Element l^* bestimmt. Allerdings haben anwendungsrelevante Optimierungsprobleme häufig einen großen Lösungsraum, so dass dieses Verfahren quasi nicht praktikabel ist. Zudem ist die Menge der zulässigen Lösungen häufig nicht implizit gegeben, sondern durch Eigenschaften charakterisiert [177].

Bei der unvollständigen Enumeration wird zur Bestimmung des optimalen Elementes l^* der Suchraum geeignet eingeschränkt. Das bekannteste Verfahren ist das *Branch-and-Bound* Verfahren. Dieses zerlegt das Problem in disjunkte Teilprobleme, die eine Vereinfachung des Ausgangsproblems darstellen. Diese Teilprobleme werden weiter aufgeteilt, so dass ein (Lösungs-) Baum entsteht. Um abzuschätzen, ob eine optimale Lösung in einem Zweig von Teilproblemen enthalten ist oder ob ein bestimmter Zweig des Baumes bei der Suche vernachlässigt (“abgeschnitten”) werden kann, werden untere und obere Schranken der Zielfunktionswerte berechnet. Schranken können durch Weglassen, Lockern oder Kombinieren von Restriktionen berechnet werden. In [171, 178, 179] sind das Prinzip des Branch-and-Bound Verfahrens und dessen Anwendung an einem Beispiel ausführlich beschrieben.

Mit Verfahren der dynamischen Optimierung können strukturierte Optimierungsprobleme effizient gelöst werden. Dazu muss die Struktur des Problems in gewissem Sinne dynamisch interpretierbar sein [179], d. h. es müssen Zustände erkennbar sein und Entscheidungen, die den Weg durch den Lösungsraum steuern, zur Laufzeit getroffen werden. In Abhängigkeit von einer Entscheidung ergibt sich ein neuer Zustand (Folgezustand). Ausgehend von diesem Folgezustand wird erneut eine Entscheidung getroffen und verändert somit den Zustand weiter. Es wird quasi bei jedem Zustand eine Entscheidung getroffen, die bestrebt ist, das Problem optimal zu lösen. Rekursive Verfahren sind typisch für dynamische Optimierungsverfahren.

Bei den Schnittebenenverfahren (*engl. cutting plane algorithm*) wird das ganzzahlige lineare Optimierungsproblem ohne die Ganzzahligkeitsbedingung betrachtet und versucht, dieses durch schrittweises Hinzufügen von Restriktionen solange einzuschränken, bis eine zulässige ganzzahlige Lösung gefunden wurde (falls eine solche existiert) [179]. Die Schnittebenenverfahren liefern häufig gute obere und untere Schranken, sind jedoch nicht ausreichend. Aus diesem Grund werden diese mit dem Branch-and-Bound Verfahren zum sogenannten *Branch-and-Cut* Verfahren kombiniert.

3.2.3 Heuristiken und Metaheuristiken

Die meisten ganzzahligen und kombinatorischen Optimierungsprobleme gehören zu den \mathcal{NP} -schweren Problemen [171]. Der Lösungsaufwand steigt also mit der Problemgröße exponentiell, so dass *große*, anwendungsrelevante Probleminstanzen häufig nicht in akzeptabler Zeit lösbar sind. Zudem kann häufig ein komplexes praktisches Optimierungsproblem nicht mit allen Restriktionen mathematisch exakt beschrieben werden [179]. Aus diesen Gründen ist das exakte Lösen unrealistisch und es werden heuristische Lösungsverfahren (Heuristiken) eingesetzt.

Heuristiken versuchen, für ein spezifisches Optimierungsproblem mit geringem Rechenaufwand und kurzer Laufzeit eine zulässige, gute, d. h. nahezu optimale Lösung nach einer gegebenen Vorschrift iterativ zu berechnen. Zwar können solche Verfahren das Erreichen der optimalen

Lösung nicht garantieren, finden dennoch eine Lösung, die der optimalen Lösung häufig sehr nahe, im Idealfall sogar identisch mit ihr ist. Die Berechnung kann unter anderem mit Hilfe von Schätzungen, Zufällen und intelligentem Auswählen erfolgen. Da es sich bei Heuristiken um nichtanalytische Verfahren handelt, kann keine Aussage über die Güte der gefundenen Lösung getroffen werden und es fehlt im Gegensatz zu Näherungsverfahren ein Konvergenzbeweis [180, 181]. Die Güte der gefundenen Lösung kann also weder mathematisch bewiesen noch näherungsweise abgeschätzt werden. Hingegen kann bei Näherungsverfahren nach Abbruch des Verfahrens ein Bereich angegeben werden, in dem die gefundene Lösung liegen muss. Jedoch kann die Beschaffenheit einer Heuristik untersucht werden, indem für *kleine* Probleminstanzen das Optimum bestimmt und mit dem gefundenen Ergebnis der Heuristik verglichen wird. Dieses Vorgehen entspricht vielmehr einem experimentellen Nachweis als einem mathematischen Beweis, ermöglicht jedoch eine Abschätzung der Güte der Ergebnisse. Ein solches Vorgehen wird vom Autor et al. in [3, 5] und in Abschnitt 6.2.1 angewandt.

Nach [180] ist eine Heuristik durch folgende Eigenschaften gekennzeichnet:

1. Es werden **nicht willkürliche Entscheidungsoperatoren** verwendet, die bewirken,
2. dass **potentielle Lösungen vom Suchprozess ausgeschlossen** werden und
3. es kann aufgrund des fehlenden Konvergenzbeweises **keine Lösungsgarantie** gegeben werden.

Im Allgemeinen werden Heuristiken für ein spezifisches Optimierungsproblem oder für eine Problemklasse entworfen (*special-purpose heuristic* [182]). Im Gegensatz dazu können *Metaheuristiken* nicht nur auf spezifische Probleme angewandt werden. Sie definieren vielmehr eine abstrakte Folge von Operationen (Schritten), die (theoretisch) auf beliebige Optimierungsprobleme oder Problemklassen angewandt werden können (*general-purpose heuristic* [182]). Allerdings muss ein Teil der Operationen wiederum problemspezifisch implementiert werden. Oft werden diese problemspezifischen Teile wiederum als Heuristiken definiert und implementiert. Genauso wie bei Heuristiken kann auch bei Metaheuristiken kein Konvergenzbeweis geführt und lediglich mit Hilfe von Experimenten eine Aussage über die Qualität der gefundenen Lösung gemacht werden. Prinzipiell hängt die Qualität und die Laufzeit von Metaheuristiken von der Definition und Implementierung der problemspezifischen Operationen und der Parametrisierung ab.

Viele Heuristiken und Metaheuristiken basieren auf dem Prinzip der Nachbarschaftssuche. Die Idee ist, dass die Suche auf benachbarte Lösungen der schon erreichten guten Lösungen begrenzt wird (Strategie der kleinen Schritte) [180]. Bei dieser Strategie muss eine Folge problemspezifischer Operationen $p(l)$ definiert werden, mittels derer aus einer zulässigen Lösung $l \in L$ eine zulässige Nachbarschaftslösung erzeugt werden kann. Eine wichtige Voraussetzung ist die Existenz sogenannter Ähnlichkeitsbeziehungen zwischen Lösungen. Zum Erzeugen von Nachbarschaftslösungen werden häufig einfache Operationen angewandt. Diese können beispielsweise bei einem Reihenfolgeproblem darin bestehen, dass die Reihenfolge der Elemente einfach verändert wird. Bei Gruppierungsproblemen kann durch Zuweisen eines Elementes zu

Algorithmus 3.2 Allgemeines Nachbarschaftssuchverfahren für ein Minimierungsproblem

```

1: Erzeuge zulässige initiale Lösung  $l \in L$                                 ▶ Eröffnungsverfahren
2: Setze  $l^* \leftarrow l$                                                 ▶  $l^*$  (aktuell) beste Lösung
3: repeat
4:    $l \leftarrow p(l^*)$                                                 ▶ Erzeuge zulässige Nachbarschaftslösung
5:   if  $z(l) \leq z(l^*)$  then                                          ▶ Akzeptanzentscheidung
6:      $l^* \leftarrow l$ 
7: until Abbruchkriterium erfüllt
8: return  $l^*$ 

```

einer anderen Gruppe womöglich eine Nachbarschaftslösung erzeugt werden. Die Auswahl eines Nachbarn kann deterministisch oder zufallsgesteuert erfolgen. Bei deterministischer Auswahl wird eine Nachbarschaftslösung nach einem bestimmten Verfahren ausgewählt. Bei einer zufallsgesteuerten Auswahl wird ein beliebiger zulässiger Nachbar ausgewählt.

In Algorithmus 3.2 ist ein allgemeines und einfaches Nachbarschaftssuchverfahren für ein Minimierungsproblem dargestellt. Ausgangspunkt ist eine initiale zulässige Lösung. Zur Erzeugung der initialen zulässigen Lösung muss ein problemspezifisches Verfahren entwickelt werden. Dieses wird Eröffnungsverfahren genannt. Zu Beginn ist die aktuell beste Lösung gleich der initialen Lösung. Anschließend wird versucht, die vom Eröffnungsverfahren gefundene Lösung iterativ zu verbessern. Hierzu wird eine zulässige Nachbarschaftslösung erzeugt und anschließend eine Akzeptanzentscheidung getroffen (Zeile 5). Anstatt nur eine Nachbarschaftslösung zu betrachten, kann auch eine Menge von Nachbarschaftslösungen gleichzeitig erzeugt und die Beste ausgewählt werden. Verfahren, die bei jedem Schritt die jeweils beste Lösung aus einer Menge benachbarter Lösungen auswählen, um so eine möglichst starke Verbesserung zu erzielen, werden auch gierige (*engl. greedy*) Verfahren genannt. Falls mit einer Nachbarschaftslösung eine Verbesserung des Zielfunktionswertes erreicht wird, wird diese Lösung akzeptiert und mit der verbesserten Lösung ebenso fortgefahren. Das Vorgehen wird solange wiederholt, bis ein definiertes Abbruchkriterium erfüllt ist.

Nachbarschaftssuchverfahren, die abbrechen, sobald in einem Schritt keine Verbesserung des Zielfunktionswertes erreicht werden konnte, werden Verbesserungsverfahren genannt. Die somit erzielte Lösung wird als *lokales Optimum* bezeichnet [176]. Ein bekannter Vertreter ist der Bergsteiger-Algorithmus (*engl. hill climbing* oder *ascent*). Anschaulich startet ein Bergsteiger an einer beliebigen Stelle (initiale Lösung) in der Berglandschaft (Suchraum) und bewegt sich stets an die höchste Stelle, die er von seiner aktuellen Position aus erreichen kann. Der Bergsteiger-Algorithmus beschreibt ein Maximierungsproblem. Natürlich kann dieses Verfahren auch für Minimierungsprobleme angewandt werden. In diesem Fall wird es Verfahren des steilsten Abstiegs (*engl. descent*) oder auch Gradientenverfahren genannt.

Der Nachteil eines Verbesserungsverfahrens ist, dass die gefundene Lösung unter anderem von der initialen Lösung abhängt und das Verfahren mit hoher Wahrscheinlichkeit in einem lokalen Optimum stecken bleibt. Es gibt keine Möglichkeit, ein lokales Optimum zu verlassen. Um dies zu vermeiden, wird häufig das Verfahren mit zufällig ausgewählten initialen zulässigen Lösungen wiederholt. Folglich wird mit einer gewissen Wahrscheinlichkeit eine bessere Lösung gefunden.

In den letzten Jahren wurden, basierend auf dem Verbesserungsverfahren, neue Nachbarschaftsuchverfahren mit dem Ziel entwickelt, einerseits einfach, vielseitig und leistungsfähig zu sein, andererseits die Gefahr, in einem lokalen Optimum zu landen, zu verringern. Zu den bekanntesten Vertretern, die diese Anforderungen erfüllen, zählen (i) Simulated Annealing, (ii) Toleranzschwellwertverfahren und (iii) Tabu Search. Diese sind weit verbreitet und werden in vielen Bereichen zur Lösung praktischer, kombinatorischer Optimierungsprobleme angewandt. Die grundlegenden Ideen und die Funktionsweisen dieser Verfahren werden im Folgenden vorgestellt.

3.2.3.1 Simulated Annealing

Das Simulated Annealing (SA) (*dt. Simuliertes Abkühlen*) ist ein Nachbarschaftssuchverfahren, dessen Grundidee aus einem Abkühlungsverfahren aus der Festkörperphysik entstammt. Bei diesem naturinspirierten Verfahren wird versucht, gewünschte Materialeigenschaften durch langsames Abkühlen zu erhalten. Durch das langsame Abkühlen haben die Atome ausreichend Zeit, sich zu ordnen und eine möglichst homogene, für das jeweilige Material typische Kristallstruktur zu bilden. Das Ergebnis ist ein energieärmer, nahezu optimaler Zustand. Umfassender wird der physikalische Hintergrund des SA in [176] erklärt.

Der Abkühlungsprozess wurde in den fünfziger Jahren erstmals von Metropolis et al. [183] mit einem einfachen Algorithmus simuliert. Ausgehend von einem Zustand i eines Festkörpers mit dem Energieniveau W_i wird durch einen Störeinfluss (*engl. perturbation*) ein Folgezustand j erzeugt. Der Zustandsübergang wird durch das Verschieben eines einzelnen Teilchens erzwungen. Das Energieniveau des Folgezustandes ist W_j . Falls die Energiedifferenz $\Delta W = W_j - W_i$ kleiner oder gleich null ist, wird der Zustand j als der aktuelle Zustand akzeptiert. Falls die Differenz größer als null ist, wird der Zustand j mit einer Wahrscheinlichkeit von

$$\exp\left(\frac{\Delta W}{k_B \tau}\right) \quad (3.7)$$

akzeptiert, wobei τ die Umgebungstemperatur, beispielsweise in einem Wärmebad, und k_B die physikalische Boltzmann Konstante ist [184].

Basierend auf diesem Algorithmus entwickelten Kirkpatrick et al. [185] und Černý [186] unabhängig voneinander einen allgemein anwendbaren Algorithmus zur Lösung kombinatorischer Optimierungsprobleme. Zulässige Lösungen, d. h. sie erfüllen alle Nebenbedingungen, eines kombinatorischen Optimierungsproblems entsprechen einem Zustand des physikalischen Systems und die Kosten einer Lösung entsprechen der Energie eines Zustandes. So kann der Algorithmus von Metropolis et al. zum Suchen nach einer optimalen Lösung verwendet werden. Die Analogien zwischen einem physikalischen System und einem Optimierungsproblem sind in [187] ausführlich beschrieben.

In Algorithmus 3.3 ist das SA Prinzip für ein Minimierungsproblem dargestellt. Der Algorithmus beginnt mit einer initialen zulässigen Lösung l . Anschließend werden iterativ zulässige zur aktuell besten Lösung l^* benachbarte Lösungen l zufällig erzeugt und bewertet. Ist die Differenz $z(l^*) - z(l)$ positiv (oder gleich null) und wird folglich im hier betrachteten Minimierungsfall

Algorithmus 3.3 Prinzip des Simulated Annealing für ein Minimierungsproblem

```

1: Erzeuge zulässige initiale Lösung  $l \in L$                                 ▶ Eröffnungsverfahren
2: Setze  $l^* \leftarrow l$                                                 ▶  $l^*$  (aktuell) beste Lösung
3: Initialisiere  $\Xi$                                                     ▶  $\Xi$  Steuerparameter
4: repeat
5:   for  $i \leftarrow 0$  to  $\text{maxIter}$  do
6:      $l \leftarrow p(l^*)$                                               ▶ Erzeuge zulässige Nachbarschaftslösung
7:     if  $(z(l^*) - z(l)) \geq 0$  then                                  ▶ Verbesserung (oder gleichwertige Lösung)
8:        $l^* \leftarrow l$                                               ▶ Akzeptiere Nachbarschaftslösung
9:     else                                                            ▶ Verschlechterung
10:      Erzeuge Zufallszahl  $r \in [0,1]$ 
11:      Akzeptiere  $l$  ( $l^* \leftarrow l$ ) mit einer Wahrscheinlichkeit von  $r < \exp\left(\frac{z(l)-z(l^*)}{\Xi}\right)$ 
12:    Verringere  $\Xi$ 
13:  until Abbruchkriterium erfüllt
14: return  $l^*$ 

```

eine Verbesserung erreicht, so wird die erzeugte Nachbarschaftslösung als aktuell beste Lösung akzeptiert (Zeile 8). Im alternativen Fall entscheidet die Akzeptanzwahrscheinlichkeit

$$P_A = \exp\left(\frac{z(l) - z(l^*)}{\Xi}\right) \quad (3.8)$$

über die Akzeptanz oder Ablehnung der Nachbarschaftslösung, wobei Ξ ein Steuerparameter ist. Ist die Wahrscheinlichkeit größer als eine gleichförmig verteilte Zufallszahl aus dem Intervall $[0,1]$, so wird die erzeugte Nachbarschaftslösung akzeptiert. Der Steuerparameter Ξ entspricht der Temperatur, die den physikalischen Abkühlungsprozess charakterisiert. Dieser Parameter wird in Abhängigkeit von der Schrittzahl entsprechend einer Berechnungsvorschrift abgesenkt. Mit dem Absenken der Temperatur sinkt im Verlauf die Wahrscheinlichkeit, dass eine schlechtere Lösung akzeptiert wird.

Das Akzeptieren schlechterer Lösungen verringert die Wahrscheinlichkeit, in einer frühen Phase des Optimierungsprozesses in einem lokalen Optimum stecken zu bleiben. Mit dem Absenken der Temperatur wird es zunehmend schwieriger, ein lokales Optimum zu verlassen. Da die aktuell beste Lösung nicht unbedingt die beste jemals gefundene Lösung ist, ist es vorteilhaft, auch die beste gefundene Lösung zu speichern [176].

Bei den Parametern eines SA Algorithmus wird zwischen allgemeinen und problemspezifischen Parametern und Operationen unterschieden [187]. Wie bereits erwähnt muss bei Nachbarschaftssuchverfahren ein problemspezifisches Vorgehen zur Erzeugung einer initialen und benachbarten Lösung entwickelt werden. Ebenso muss definiert werden, wie der Zielfunktionswert einer Lösung berechnet wird. Die allgemeinen Parameter eines SA Algorithmus sind nach [187, 188]:

- Anfangswerte des Steuerparameters Ξ
- Berechnungsvorschrift zur Verminderung von Ξ (*engl. cooling ratio*)

- Zahl der Iterationen (Zustandsübergänge) für jeden Wert von Ξ (*engl. temperature length*)
- Abbruchkriterium /-kriterien (*engl. frozen state*)

Diese Menge von Parametern wird Cooling Schedule genannt. Für die Parameterwahl können keine allgemeinen Aussagen gemacht werden. Typischerweise liegt die anfängliche Akzeptanzwahrscheinlichkeit P_A zwischen 0,9 und 0,99. Folglich werden zu Beginn schlechtere Lösungen mit einer hohen Wahrscheinlichkeit akzeptiert. Empfehlungen zur Festlegung des Cooling Schedule werden in [184] gegeben. In der Praxis werden die allgemeinen und problemspezifischen Parameter und Operationen häufig experimentell ermittelt und überprüft [150].

Theoretische Untersuchungen haben gezeigt, dass mit einem SA Verfahren unter gewissen Bedingungen ein globales Optimum angenähert wird. Allerdings ist die Laufzeit eines *exakten* SA Verfahrens exponentiell zur Problemgröße [189]. Da aber kein praktisches Problem bekannt ist, das die Bedingungen für ein Konvergenzverhalten erfüllt, wird SA als eine Metaheuristik betrachtet [176].

Nach Faigle und Schrader [190] müssen beim Entwurf nicht unbedingt die physikalischen Grundlagen des Abkühlungsprozesses eingehalten werden. Es genügt, lediglich die Temperatur langsam genug absinken zu lassen. Das SA zählt zu den am meisten angewandten Metaheuristiken zur Lösung von Optimierungsproblemen. Dies zeigt auch die Literaturliste von Collins et al. [191] mit ca. 500 Veröffentlichungen. Auch zur Optimierung von Kommunikationsnetzen wird SA häufig eingesetzt [192–195].

3.2.3.2 Toleranzschwellwert- und Sintflutverfahren

Im Jahre 1990 veröffentlichten Dueck und Scheuer [196] eine Variante zum SA mit dem Namen Threshold Accepting (*dt. Toleranzschwellwertverfahren*). Bei diesem Verfahren, dargestellt in Algorithmus 3.4, werden ebenfalls iterativ zulässige Nachbarschaftslösungen erzeugt und bewertet. Während SA energetisch günstige Lösungen sofort und ungünstigere mit einer gewissen Wahrscheinlichkeit annimmt, akzeptiert das Toleranzschwellwertverfahren alle Lösungen, bei denen die Differenz zur aktuell besten Lösung unterhalb (Minimierungsproblem) eines Schwellwertes τ bleibt (siehe Zeile 7). Der Schwellwert kann im Verlauf des Optimierungsprozesses gesenkt werden. Der Unterschied zum SA liegt also in der Akzeptanz schlechterer Lösungen. Es ist kein Zufallsexperiment nötig, um eine Akzeptanzentscheidung zu treffen [176].

Das Toleranzschwellwertverfahren ist relativ einfach, aber dennoch in der Lage, gute Lösungen innerhalb kürzester Zeit zu finden. Der Vorteil gegenüber SA liegt in der geringeren Anzahl einzustellender Parameter. Das Bestimmen eines optimalen Cooling Schedule entfällt.

Genau wie das Toleranzschwellwertverfahren wurde auch das Sintflutverfahren von Dueck entwickelt [197]. Dieses Prinzip basiert auf der Idee, dass ein fiktiver Wanderer vor einer langsam ansteigenden Flut (Maximierungsproblem) flieht und er in hohe Gebiete (Optima) läuft. So wird die zufällige Suche im Lösungsraum durch einen modellhaft steigenden Wasserspiegel (Sintflut) mit der Zeit eingeschränkt und eine nahezu optimale Lösung gefunden. Eine Veränderung wird nicht akzeptiert, wenn die Kosten der zufällig erzeugten benachbarten Lösung unterhalb

Algorithmus 3.4 Prinzip des Toleranzschwellwertverfahrens für ein Minimierungsproblem

```

1: Erzeuge zulässige initiale Lösung  $l \in L$                                 ▶ Eröffnungsverfahren
2: Setze  $l^* \leftarrow l$                                                 ▶  $l^*$  (aktuell) beste Lösung
3: Initialisiere  $\tau > 0$                                                 ▶  $\tau$  Toleranzschwellwert
4: repeat
5:   for  $i \leftarrow 0$  to  $\text{maxIter}$  do
6:      $l \leftarrow p(l^*)$                                                 ▶ Erzeuge zulässige Nachbarschaftslösung
7:     if  $(z(l) - z(l^*)) \leq \tau$  then                                  ▶ Akzeptanzentscheidung
8:        $l^* \leftarrow l$                                                 ▶ Akzeptiere Nachbarschaftslösung
9:   Verringere  $\tau$ 
10: until Abbruchkriterium erfüllt
11: return  $l^*$ 

```

des *aktuellen Wasserstandes* (Schranke) liegen. Während der Zeit steigt dieser Wasserstand langsam an. Das Verfahren endet, wenn in allen Richtungen nur noch Wasser ist [198]. Im Vergleich zum Toleranzschwellwertverfahren ist bei diesem Verfahren nur die Akzeptanzregel anders. Das Sintflutverfahren akzeptiert alle Lösungen, die oberhalb des aktuellen Wasserstandes liegen. Wenn eine neue beste Lösung akzeptiert wurde, wird der Wasserstand nach einer definierten Regel erhöht [199]. Zur Lösung eines Minimierungsproblems wird der Wasserstand langsam verringert und es werden nur Lösungen unterhalb des Wasserstandes akzeptiert.

Eine Weiterentwicklung des Sintflutverfahrens ist das Record-to-Record-Travel-Verfahren [197]. Bei diesem Verfahren wird die Schranke in Abhängigkeit des Suchlaufes angepasst. Dadurch wird ein zu schnelles oder zu langsames Ansteigen der Schranke vermieden.

Sowohl das Toleranzschwellwertverfahren als auch das Sintflutverfahren sind deterministisch in der Akzeptanz schlechterer Lösungen. Deren Leistungsfähigkeit wurde von Dueck et al. durch Anwendung auf eine Reihe von Optimierungsproblemen experimentell nachgewiesen [196–198].

3.2.3.3 Tabu Search

Der Metaheuristik Tabu Search (TS) (*dt. Tabu Suche*) liegt keine Analogie aus der Natur zugrunde. Vielmehr basiert sie auf einem Verbesserungsverfahren und verfügt über Mechanismen, um die Gefahr, in einem lokalen Optimum zu landen, zu verringern. Das Verfahren wurde in den achtziger Jahren von Glover [200–202] entwickelt. Ein ähnliches Verfahren mit dem Namen *The Steepest Ascent Mildest Descent Heuristic* wurde 1986 von Hansen [203] vorgestellt. Ein Unterschied zwischen TS und SA ist, dass Akzeptanzentscheidungen deterministisch sind und von keinem stochastischen Prozess beeinflusst werden.

Genauso wie bei jedem Verbesserungsverfahren wird ausgehend von einer initialen zulässigen Lösung iterativ eine Menge von Nachbarschaftslösungen erzeugt und untersucht. Abhängig vom Problem können alle Nachbarschaftslösungen oder nur eine Teilmenge evaluiert werden. Die Teilmenge kann nach bestimmten Kriterien bestimmt oder zufällig ausgewählt werden. In jeder Iteration wird grundsätzlich die aus dieser Menge beste erreichbare Lösung als aktuell beste Lösung ausgewählt. Dies hat gewisse Ähnlichkeit zu einem gierigen Vorgehen.

Während Verbesserungsverfahren abbrechen, sobald keine Lösung mit einem verbesserten Zielfunktionswert gefunden wird, können bei Tabu Search sogenannte Zyklen auftreten. Ein Zyklus ist eine regelmäßig wiederkehrende Folge von Lösungen und macht eine weitere Suche sinnlos, weil nun keine neuen Lösungen gefunden werden können [176].

Im Gegensatz zu einfachen Verbesserungsverfahren versucht Tabu Search lokalen Optima zu entkommen. Hierfür und zur Vermeidung von Zyklen werden die letzten (Teil-) Lösungen oder Schritte der Suche in einer Tabuliste geeignet gespeichert. Diese Liste wird auch Kurzzeitgedächtnis (*engl. short term memory*) [204] genannt. Die in der Liste gespeicherten Lösungen sind für die weitere Suche tabu – daher der Name des Verfahrens. Durch die Liste wird die Suche eingeschränkt. Anstelle einer Menge von Lösungen kann auch eine Menge von Schritten gespeichert werden. Ein Schritt in der Tabuliste kann durch unterschiedliche Eigenschaften einer Lösung gekennzeichnet sein. Beispielsweise kann nach [176] ein Schritt tabu sein, wenn

- der Wert einer Variablen auf k gesetzt wird und sie innerhalb der letzten n Schritte von k ausgehend verändert wurde,
- der Wert einer Variablen auf einen Wert ungleich k gesetzt wird und die Variable innerhalb der letzten n Schritte auf k gesetzt wurde oder
- der Wert einer speziell zu definierenden Funktion auftritt und dieser innerhalb der letzten n Schritte bereits aufgetreten ist.

Dabei ist n die Länge bzw. die Anzahl der Einträge in der Tabuliste. Eine geeignete Länge ist abhängig von der Problemgröße und -art zu bestimmen. Bewährt hat sich eine Länge von $7 \leq n \leq 40$ [176]. Die Länge kann auch während des Ablaufes verändert werden (dynamische Tabulistenlänge). Durch die Tabuliste ergibt sich im schlechtesten Fall ein theoretisch minimaler Zyklus der Länge $n + 1$. Es ist aber nicht zwangsläufig, dass überhaupt ein Zyklus auftritt.

Falls zu einem Zeitpunkt alle möglichen Schritte tabu sind, so wird kein Schritt ausgeführt und der älteste Eintrag aus der Tabuliste entfernt. Durch die Liste können auch bessere Lösungen zeitweise ausgeschlossen werden, die nicht zu Zyklen führen würden. Um dies zu vermeiden, werden zusätzlich Aspirationskriterien [201] definiert, bei denen ein Tabustatus übergangen bzw. aufgebrochen werden kann. Diese Kriterien können nach [176] erfüllt sein, wenn

- eine neue beste bekannte Lösung gefunden wird,
- eine bessere Lösung gefunden wird als vor Verbot des Schrittes vorlag oder
- der Schritt eine Verbesserung ist und er bei Verbot ein verschlechternder war oder umgekehrt.

Für Tabu Search können unterschiedliche Abbruchkriterien definiert werden. Das Verfahren kann nach einer festgelegten Anzahl von Iteration abgebrochen werden, nachdem über eine bestimmte Anzahl von Iterationen keine Verbesserung erzielt wurde oder wenn eine Lösung mit festgesetztem Zielfunktionswert gefunden wurde. Ebenso kann das Verfahren nach einer bestimmten Rechenzeit abgebrochen werden.

Algorithmus 3.5 Prinzip des Tabu Search für ein Minimierungsproblem

```

1: Erzeuge zulässige initiale Lösung  $l \in L$                                 ▶ Eröffnungsverfahren
2: Initialisiere Tabuliste  $T = \emptyset$                                 ▶ Menge der zuletzt betrachteten Lösungen
3: Setze  $l^* \leftarrow l$                                             ▶  $l^*$  (aktuell) beste Lösung
4: repeat
5:    $L' \leftarrow \{p_1(l^*), p_2(l^*), \dots, p_n(l^*)\}$             ▶ Erzeuge Menge benachbarter Lösungen
6:   Setze  $l \leftarrow \arg \min (z(l') : l' \in L' \setminus T)$     ▶ Wähle die beste erlaubte Lösung aus
7:   if  $z(l) \leq z(l^*)$  then                                       ▶ Akzeptanzentscheidung
8:      $l^* \leftarrow l$ 
9:   Aktualisiere  $T$ 
10: until Abbruchkriterium erfüllt
11: return  $l^*$ 

```

Bei Tabu Search kann auch die Häufigkeit des Auftretens bestimmter Eigenschaften der untersuchten Lösungen oder der ausgeführten Schritte ermittelt werden. Hierfür ist ein Häufigkeitsspeicher (*engl. frequency based memory*) notwendig. Dieser wird auch Langzeitgedächtnis (*engl. long term memory*) genannt [204]. Basierend darauf kann die Suche gezielt gesteuert und vielversprechende Bereiche des Lösungsraumes näher untersucht werden. Dieses Vorgehen wird Intensivierung genannt. Ebenso kann der Häufigkeitsspeicher dazu verwendet werden, nach einer gewissen Zeit einen bereits intensiv untersuchten Bereich zu verlassen und die Suche in neue Bereiche des Lösungsraumes zu lenken. Dieses Vorgehen wird Diversifizierung genannt.

Der Algorithmus 3.5 zeigt das Prinzip des Tabu Search für ein Minimierungsproblem. Aus Gründen der Übersichtlichkeit wurden die Intensivierung und die Diversifizierung vernachlässigt. Anfangs wird mit einem geeigneten Eröffnungsverfahren eine zulässige initiale Lösung erzeugt und die Tabuliste T initialisiert. Vereinfacht wird angenommen, dass in der Tabuliste vollständige Lösungen gespeichert werden. In Zeile 5 wird eine Menge L' zulässiger, zu l^* benachbarter Lösungen erzeugt. Idealerweise sind die Elemente der Menge L' verschieden. Anschließend wird in Zeile 6 die beste, nicht in der Tabuliste T enthaltene Lösung ausgewählt und festgestellt, ob eine Verbesserung erreicht wurde. Falls eine Verbesserung erreicht wurde, wird diese Lösung als neue aktuell beste Lösung akzeptiert. Anschließend wird die Tabuliste aktualisiert. Dieses Vorgehen wird solange wiederholt, bis das Abbruchkriterium erfüllt ist.

In den letzten Jahren wurde die Eignung von Tabu Search durch das erfolgreiche Anwenden auf unterschiedliche Optimierungsprobleme nachgewiesen. Ein detaillierter Überblick wird in [204–206] gegeben. Glover und Laguna [204] stellen zudem erweiterte Suchstrategien vor, unter anderem eine probabilistische Lösungs- bzw. Schrittakzeptanz. Auch zur Optimierung von Kommunikationsnetzen wurde Tabu Search bereits häufiger eingesetzt. Beispielsweise nutzen Glover et al. [207] Tabu Search zum Bestimmen einer kostengünstigen Topologie mit einem zentralisierten Dienst und Costamagna et al. [208] nutzen es zur Optimierung von Breitbandnetzen.

3.2.3.4 Bewertung der Metaheuristiken

Beobachtungen, die bei Untersuchungen an praxisrelevanten Optimierungsproblemen und an *künstlich erzeugten Probleminstanzen* gemacht wurden, zeigen, dass Optimierungsprobleme häufig in ihrer Struktur so geartet sind, dass es viele sehr gute Lösungen gibt, die teilweise sehr unterschiedlich aussehen [199]. Diese Erkenntnis vereinfacht das Finden einer nahezu optimalen Lösung erheblich und ermöglicht ein sinnvolles Anwenden von Heuristiken auf viele, teilweise nicht exakt lösbare Optimierungsprobleme.

Eine quantitative Bewertung der behandelten Metaheuristiken ist nicht trivial, da kein Konvergenzbeweis geführt werden kann. Zudem ist die Qualität der Ergebnisse von den problemspezifischen Operationen sowie der Parameterkonfiguration abhängig. Alle behandelten Verfahren wurden bereits mit Erfolg auf unterschiedliche Optimierungsprobleme angewandt, so dass deren Leistungsfähigkeit in der Praxis quasi als nachgewiesen gilt.

In allen Verfahren müssen problemspezifische Schritte zur Erzeugung von Nachbarschaftslösungen definiert werden. In jeder Iteration werden stets eine oder mehrere Nachbarschaftslösungen erzeugt und in Abhängigkeit eines definierten Akzeptanzkriteriums gegebenenfalls als neue aktuell beste Lösung akzeptiert. Heinrici [176] klassifiziert Nachbarschaftssuchverfahren nach ihrer Schrittauswahlmethode und der Schrittakzeptanzmethode. Beim SA und dem probabilistischen Tabu Search ist die Lösungs- bzw. Schrittakzeptanz zufallsgesteuert. Wohingegen beim Tabu Search, beim Toleranzschwellwertverfahren und beim Sintflutverfahren die Akzeptanz deterministisch ist.

Zumindest SA und das Toleranzschwellwertverfahren (bzw. Sintflutverfahren) sind in ihrer Idee und Struktur ähnlich und können als *verwandt* zueinander angesehen werden. Es besteht lediglich ein Unterschied im Akzeptanzverhalten schlechterer Lösungen. Beim Toleranzschwellwertverfahren werden nur geringfügig schlechtere Lösungen akzeptiert, d. h. Lösungen, deren Zielfunktionswerte unterhalb oder oberhalb eines Schwellwertes liegen. Beim SA hingegen werden schlechtere Lösungen zufällig akzeptiert.

Sichtlich anders ist Tabu Search. Dieses Verfahren hat eine Art Gedächtnis und Mechanismen zur Intensivierung und Diversifizierung der Suche. Zudem strebt das Verfahren schnell eine hohe Lösungsqualität an, da in jedem Schritt eine Menge von Nachbarschaftslösungen erzeugt und die jeweils beste, zulässige Lösung ausgewählt wird. Insgesamt verläuft die Suche wesentlich strukturierter und zielgerichteter.

Heinrici [176] vergleicht mehrere Nachbarschaftssuchverfahren hinsichtlich ihrer Leistung und Effizienz miteinander. Für den Leistungsvergleich wählt er ein Problem aus dem Bereich der Produktionsplanung, das sogenannte Bandabgleichproblem. Die Untersuchungen zeigen, dass in den meisten Fällen Tabu Search die besten Ergebnisse findet. Paulli [209] vergleicht SA und Tabu Search an Hand des quadratischen Zuordnungsproblems. Bei diesem Vergleich jedoch liefert SA qualitativ bessere Ergebnisse. Es kann also kein Verfahren als besser bewertet werden, sondern muss für jedes Optimierungsproblem spezifisch untersucht werden.

4 Mathematisches Modell

In diesem Kapitel wird das in dieser Arbeit behandelte Optimierungsproblem analysiert und beschrieben. Es werden die wesentlichen Netzkomponenten und -kosten sowie die Herausforderungen diskutiert. Für die Optimierung müssen Eigenschaften eingebetteter Kommunikationsnetze und Nebenbedingungen berücksichtigt sowie Zielgrößen identifiziert werden. Diese werden in Abschnitt 4.2 ermittelt und geeignet modelliert. Ausgehend davon werden ein mathematisches Modell und die Ziel- bzw. Kostenfunktion formuliert. Letztere bildet später die Grundlage für die Optimierung. Bei der Optimierung müssen neben den Eigenschaften auch Anforderungen berücksichtigt werden. Bei diesem Optimierungsproblem handelt es sich dabei um die Verkehrsanforderungen und die Ausfallsicherheit. Abschließend wird in diesem Kapitel die Komplexität und die Größe des Lösungsraumes abgeschätzt.

Diese Arbeit ist motiviert durch die zuvor aufgezeigte Entwicklung mit der zunehmenden Verbreitung und Bedeutung von Ethernet sowie den Anforderungen und Eigenschaften eingebetteter Kommunikationsnetze. Jedoch sind die im Folgenden beschriebenen mathematischen Modelle und Lösungsverfahren nicht auf Ethernet-Netze beschränkt, sondern grundsätzlich zur Kostenoptimierung paketvermittelnder Vernetzungstechnologien anwendbar, die ausschließlich Vollduplexleitungen verwenden.

4.1 Problembeschreibung

In Kapitel 2 wurden die funktionalen und nichtfunktionalen Anforderungen und deren Bedeutung diskutiert. Durch den Einsatz von Vermittlungsknoten und Vollduplexleitungen werden bei heutigen Ethernet-Netzen die Kosten und der benötigte Platzbedarf im Vergleich zu einem gemeinsam genutzten Medium erhöht. Viele Produkte mit eingebetteten Kommunikationsnetzen werden in einer großen Stückzahl hergestellt, unterliegen einem enormen Wettbewerb und sind somit besonders kostensensitiv. Wie bereits erwähnt spielen vor allem bei Transportmitteln und mobilen Geräten die Größe und das Gewicht eine bedeutende Rolle. Das Finden einer Topologie, die diese Anforderungen möglichst gut erfüllt, stellt ein Optimierungsproblem dar. Dabei sind die Kosten der Topologie zu minimieren.

4.1.1 Vernetzungskosten

Anfangs wurden Ethernet-Netze mit einer Bustopologie realisiert. Dabei sind Kollisionen aufgetreten. Dies führte bei einer steigenden Anzahl von Stationen, die versuchten zu senden, zu

einer geringeren Leistungsfähigkeit. Zur Verbesserung der Leistungsfähigkeit werden heutzutage Vermittlungsknoten und Vollduplexleitungen eingesetzt. Demzufolge besteht ein modernes Ethernet-Netz im Wesentlichen aus (i) den zu vernetzenden Endknoten, (ii) den Vermittlungsknoten auf OSI-Schicht 2 (*engl. switches*) und (iii) den Vollduplexleitungen. Somit betragen die (Anschaffungs- und Installations-¹) Kosten für ein eingebettetes Kommunikationsnetz [3]

$$\underbrace{n \cdot c_N}_{\text{Kosten für die Endknoten}} + \underbrace{m \cdot c_S}_{\text{Kosten für die Vermittlungsknoten}} + \underbrace{l_{\Sigma L} \cdot c_L}_{\text{Leitungskosten } c_{\Sigma L}} \quad (4.1)$$

Vernetzungskosten

mit

- n Anzahl der Endknoten,
- m Anzahl der Vermittlungsknoten,
- c_N Kosten für einen Endknoten inklusive Netzanschlusskosten (vereinfachend wird angenommen, dass alle Endknoten gleich teuer sind),
- c_S Kosten für einen Vermittlungsknoten (vereinfachend werden für alle Vermittlungsknoten identische Kosten angenommen, unabhängig von deren Anzahl der Anschlüsse und Funktionsumfang),
- $l_{\Sigma L}$ Gesamtlänge aller physikalischen Vollduplexleitungen und
- c_L Kosten für eine Längeneinheit einer physikalischen Vollduplexleitung auf eine Kosteneinheit (KE) normiert. Es wird angenommen, dass in einem Netz identische Medientypen (siehe IEEE 802.3 Standard [79]) mit identischen Eigenschaften verwendet werden und die Kosten für einzelne Leitungen nicht abweichen.

Die Anzahl der zu vernetzenden Endknoten n sowie deren Positionen sind vorgegeben und demzufolge invariant. Deshalb besteht bei den Kosten für die Endknoten kein Optimierungspotential. Es kann zwar deren Anzahl mit Verfahren wie der Virtualisierung reduziert werden; jedoch ist dies nicht Gegenstand dieser Arbeit.

4.1.2 Herausforderungen

Die Vernetzungskosten setzen sich aus den Kosten für die Vermittlungsknoten und den Leitungskosten zusammen. Für eine kostenoptimale Topologie müssen diese Kosten minimiert werden:

$$\min(m \cdot c_S + l_{\Sigma L} \cdot c_L) \quad (4.2)$$

¹Da eingebettete Kommunikationsnetze häufig in Serien- und Massenprodukte installiert werden, werden in dieser Arbeit ausschließlich die Anschaffungs- und Installationskosten berücksichtigt.

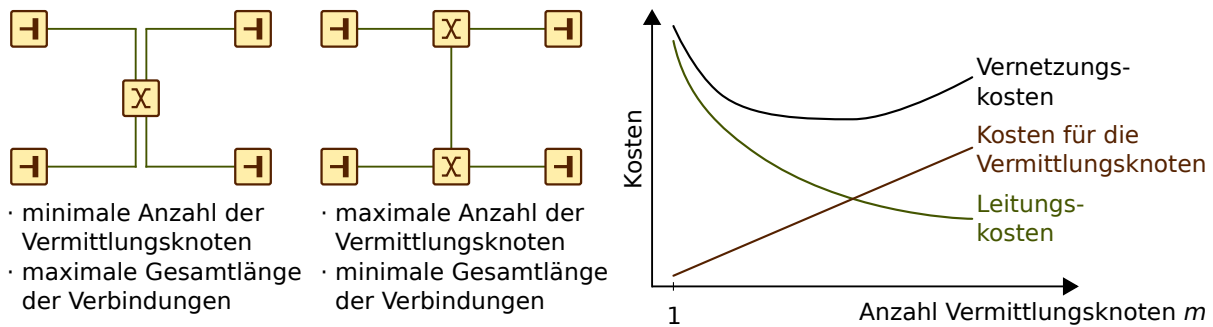


Abbildung 4.1: Zusammenhang zwischen der Anzahl der Vermittlungsknoten und der Gesamtlänge der Leitungen

Die Vernetzungskosten sind minimal, wenn die beiden Summanden minimal sind. Folglich müssen zwei (Teil-) Ziele erreicht werden:

1. Minimierung der Kosten für die Vermittlungsknoten: $\min(m \cdot c_S)$
2. Minimierung der Leitungskosten: $\min(l_{\Sigma L} \cdot c_L)$

Allerdings besteht ein Zusammenhang zwischen diesen Zielen. Diese Abhängigkeit ist in Abbildung 4.1 dargestellt. Für ein vollständig segmentiertes Ethernet-Netz muss mindestens ein Vermittlungsknoten (Sterntopologie $m = 1$) installiert werden. In diesem Fall ist die Gesamtlänge $l_{\Sigma L}$ maximal. Diese kann durch Installation zusätzlicher Vermittlungsknoten verringert werden. Es gilt also folgender Zusammenhang:

“Je größer die Anzahl der Vermittlungsknoten, desto geringer die Gesamtlänge der Leitungen und umgekehrt.”

Diese beiden voneinander abhängigen Ziele sind widersprüchlich zueinander. Mit der Erfüllung eines Zieltes kann das andere nicht erfüllt werden. Aus diesem Grund müssen beide Ziele gemeinsam optimiert werden. Allgemein wird ein Optimierungsproblem mit mehreren voneinander abhängigen Zielen bzw. Zielfunktionen multikriterielles Optimierungsproblem, Optimierungsproblem mit mehrfacher Zielsetzung oder Pareto-Optimierungsproblem genannt [210].

Ein etabliertes Vorgehen zur Lösung multikriterieller Optimierungsprobleme mit Zielkonflikten ist, die interessierenden Ziele als Teilziele mit Gewichtungsfaktoren zu formulieren. Die Teilziele wurden bereits formuliert. Die Gewichtungsfaktoren sind c_S und c_L . Unter Berücksichtigung dieser Faktoren kann mit geeigneten Verfahren eine Kompromisslösung bestimmt werden. Allgemeine Verfahren zur Lösung multikriterieller Optimierungsprobleme sind zu finden in [172, 211, 212].

In der Regel sind die beiden Gewichtungsfaktoren c_S und c_L von mehreren Faktoren abhängig und können im Prinzip nur vom Hersteller bzw. Ausrüster quantifiziert werden. In Abschnitt 5.4 wird ein Verfahren zur Bestimmung einer Kompromisslösung in Abhängigkeit von den beiden Gewichtungsfaktoren entwickelt. Für dieses Verfahren muss jedoch die Gesamtlänge $l_{\Sigma L}$ bzw. die Leitungskosten in Abhängigkeit von einer gegebenen Anzahl von Vermittlungsknoten minimiert werden. Folglich wird zunächst eine Zielfunktion zur Minimierung der Leitungskosten hergeleitet.

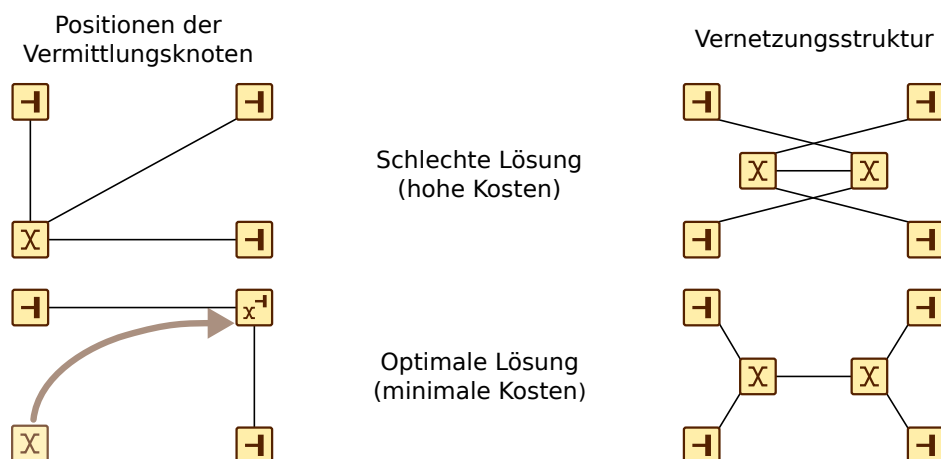


Abbildung 4.2: Einfluss der Positionen der Vermittlungsknoten und der Vernetzungsstruktur auf die Leitungskosten

Ist die Anzahl der Vermittlungsknoten gegeben, so sind die Leitungskosten, wie in Abbildung 4.2 dargestellt, abhängig von

1. den Positionen der Vermittlungsknoten und
2. der Vernetzungsstruktur.

Selbstverständlich sind die Leitungskosten auch abhängig von den Endknotenpositionen. Da diese jedoch invariant sind, können diese nicht verschoben und dadurch die Leitungskosten nicht beeinflusst werden.

Bei der Positionierung der Vermittlungsknoten wird zwischen beliebig positionierbaren und sogenannten *integrierten* Vermittlungsknoten unterschieden. Letztere werden in bestehende Endknoten integriert. Dies hat den Vorteil, dass sie in ein vorhandenes Gehäuse integriert werden können und keine Verkabelung für deren Energieversorgung benötigt wird. Integrierte Vermittlungsknoten sind platzsparender und kostengünstiger. Die Nachteile sind, dass bei Ausfall eines Endknotens auch der Vermittlungsknoten von diesem Ausfall betroffen sein und ausfallen kann. Zudem sind die möglichen Positionen eingeschränkt, was ein geringeres Optimierungspotential hinsichtlich der Vernetzungsstruktur zur Folge hat.

Aus den beiden in Abbildung 4.2 dargestellten Abhängigkeiten ergeben sich zwei Teilprobleme. Zum einen müssen die kostenoptimalen Positionen für eine gegebene Anzahl von Vermittlungsknoten und zum anderen die kostenoptimale Vernetzungsstruktur bestimmt werden. Die Vernetzungsstruktur wiederum kann aufgeteilt werden in:

1. Vernetzungsstruktur zur Anbindung der Endknoten an die Vermittlungsknoten (Zugangsnetz)
2. Vernetzungsstruktur zwischen den Vermittlungsknoten (Vermittlungsnetz)

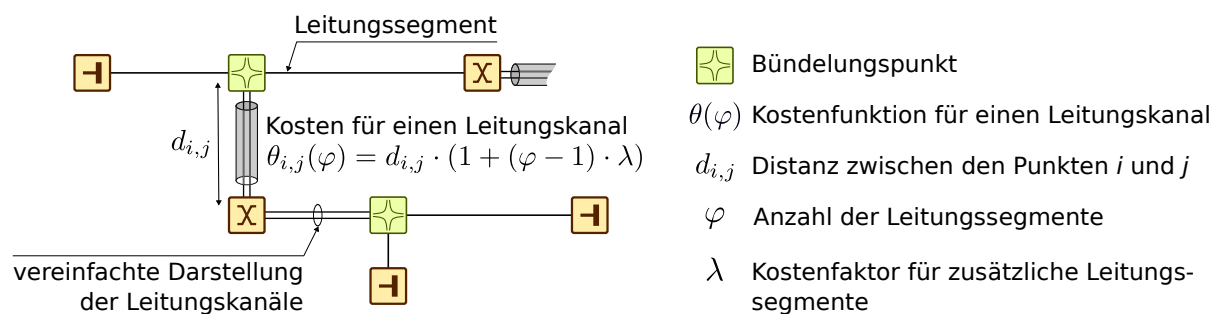


Abbildung 4.3: Modellkomponenten eines Kabelbaumes

Die Herausforderung besteht nun darin, sowohl für das Zugangs- als auch für das Vermittlungsnetz eine kostenoptimale Vernetzungsstruktur zu bestimmen. In Abbildung 4.2 (rechte Seite) ist das Vermittlungsnetz bereits optimal. Lediglich eine schlechte (ungünstige) und die optimale Lösung für das Zugangsnetz sind dargestellt. Die Vernetzungsstruktur ist neben den Positionen der Vermittlungsknoten auch von anwendungsspezifischen Eigenschaften und Anforderungen abhängig. Diese werden im nächsten und in Abschnitt 4.4 beschrieben und modelliert.

4.2 Berücksichtigung von Eigenschaften

4.2.1 Leitungsbaum

Aus Wirtschaftlichkeitsgründen werden bei eingebetteten Kommunikationsnetzen mehrere physikalische Leitungen gebündelt (siehe Abschnitt 2.3). Zusammenhängende Leitungsbündel werden Kabelbaum genannt. Ein solcher Kabelbaum wird dann in das technische System oder die Umgebung eingebettet. Bei der Kostenoptimierung muss diese Eigenschaft zwingend berücksichtigt werden. In Abbildung 4.3 ist ein Ausschnitt eines Kabelbaumes für ein Ethernet-Netz dargestellt. Dieser besteht aus Endknoten, Vermittlungsknoten, Bündelungspunkten, Leitungen, Leitungssegmenten und Leitungskanälen.

Zwischen einem End- und Vermittlungsknoten bzw. zwischen zwei Vermittlungsknoten werden ausschließlich Vollduplexleitungen installiert. Diese Knoten haben damit eine terminierende Eigenschaft. An Bündelungspunkten werden lediglich mehrere Leitungen oder Leitungsbündel zusammengefasst. Eine Leitung endet dort nicht. Die Leitungsteile von und zu den Bündelungspunkten werden Leitungssegmente genannt. Es wird angenommen, dass Leitungen ausschließlich in Leitungskanälen installiert werden.

Der wesentliche Vorteil eines Kabelbaumes besteht darin, dass durch Bündelung mehrerer Leitungen die Kosten verringert werden. Durch diesen Bündelungsgewinn kann es kostengünstiger sein, eine Leitung nicht auf dem (physikalisch) kürzesten Weg zwischen zwei Punkten zu installieren, sondern auf einem Umweg. Dadurch verliert die in der Regel bei der Topologieoptimierung gültige Dreiecksungleichung der Geometrie ihre Gültigkeit. Die Dreiecksungleichung besagt, dass die Summe der Längen zweier Seiten a , b stets größer (oder gleich) der Länge der dritten Seite c ist, d. h. $c \leq a + b$. Ein solcher Fall ist in Abbildung 4.4 dargestellt. Die kürzeste Verbindung (Kante) zwischen dem Endknoten n_1 und dem Vermittlungsknoten s_1 wäre c . Jedoch werden unter Umständen die Kosten eines längeren Weges – in Abbildung

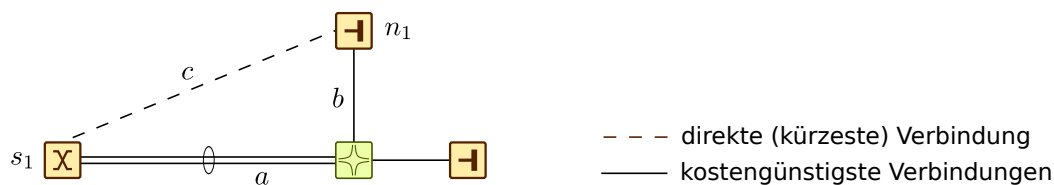


Abbildung 4.4: Beispiel für eine Kostenreduzierung durch Leitungsbündelung (Bündelungsgewinn)

4.4 bestehend aus den Segmenten a und b – durch den Bündelungsgewinn kompensiert. Eine geeignete Kostenfunktion muss also folgende Eigenschaften abbilden:

1. φ gebündelte Leitungen sollten günstiger sein als φ einzelne Leitungen (Bündelungsgewinn).
2. Ein Leitungskanal mit genau einer Leitung ($\varphi = 1$) sollte identische Kosten wie eine einzelne Leitung (außerhalb eines Leitungskanals) verursachen.
3. Ein leerer Leitungskanal ($\varphi = 0$) sollte keine Kosten verursachen.
4. Die Kosten sollten mit zunehmender Anzahl gebündelter Leitungen ansteigen. Folglich sollten die Kosten gegen unendlich gehen, falls die Anzahl gebündelter Leitungen gegen unendlich geht ($\varphi \rightarrow \infty$).

Unter Berücksichtigung dieser Eigenschaften wird folgende lineare Kostenfunktion, die in Abbildung 4.5 dargestellt ist, für einen Leitungskanal zwischen den beiden Punkten i und j mit den Koordinaten (x_i, y_i) und (x_j, y_j) bestehend aus φ gebündelten Leitungen definiert:

$$\theta_{i,j}(\varphi) = \begin{cases} 0 & \text{falls } \varphi = 0, \\ d_{i,j} \cdot (1 + (\varphi - 1) \cdot \lambda) & \text{sonst.} \end{cases} \quad (4.3)$$

wobei $\lambda \in (0, 1]$ den relativen Kostenfaktor für zusätzliche Leitungssegmente der Einheitslänge beschreibt. Somit verursacht jedes zusätzliche Leitungssegment in einem bestehenden

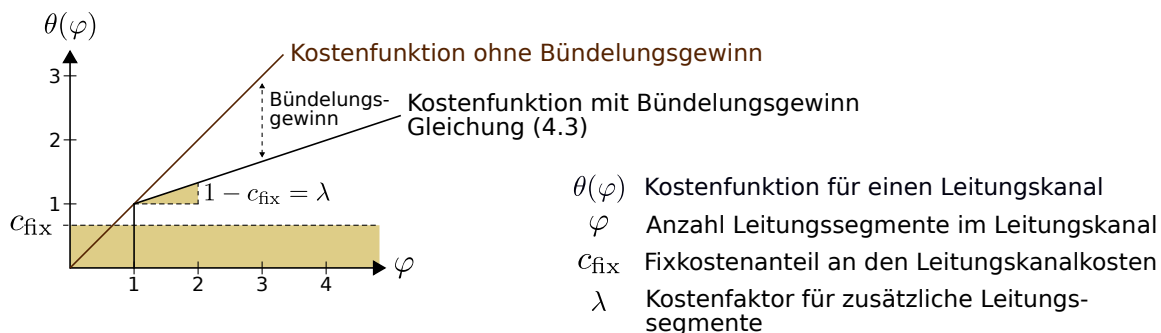


Abbildung 4.5: Kostenfunktion für einen Leitungskanal ohne und mit Bündelungsgewinn

Kanal lediglich die Mehrkosten von $\lambda \cdot d_{i,j}$ Kosteneinheiten. Die Kosten für einen Leitungskanal teilen sich also in Kosten für die Leitungssegmente und einen Fixkostenanteil c_{fix} auf. Der Fixkostenanteil ist unabhängig von der Anzahl der Leitungssegmente. Da ein Leitungskanal mit genau einer Leitung ($\varphi = 1$) identische Kosten wie eine einzelne Leitung verursacht, muss also gelten: $c_{\text{fix}} + \lambda = 1$. Folglich kann die in Gleichung 4.3 formulierte Kostenfunktion wie folgt umgeformt werden: $d_{i,j} \cdot (c_{\text{fix}} + \varphi \cdot \lambda)$. Weitere geeignete Kostenfunktionen, die die oben genannten Eigenschaften weitestgehend erfüllen, werden in [15] diskutiert. Jedoch ist die vorgestellte Kostenfunktion am realitätsnächsten. Die Variable $d_{i,j}$ bezeichnet die Entfernung bzw. Distanz zwischen den beiden Punkten i und j . Die Distanz zwischen zwei Punkten ist von der verwendeten Distanzmetrik abhängig. Eine Distanzmetrik oder ein Distanzmaß ist eine Abbildung $d_{i,j} : K \times K \rightarrow \mathbb{R}^+$ mit K als die Menge der zulässigen Koordinaten in der Ebene. Eine Distanzmetrik muss für beliebige Elemente $i, j, k \in K$ folgende axiomatische Bedingungen erfüllen [213, 214]:

1. $d_{i,i} = 0$: Jeder Punkt hat zu sich selbst den Abstand null
2. $d_{i,j} \geq 0$: Nicht-Negativität
3. $d_{i,j} = 0$ genau dann, wenn $i = j$: Definitheit
4. $d_{i,j} = d_{j,i}$: Symmetriebedingung
5. $d_{i,k} \leq d_{i,j} + d_{j,k}$: Dreiecksungleichung

Die in der Praxis am häufigsten angewandten Distanzmetriken sind [215]:

- Manhattan Distanz: $d_{i,j} = |x_i - x_j| + |y_i - y_j|$ und
- Euklidische Distanz: $d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$

Durch den Bündelungseffekt ist der Ausdruck $l_{\Sigma L} \cdot c_L$ der Gleichung 4.1 zur Berechnung der Leitungskosten nicht mehr gültig. Die Leitungskosten können nun als die Summe der einzelnen Leitungskanalkosten berechnet werden. Zudem ist durch den Bündelungseffekt das Bestimmen einer kostenoptimalen Topologie wesentlich komplexer. Nun müssen zusätzlich zu den Vermittlungsknoten Bündelungspunkte kostenoptimal positioniert werden. Außerdem muss neben deren kostenoptimalen Positionierung auch deren optimale Anzahl bestimmt werden.

Ein Leitungsbaum kann als zusammenhängender, ungerichteter, gewichteter Graph $G = (V, E, c)$ modelliert werden, dessen Leitungskanäle gewichtete Kanten sind [1]. Die Menge der Endknoten, Vermittlungsknoten und Bündelungspunkte entspricht der Menge der Knoten V des Graphen G . Die Kosten eines Leitungskanals (siehe Gleichung (4.3)) entsprechen den Kosten bzw. dem Gewicht einer Kante. Folglich lautet die Funktion, die einer Kante $e = \{i, j\} \mid i, j \in V$ ein Gewicht zuweist, $c : e \rightarrow \theta_{i,j}(\varphi)$.

Graphentheoretisch kann ein Leitungskanal mit mehreren gebündelten Leitungen als Mehrfachkante betrachtet werden. Eine vollständige physikalische Leitung zwischen einem End- und

Vermittlungsknoten bzw. zwischen zwei Vermittlungsknoten kann beliebig viele Bündelungspunkte durchschreiten und aus mehreren Segmenten bestehen. Eine mehrsegmentige Leitung kann somit auch als eine spezielle Art einer Hyperkante betrachtet werden [13], die die beiden Knoten sowie die durchschrittenen Bündelungspunkte miteinander verbindet.

4.2.2 Positionsabhängige Kosten

Beim Entwurf und der Optimierung eingebetteter Kommunikationsnetze muss die Umgebung, in die das Netz später eingebettet wird, berücksichtigt werden, da diese die Vernetzungskosten beeinflusst. Beispielsweise existieren in einem Fahrzeug Bereiche, in denen es unter Umständen aufwändig und somit teuer ist, Komponenten zu installieren. So werden in Bereichen mit extremer Umgebungstemperatur Komponenten mit einem geeigneten Gehäuse benötigt. In Bereichen mit starken elektromagnetischen Feldern werden Leitungen mit einer besseren Abschirmung eingesetzt. In manchen Bereichen kann es sogar nahezu unmöglich sein, Komponenten zu verbauen oder Leitungen zu installieren. Hingegen sind in anderen Bereichen die Kosten wesentlich geringer.

Um solche positions- bzw. ortsabhängigen Kosten zu berücksichtigen, wird die Zielumgebung auf eine zweidimensionale Ebene abgebildet. Diese Ebene wird diskretisiert und in $|X| \cdot |Y|$ Teilbereiche unterteilt. Jedem Teilbereich (Punkt) i mit den Koordinaten (x_i, y_i) wird ein positiver rationaler Wert bzw. Gewichtungsfaktor $\gamma_{x_i, y_i} \in \mathbb{Q}^+$ zugewiesen und in der $|X| \times |Y|$ Kostenmatrix Γ gespeichert. Der Wert γ_{x_i, y_i} beschreibt also, welche Kosten entstehen oder wie diese zu gewichten sind, falls in dem Teilbereich i ein Leitungssegment installiert wird.

Durch die Berücksichtigung der positionsabhängigen Kosten, auch Umgebungskosten genannt, ist der Weg zwischen den Punkten i und j mit den geringsten Kosten $c_{i,j}$ von Interesse und nicht die Manhattan oder Euklidische Distanz. Demzufolge gilt eine modifizierte Kostenfunktion für einen Leitungskanal mit φ gebündelten Leitungen zwischen den Punkten i und j mit $1 \leq i \leq |X|, 1 \leq j \leq |Y|$:

$$\theta_{i,j}(\varphi) = \begin{cases} 0 & \text{falls } \varphi = 0, \\ c_{i,j} \cdot (1 + (\varphi - 1) \cdot \lambda) & \text{sonst.} \end{cases} \quad (4.4)$$

Zur Berechnung des kürzesten Weges zwischen zwei Punkten kann der Algorithmus von Dijkstra (siehe Abschnitt 3.1.4) angewandt werden. Während des Optimierungsprozesses sind nur die Kosten des kürzesten Weges und nicht der Weg selbst von Interesse. Zudem kann es vorkommen, dass derselbe Weg mehrfach berechnet werden muss. Aus diesem Grund können die Kosten für die kürzesten Wege zwischen allen Punkten in der Ebene mit dem Algorithmus von Floyd und Warshall (siehe Abschnitt 3.1.4) vorberechnet und in einer $(|X| \cdot |Y|) \times (|X| \cdot |Y|)$ Distanzmatrix gespeichert werden. Am Ende der Optimierung können dann die genutzten Wege vollständig berechnet und ausgegeben werden.

Die Struktur der diskretisierten Ebene bzw. der Kostenmatrix, die die Umgebungskosten abbildet, ist abhängig vom Anwendungsszenario. So hat die Kostenmatrix eines Fahrzeugs eine andere Struktur als die eines Gebäudes. Normalerweise kann nur der Ausrüster bzw. Hersteller

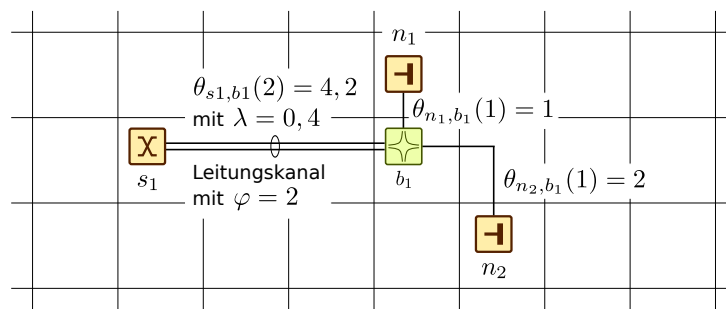


Abbildung 4.6: Beispiel zur Berechnung der Leitungskosten $c_{\Sigma L}$

die Kostenmatrix benennen. Dies kann entweder manuell oder rechnergestützt in einer frühen Phase des Entwurfsprozesses erfolgen. Aus Gründen der Unabhängigkeit der Optimierungsverfahren werden zufällig erzeugte, diskrete Kostenmatrizen verwendet. Diese sind hinreichend, um die Leistungsfähigkeit der Optimierungsverfahren und die Qualität der gefundenen Ergebnisse bewerten zu können [2]. In der Regel sind Bereiche mit ähnlichen Kosten benachbart. Aus diesem Grund sind rein zufällig erzeugte Kostenmatrizen begrenzt sinnvoll. Geeignete Kostenmatrizen müssen eine Autokorrelationsstruktur besitzen. Ein Verfahren zur Erzeugung geeigneter Kostenmatrizen mit Autokorrelationsstruktur veröffentlichten der Autor et al. in [5]. In der vorliegenden Arbeit werden nach diesem Verfahren erzeugte Kostenmatrizen verwendet. Das Verwenden zufällig erzeugter (Kosten-) Matrizen mit einer Autokorrelationsstruktur ist auch in anderen Bereichen etabliert. Beispielsweise werden solche Matrizen beim Simulieren von Mobilfunknetzen zur Modellierung der Abschattung verwendet [216].

Normalerweise sind die Teilbereiche der diskretisierten Ebene größer als die Leitungsmaße. Deshalb werden im weiteren Verlauf die Leitungskosten wie folgt berechnet: Für Leitungsegmente bzw. -kanäle, die in einem Teilbereich enden, werden nur die Hälfte der Kosten berechnet; unabhängig davon, an welcher Stelle sie genau im Teilbereich enden. Zudem können Teilbereiche nur in horizontaler oder vertikaler Richtung überwunden werden (ähnlich zur Manhattan Metrik). Ein Beispiel zur Berechnung der Leitungskosten ist in Abbildung 4.6 dargestellt. Vereinfacht wird in diesem Beispiel eine homogene Kostenmatrix, d. h. alle Teilbereiche haben identische Kosten, mit $\gamma_i = 1 \forall i \in |X| \times |Y|$ angenommen. Die Leitungskosten bzw. Leitungskanalkosten zwischen dem Endknoten n_1 und dem Bündelungspunkt b_1 betragen $\theta_{n_1, b_1}(1) = 0,5 + 0,5 = 1$ (eine Hälfte in jedem Teilbereich). Die Kosten zwischen dem Endknoten n_2 und dem Bündelungspunkt b_1 betragen $\theta_{n_2, b_1}(1) = 2$. Der Kostenfaktor für zusätzliche Leitungsegmente in einem Kanal beträgt $\lambda = 0,4$. Folglich betragen die Kosten zwischen dem Vermittlungsknoten s_1 und dem Bündelungspunkt b_1 $\theta_{s_1, b_1}(2) = 3 \cdot 1,4 = 4,2$. Die gesamten Leitungskosten betragen somit $c_{\Sigma L} = 7,2$.

4.3 Formulierung der Kostenfunktion

Durch den Bündelungseffekt und das Berücksichtigen der Umgebungskosten ist der Ausdruck $l_{\Sigma L} \cdot c_L$ der Gleichung 4.1 zur Berechnung der Leitungskosten nicht mehr gültig. Zur Optimierung der Vernetzungskosten muss unter anderem eine geeignete Kostenfunk-

tion für die Leitungskosten formuliert werden. Ausgangspunkt ist eine Menge zu vernetzender Endknoten $N = \{n_1, n_2, \dots\}$, Vermittlungsknoten $S = \{s_1, s_2, \dots\}$ und Bündelungspunkte $B = \{b_{|S|+1}, b_{|S|+2}, \dots\}$. Diese Mengen sind disjunkt und jedes Element $i \in N \cup S \cup B$ befindet sich an einem diskreten Punkt mit den Koordinaten (x_i, y_i) . Zur Formulierung und Speicherung einer Probleminstanz werden folgende Matrizen verwendet [1, 2]:

- $\Psi = (\psi_{i,j})$ ist eine $|N| \times (|S| + |B|)$ Matrix, die die Existenz von Leitungskanälen zwischen den End- und Vermittlungsknoten bzw. zwischen den Endknoten und Bündelungspunkten speichert. Die binäre Variable $\psi_{i,j}$ kennzeichnet, ob ein Leitungskanal zwischen Endknoten $n_i \in N$ und Vermittlungsknoten $s_j \in S$ oder zwischen Knoten n_i und Bündelungspunkt $b_k \in B$ existiert.

$$\psi_{i,j} = \begin{cases} 1 & \text{falls ein Leitungskanal zwischen } n_i \text{ und } s_j \text{ oder } b_k \text{ existiert,} \\ 0 & \text{sonst.} \end{cases} \quad (4.5)$$

Im Grunde genommen ist die Matrix Ψ , wie im Folgenden dargestellt, zweigeteilt aufgebaut:

$$\begin{array}{c} n_1 \\ \vdots \\ n_{|N|} \end{array} \left(\begin{array}{ccc|ccc} s_1 & \cdots & s_{|S|} & b_{|S|+1} & \cdots & b_{|S|+|B|} \\ \psi_{1,1} & \cdots & \psi_{1,|S|} & \psi_{1,|S|+1} & \cdots & \psi_{1,|S|+|B|} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \psi_{|N|,1} & \cdots & \psi_{|N|,|S|} & \psi_{|N|,|S|+1} & \cdots & \psi_{|N|,|S|+|B|} \end{array} \right)$$

Der linke Teil kennzeichnet die Existenz von Leitungskanälen zwischen den End- und Vermittlungsknoten, der rechte Teil zwischen den Endknoten und Bündelungspunkten.

Bei einer baumförmig vernetzten Topologie, d. h. ohne Ausfallsicherheit, existiert genau ein Leitungskanal zwischen einem Endknoten und entweder einem Vermittlungsknoten oder einem Bündelungspunkt. Daher gilt hier die Bedingung: $\forall i = 1, \dots, |N|$

$$\sum_{j=1}^{|S|+|B|} \psi_{i,j} = 1 \quad (4.6)$$

- $\Delta = (\delta_{i,j})$ ist eine $|N| \times (|S| + |B|)$ Matrix, die die Anzahl der Leitungssegmente zwischen den End- und Vermittlungsknoten bzw. zwischen den Endknoten und Bündelungspunkten speichert. Die binäre Variable $\delta_{i,j}$ kennzeichnet zunächst, ob ein Leitungssegment zwischen Endknoten $n_i \in N$ und Vermittlungsknoten $s_j \in S$ oder zwischen Endknoten n_i und Bündelungspunkt $b_k \in B$ existiert.

$$\delta_{i,j} = \begin{cases} 1 & \text{falls ein Leitungssegment zwischen } n_i \text{ und } s_j \text{ oder } b_k \text{ existiert,} \\ 0 & \text{sonst.} \end{cases} \quad (4.7)$$

Da ein Leitungssegment zwischen zwei Punkten genau dann installiert werden kann, wenn ein Leitungskanal zwischen diesen existiert, sind die beiden Matrizen Ψ und Δ identisch. Für die spätere Berücksichtigung von Anforderungen und für den schrittweisen Aufbau der Vernetzungsstruktur bei den Optimierungsverfahren ist die Verwendung von zwei Matrizen wesentlich verständlicher und hilfreich.

- $\Upsilon = (\nu_{i,j})$ ist eine $(|S| + |B|) \times (|S| + |B|)$ Matrix, die die Existenz von Leitungskanälen zwischen den Vermittlungsknoten und Bündelungspunkten speichert. Die binäre Variable $\nu_{i,j}$ kennzeichnet, ob ein Leitungskanal zwischen Vermittlungsknoten $s_i \in S$ bzw. Bündelungspunkt $b_i \in B$ und Vermittlungsknoten $s_j \in S$ bzw. Bündelungspunkt $b_j \in B$ existiert.

$$\nu_{i,j} = \begin{cases} 1 & \text{falls ein Leitungskanal zwischen } s_i \text{ oder } b_i \text{ und } s_j \text{ oder } b_j \text{ existiert,} \\ 0 & \text{sonst.} \end{cases} \quad (4.8)$$

Es wird angenommen, dass in der Matrix Υ in den Zeilen 1 bis $|S|$ die Vermittlungsknoten und in den Zeilen $|S| + 1$ bis $|S| + |B|$ die Bündelungspunkte angeordnet sind. Das Gleiche gilt für die Spalten. Folglich kennzeichnet der linke obere Teil die Existenz von Leitungskanälen zwischen den Vermittlungsknoten, der rechte untere Teil zwischen den Bündelungspunkten und die beiden übrigen Teile zwischen den Vermittlungsknoten und Bündelungspunkten.

- $\Phi = (\phi_{i,j})$ ist eine $(|S| + |B|) \times (|S| + |B|)$ Matrix, die die Anzahl der Leitungssegmente zwischen den Vermittlungsknoten und Bündelungspunkten speichert. Die positive ganzzahlige Variable $\phi_{i,j} \in \mathbb{N}$ definiert die Anzahl der Leitungssegmente zwischen Vermittlungsknoten $s_i \in S$ bzw. Bündelungspunkt $b_i \in B$ und Vermittlungsknoten $s_j \in S$ bzw. Bündelungspunkt $b_j \in B$.

$$\phi_{i,j} = \begin{cases} \geq 1 & \text{falls ein oder mehrere Leitungssegmente zwischen} \\ & s_i \text{ oder } b_i \text{ und } s_j \text{ oder } b_j \text{ existieren,} \\ 0 & \text{sonst.} \end{cases} \quad (4.9)$$

Da es sich bei den Leitungskanälen um ungerichtete Kanten handelt und im Netz nur Vollplexleitungen eingesetzt werden, sind die beiden Matrizen Υ und Φ symmetrisch. Folglich gilt:

$$\nu_{i,j} = \nu_{j,i} \quad \text{und} \quad \phi_{i,j} = \phi_{j,i}. \quad (4.10)$$

Abschließend wird die (Ziel-) Funktion für die Leitungskosten $c_{\Sigma L}$ wie folgt formuliert

$$c_{\Sigma L} = z(\Delta, \Phi) = \sum_{i=1}^{|N|} \sum_{j=1}^{|S|+|B|} \theta_{i,j}(\delta_{i,j}) + \sum_{i=1}^{|S|+|B|-1} \sum_{j=i+1}^{|S|+|B|} \theta_{i,j}(\phi_{i,j}) \quad (4.11)$$

4.4 Berücksichtigung von Anforderungen

4.4.1 Verkehrsanforderungen

In Abschnitt 2.3 wurden die Verkehrscharakteristika eingebetteter Kommunikationsnetze diskutiert. Für die Übertragung des aufkommenden Verkehrs müssen im Netz ausreichend Ressourcen bereit gestellt werden. Die Ressourcen einfach großzügig zu dimensionieren, würde im Widerspruch zur Minimierung der Kosten stehen. Aus diesem Grund werden beim Netzentwurf

die tatsächlichen Kommunikations- bzw. Verkehrsanforderungen berücksichtigt und die Ressourcen optimal dimensioniert. Falls die Übertragungsrate einer einzelnen Leitung nicht ausreicht, müssen zusätzliche Leitungen installiert werden. Bei Ethernet kann die Übertragungsrate einer Leitung durch Bündelung mehrerer physikalischer Leitungen erhöht werden (siehe Abschnitt 2.6.2.2).

Die Verkehrsanforderungen werden in einer $|N| \times |N|$ Verkehrsanforderungsmatrix \mathbf{R} gespeichert (mit $|N| = \text{Anzahl der Endknoten}$). Die positiven rationalen Elemente $r_{i,j} \in \mathbb{Q}_0^+$ der Matrix \mathbf{R} repräsentieren die Verkehrsanforderungen vom Sender $n_i \in N$ zum Empfänger $n_j \in N$. Häufig ist die Kommunikation zwischen den Endknoten asymmetrisch; beispielsweise bei einer Messsonde oder einer Videoquelle. Demzufolge ist die Verkehrsanforderungsmatrix ebenfalls asymmetrisch.

Neben den Verkehrsanforderungen muss auch die in der Umgebung eingesetzte Technologie bzw. die maximale Übertragungsrate einer Leitung festgelegt werden. Ausgehend von der gewählten Technologie, beispielsweise Fast Ethernet (siehe Abschnitt 2.6.2.1), den Verkehrsanforderungen und den Leitungskanälen kann die Anzahl benötigter physikalischer Leitungen im Zugangs- und Vermittlungsnetz berechnet werden. Falls die Verkehrsanforderungen zu oder von einem Endknoten die verfügbare Übertragungsrate überschreitet, wird eine zusätzliche Leitung oder eine Leitung mit höherer Übertragungsrate installiert. Es werden also ganzzahlige Vielfache von Leitungen mit einer definierten maximalen Basisübertragungsrate im Netz entsprechend den Verkehrsanforderungen installiert. Die Entscheidung, ob zusätzliche physikalische Leitungen benötigt werden, erfolgt somit erst nach der Installation der Leitungskanäle. Durch Berücksichtigung der Verkehrsanforderungen ändert sich die Matrix Δ aus der Gleichung (4.7) wie folgt:

$$\delta_{i,j} = \begin{cases} \geq 1 & \text{falls ein oder mehrere Leitungssegmente zwischen} \\ & n_i \text{ und } s_j \text{ oder } b_k \text{ existieren,} \\ 0 & \text{sonst.} \end{cases} \quad (4.12)$$

Bisher waren die beiden Matrizen Ψ (Leitungskanäle zur Anbindung der Endknoten) und Δ (Anzahl der Leitungen von/zu einem Endknoten) identisch (siehe Abschnitt 4.3). Nun sind lediglich deren Null-Elemente identisch.

Analog zu den positionsabhängigen Kosten sind die Verkehrsanforderungen abhängig vom Anwendungsszenario. Wegen der Unabhängigkeit der Optimierungsverfahren werden bei späteren Untersuchungen die Verkehrsanforderungen zufällig erzeugt. Ein parametrierbares Verfahren zur Erzeugung geeigneter zufälliger Verkehrsanforderungsmatrizen veröffentlichte der Autor in [1]. Bei diesem Verfahren werden die zufällig erzeugten Werte der Verkehrsanforderungsmatrix auf einen normierten Wert der maximalen Übertragungsrate der gewählten Technologie abgebildet.

4.4.2 Ausfallsichere Vernetzung

Die Zuverlässigkeit ist eine wesentliche Anforderung an eingebettete Kommunikationsnetze und kann durch eine ausfallsichere Vernetzungsstruktur erhöht werden. Dazu werden normalerweise Netzkomponenten redundant ausgelegt. Je nach Auslegung können Einzelfehler oder Mehrfachfehler kompensiert werden. Bei Einzelfehlern wird zwischen dem Ausfall einer Leitung (*engl. single-link failure*) und dem Ausfall eines Knotens (*engl. single-node failure*) unterschieden [217]. In dieser Arbeit wird unter Ausfallsicherheit die Fähigkeit, einen Ausfall eines einzelnen Vermittlungsknotens oder eines einzelnen Bündelungspunktes zu kompensieren, verstanden. Ist diese Fähigkeit gegeben, so kann auch der Ausfall einer einzelnen Leitung kompensiert werden.

Ein eingebettetes Kommunikationsnetz muss nicht zwingend vollständig ausfallsicher ausgelegt sein. So kann es durchaus ausreichend sein, nur ausgewählte, sicherheitskritische Endknoten redundant anzubinden oder nur zwischen ausgewählten Vermittlungsknoten disjunkte Pfade zu installieren. In dieser Arbeit wird bei einer ausfallsicheren Vernetzung jedoch angenommen, dass alle Endknoten genau zweifach angebunden werden sollen. Folglich existieren genau zwei Leitungskanäle zwischen einem Endknoten und einem Vermittlungsknoten bzw. Bündelungspunkt. Daher gilt $\forall i = 1, \dots, |N|$:

$$\sum_{j=1}^{|S|+|B|} \psi_{i,j} = 2 \quad (4.13)$$

Zudem wird gefordert, dass der Ausfall eines einzelnen Vermittlungsknotens oder eines einzelnen Bündelungspunktes immer kompensiert werden kann. Folglich müssen zwischen zwei Endknoten mindestens zwei knoten- und kantendisjunkte Pfade existieren. In Abschnitt 2.4.2 wurden hierfür mögliche Schutzpfade diskutiert. Bei eingebetteten Kommunikationsnetzen werden häufig sicherheitskritische Informationen übertragen. Bei einem Ausfall sollten keine zusätzlichen Verzögerungen durch Neuberechnen und/oder Umschalten auf einen Ersatzpfad auftreten. Deshalb wird in dieser Arbeit bei einer ausfallsicheren Vernetzung eine 1+1 Absicherung eingerichtet.

Für die Ausfallsicherheit ist es nicht ausreichend, dass kantendisjunkte Pfade auf der Leitungsebene vorhanden sind. So könnte sich eine absichernde Leitung (Ersatzpfad) im selben Leitungskanal wie die abzusichernde Leitung befinden. Bei Beschädigung bzw. Ausfall wären dann beide Leitungen betroffen. Demzufolge müssen mindestens zwei kantendisjunkte Pfade auf Ebene der Leitungskanäle vorhanden sein.

4.5 Komplexitätsabschätzung

Für eine Topologie mit minimalen Kosten müssen die Vermittlungsknoten kostenoptimal in einer zweidimensionalen Ebene positioniert werden, die optimale Anzahl von Bündelungspunkten bestimmt und diese ebenfalls kostenoptimal positioniert werden. Zudem muss die Vernetzungsstruktur kostenoptimal gestaltet werden. Meistens ist bei Netzdimensionierungsproblemen die Anordnung der Knoten gegeben und es ist lediglich die Vernetzungsstruktur

zwischen diesen zu optimieren. Normalerweise berücksichtigen Netzdimensionierungsprobleme auch keine komplexen Kostenfunktionen, die eine Bündelung von Leitungen bevorzugen.

Mit erheblichen Einschränkungen kann das Optimierungsproblem auf bereits bekannte Probleme abgebildet werden. Ohne Berücksichtigung des Bündelungseffekts, von positionsabhängigen Kosten, von Verkehrsanforderungen, von Ausfallsicherheit und nicht festgelegter Anzahl von Vermittlungsknoten reduziert sich das Problem auf ein Steinerbaumproblem (siehe Abschnitt 3.1.3) [2]. Dieses ist bereits \mathcal{NP} -vollständig [218]. Falls die Anordnung der Vermittlungsknoten gegeben ist und kein Bündelungseffekt, keine positionsabhängigen Kosten, keine Verkehrsanforderungen und keine Ausfallsicherheit berücksichtigt werden, beschränkt sich das Problem auf das Bestimmen eines minimal aufspannenden Baumes [5].

Falls Verkehrsanforderungen berücksichtigt werden und die Anordnung der Knoten gegeben ist, kann das Problem auf das bekannte Optimal Communication Spanning Tree (OCST) Problem abgebildet werden. Ein OCST ist ein aufspannender Baum mit minimalen Kosten, der alle Knoten in einem Netz verbindet und gegebene Verkehrsanforderungen erfüllt [219, 220]. Das OCST Problem berücksichtigt jedoch keinen Bündelungseffekt [1] und ist \mathcal{NP} -schwer [221].

Da bereits verwandte Probleme \mathcal{NP} -schwer sind, ist auch das in dieser Arbeit behandelte Optimierungsproblem \mathcal{NP} -schwer. Um zu verdeutlichen, dass dieses Optimierungsproblem nur mit heuristischen Verfahren gelöst werden kann, wird die Größe des Lösungsraumes für das uneingeschränkte Problem abgeschätzt. Beim uneingeschränkten Problem müssen $|N|$ Endknoten, $|S|$ Vermittlungsknoten und $|B|$ Bündelungspunkte kostenoptimal positioniert werden. Bei Vernachlässigung der Ausfallsicherheit wird zwischen jedem Endknoten und Vermittlungsknoten bzw. Bündelungspunkt genau ein Leitungskanal installiert. Folglich existieren hierfür $|N| \cdot (|S| + |B|)$ Möglichkeiten. Für die baumförmige Vernetzung der Vermittlungsknoten und Bündelungspunkte existieren nach Cayley [222] $(|S| + |B|)^{|S|+|B|-2}$ Möglichkeiten.

Zur Berücksichtigung der positionsabhängigen Kosten wird die Zielumgebung diskretisiert und in $|X| \cdot |Y|$ Teilbereiche unterteilt. Daher ergeben sich für die Positionierung der Vermittlungsknoten und der Bündelungspunkte $(|X| \cdot |Y|)^{|S|+|B|} / |S|! \cdot |B|!$ Möglichkeiten. Die Größe des Lösungsraumes für eine baumförmige Topologie beträgt also:

$$|N| \cdot (|S| + |B|) \cdot (|S| + |B|)^{|S|+|B|-2} \cdot \frac{(|X| \cdot |Y|)^{|S|+|B|}}{|S|! \cdot |B|!} \quad (4.14)$$

Selbst wenn einige Kombinationen als nicht sinnvoll erscheinen, beispielsweise zwei Vermittlungsknoten an der identischen Position, so hat der Lösungsraum auch bei anwendungsrelevanten Problemen eine Größe, die eine vollständige Enumeration unmöglich macht. Den gesamten Lösungsraum parallel auf mehreren Rechnern zu durchlaufen, ist ebenfalls nicht möglich. Selbst durch massive Parallelisierung auf 1000 Rechnern (bzw. Rechenkerne) könnte ein Geschwindigkeitsfaktor von maximal 1000 erreicht werden. Dies fällt aber bei der Größe des Lösungsraumes nicht ins Gewicht. Es müssen also zwingend Verfahren entwickelt werden, die in akzeptabler Zeit Lösungen finden, die möglichst nahe am Optimum sind. Solche Verfahren sind Gegenstand des nächsten Kapitels.

5 Verfahren zur Kostenoptimierung von Topologien

Die Topologie hat Einfluss auf die Leistungsfähigkeit und die Kosten eines Kommunikationsnetzes. Daher ist deren Optimierung ein wichtiger Aspekt beim Entwurf und der Planung. In diesem Kapitel werden neue, anwendungsreife und leistungsfähige Verfahren zur Kostenoptimierung von Topologien eingebetteter Kommunikationsnetze entwickelt. Zuvor werden verwandte Probleme und Verfahren identifiziert und abgegrenzt.

Bei dem vorliegenden komplexen Optimierungsproblem müssen sowohl die Vermittlungsknoten als auch die Bündelungspunkte optimal positioniert werden. Außerdem muss eine kostenoptimale Vernetzungsstruktur erstellt werden. Deshalb werden zuerst Verfahren zur kostengünstigen Positionierung und anschließend Verfahren zur Erstellung einer baumförmigen und einer ausfallsicheren kostengünstigen Vernetzungsstruktur entwickelt. Abschließend wird noch ein Verfahren zum Bestimmen einer Kompromisslösung entwickelt. Letzteres löst den Zielkonflikt zwischen Minimierung der Anzahl der Vermittlungsknoten und Minimierung der Leitungskosten.

5.1 Verwandte Probleme und Verfahren

Probleme der Topologieoptimierung werden häufig auf Optimierungsprobleme aus anderen Bereichen abgebildet. So können unter Umständen Lösungsverfahren aus anderen Bereichen der kombinatorischen Optimierung in angepasster Form zur Topologieoptimierung angewandt werden. In diesem Abschnitt werden verwandte Verfahren vorgestellt und es wird erläutert, warum diese nur eingeschränkt auf das vorliegende Optimierungsproblem anwendbar sind.

Zur Kostenoptimierung von Topologien eingebetteter Kommunikationsnetze müssen unter Berücksichtigung der Umgebungskosten, des Bündelungseffektes und der Verkehrsanforderungen sowie unter Umständen der Ausfallsicherheit

- die optimale Anzahl von Vermittlungsknoten und Bündelungspunkten bestimmt,
- diese kostenoptimal positioniert und
- die Endknoten und die Vermittlungsknoten kostenoptimal miteinander vernetzt werden.

Die Vermittlungsknoten sind optimal zu positionierende, begrenzte Ressourcen. Das optimale Positionieren von Ressourcen tritt auch in anderen Bereichen des Operations Research auf. Zum Beispiel muss bei der Planung logistischer Systeme unter Berücksichtigung betriebswirtschaftlicher Aspekte die optimale Position von (Produktions-) Standorten bestimmt werden. Dabei werden in der Regel Produktions- bzw. Lagerungs-, Errichtungs- und Transportkosten berücksichtigt. In der Literatur ist dieses Problem unter dem Namen *Facility-Location-Problem* bekannt. In [223] stellen Domschke und Drexl Formulierungen und Lösungsverfahren für derartige Probleme vor.

Das Bestimmen der optimalen Vernetzungsstruktur hat gewisse Ähnlichkeiten zu bekannten Transport-, Rundreise- und Tourenplanungsproblemen (*engl. Traveling Salesman Problem*). Die Herausforderung bei diesen Problemen besteht darin, einen Weg (Reihenfolge) mit minimalen Kosten zu finden, der alle Punkte einer gegebenen Menge miteinander verbindet. In Abhängigkeit von der Problemstellung dürfen auf dem Weg Punkte mehrmals oder genau einmal überquert werden. Das Traveling Salesman Problem zählt zu den grundlegenden Problemen der kombinatorischen Optimierung [39]. Entsprechend existiert eine Vielzahl heuristischer und exakter Lösungsverfahren, mit denen auch *schwierige* Probleme mit mehreren tausend Punkten optimal gelöst werden können. Häufig werden Rundreiseprobleme dazu verwendet, neuartige Heuristiken und Metaheuristiken vergleichend zu bewerten (beispielsweise in [196, 199]).

Neben den Lösungsverfahren aus anderen Bereichen der kombinatorischen Optimierung existieren auch spezifische Verfahren zur Topologieoptimierung. In [224] beschreibt Schönlein ein Lösungsverfahren zur Planung und Optimierung eines Versorgungsnetzes, bei dem eine Menge Servicetechniker kostenoptimal eingesetzt werden soll. Topologieoptimierung spielt auch beim Entwurf von Energienetzen eine wichtige Rolle [179]. Der wohl bekannteste Anwendungsbereich ist jedoch die Topologieoptimierung von Kommunikationsnetzen. Einen umfassenden Überblick über Veröffentlichungen zum Thema Entwurf und Optimierung von Topologien geben Pióro und Medhi [225] sowie Boorstyn und Frank [226]. Typischerweise werden Probleme zur Topologieoptimierung von Kommunikationsnetzen wie folgt klassifiziert:

Netzdimensionierungsprobleme Diese werden in der Literatur auch (Übertragungs-) Kapazitätsprobleme (*engl. capacity problems*) genannt. Dabei müssen Netzstrukturen mit minimalen Übertragungskapazitäten/-raten bzw. minimalen Kosten unter Berücksichtigung von Dienstgüteanforderungen entworfen werden. Häufig sind die Kanten im Netz vorgegeben und es müssen Flüsse, die auch fraktional sein können, durch das Netz durch Ausnutzung vorhandener Übertragungsressourcen optimal geleitet werden [227]. Gerla und Kleinrock [228] stellen ein Lösungsverfahren zur Minimierung der Übertragungskosten unter Berücksichtigung von Verkehrsanforderungen, Leitungsauslastungen und der Ausfallsicherheit vor. Nach Zerlegung des Ausgangsproblems und Analyse der Teilprobleme schlagen die Autoren eine Heuristik zur Lösung vor.

Zu den Netzdimensionierungsproblemen gehören auch sogenannte *Constrained Minimal Spanning Tree*- oder OCST-Probleme. Rothlauf [220] präsentiert mehrere Heuristiken zur Lösung des OCST Problems und vergleicht diese miteinander. Weitere Lösungsverfahren für dieses Problem werden in [219, 229, 230] behandelt. Beim *Fixed-Charged Network Design Problem* werden lineare Flusskosten und feste Installationskosten für das Installieren einer Kante angenommen [231, 232]. Ein verwandtes Problem ist das *Network Loading Problem* [233]. Bei diesem wird eine stufenweise Funktion für die Installationskosten angenommen. Zur Bereit-

stellung der benötigten Transportleistung können ganzzahlige Vielfache von *Grundbaustein-Kanten* [227] mit bestimmten Übertragungsraten installiert werden. Zu jeder dieser Basisübertragungsraten sind dann Installationskosten einer Einheit dieses Grundbausteins (Basisleitung) gegeben.

Positionierungsprobleme Bei dieser Problemklasse müssen Netzressourcen optimal positioniert werden. Dabei kann eine endliche Menge möglicher Positionen (*engl. candidate points*) vorgegeben sein oder die Ressourcen können beliebig im Raum positioniert werden. Es wird daher zwischen (i) diskreten Problemen und (ii) kontinuierlichen Problemen unterschieden [224]. Bei Letzterem ist der gültige Lösungsraum quasi nahezu unendlich. In der Literatur existieren mehrere Arbeiten zum Lösen von Positionierungsproblemen. Beispielsweise präsentiert Robertazzi [234] ein Verfahren zur optimalen Positionierung eines Konzentrators an einer gegebenen endlichen Menge von Punkten. Gupta und Ross [235] beschreiben ein Modell zur optimalen Positionierung von Signalverstärkern und Koppелеlementen in einem lokalen Netz unter Berücksichtigung mittlerer Ende-zu-Ende Verzögerungen. Auch Pióro und Medhi [225] beschreiben ausführlich Lösungsansätze zur optimalen Knotenpositionierung.

Kombination aus Netzdimensionierungs- und Positionierungsproblemen Die bekannteste Kombination ist das *Hub Location Problem*. Dieses Problem entstammt aus der Planung logistischer Systeme. Bei diesem Problem müssen zentral gelegene Verkehrsknoten, sogenannte Umschlagpunkte (*engl. hubs*), unter Berücksichtigung von Güterflüssen zwischen den Knoten optimal positioniert werden. Es handelt sich um ein zusammengesetztes Problem, da zum einen die zentralen Verkehrsknoten optimal positioniert und zum anderen die optimalen Güterflüsse bestimmt werden müssen. Dabei kann ein Fluss über ein oder mehrere Hubs geleitet werden. Häufig wird die Anzahl einzurichtender Hubs entweder über Fixkosten gesteuert oder von vornherein vorgegeben (*p-Hub Median Problem*) [236].

Bei der Topologieoptimierung können die Vermittlungsknoten als zentral gelegene Verkehrsknoten, an denen Verkehr aggregiert und aufgeteilt wird, die Güterflüsse als Verkehrsanforderungen und die Transportwege als Leitungen angesehen werden. Aufgrund der Bedeutung des Problems in unterschiedlichen Bereichen wurden in den letzten Jahren unterschiedliche Lösungsverfahren für Probleme mit bestimmten Randbedingungen (zum Beispiel maximale Anzahl zu überquerender Verkehrsknoten) entwickelt. Campbell et al. [237] geben eine umfassende Übersicht über das Problem sowie über existierende Formulierungs- und Lösungsansätze. Pióro und Medhi [225] präsentieren sowohl exakte als auch heuristische Lösungsverfahren für das, wie sie es nennen, *Optimal Node Location and Link Connectivity Problem*. Dieses Problem entspricht im Wesentlichen dem Hub Location Problem. Campbell et al. [237] diskutieren die begrenzten Möglichkeiten beim Lösen von Hub Location Problemen und erwähnen, dass die besten Verfahren derzeit nur für *kleine* Probleminstanzen mit maximal 50 Knoten und mit teilweise erheblichen Einschränkungen anwendbar sind. Aus diesem Grund werden für praxisrelevante, große Probleminstanzen bevorzugt heuristische Lösungsverfahren angewandt.

Zuordnungsprobleme Bei dieser Problemklasse sollen optimale, paarweise Relationen zwischen zwei Ressourcen oder Aktivitäten hergestellt werden, die zusätzliche Bedingungen erfüllen. Diese sind ähnlich zu den Problemen der Standort- und Transportplanung. Aus diesem Grund können Zuordnungsprobleme mit Verfahren zur Optimierung von Transportplanungsproblemen gelöst werden [160]. Beispielsweise muss beim *Terminal Assignment Problem* bzw. *Cell Assignment Problem* eine gegebene Menge von Terminals bzw. Zellen einer gegebenen

Menge von Ressourcen (Vermittlungsknoten) optimal zugeordnet werden. Die Ressourcen befinden sich an vorgegebenen Positionen. Häufig ist die Anzahl der Terminals bzw. Zellen, die einer Ressource zugeordnet werden kann, begrenzt [234, 238].

Aus den in Abschnitt 4.1.2 beschriebenen Herausforderungen folgt, dass das Optimieren von Topologien eingebetteter Kommunikationsnetze eine Kombination bestehend aus den folgenden (Teil-) Problemen ist:

- Netzdimensionierungsproblem: Bestimmen einer kostenoptimalen Vernetzungsstruktur zwischen den Vermittlungsknoten.
- Positionierungsproblem: Bestimmen der kostenoptimalen Positionen der Vermittlungsknoten und Bündelungspunkte.
- Zuordnungsproblem: Bestimmen der kostenoptimalen Anbindung der Endknoten an die Vermittlungsknoten und Bündelungspunkte.

Obwohl eine Vielzahl verwandter Arbeiten existiert, berücksichtigt keine die Eigenschaften und Anforderungen, die genau denen in eingebetteten Kommunikationsnetzen entsprechen. Zudem existieren häufig Lösungsverfahren für lediglich *kleine* Probleminstanzen. Folglich können bekannte Verfahren nur zur Lösung eingeschränkter und/oder kleiner Probleminstanzen angewandt oder zur Inspiration bei der Entwicklung neuer, problemspezifischer Verfahren herangezogen werden. Häufig werden vereinfachte Distanzmetriken angenommen. Allerdings ist die Kostenfunktion für eine Kante bzw. einen Leitungskanal in eingebetteten Kommunikationsnetzen komplexer (siehe Gleichung (4.3)). Aus diesem Grund können aus der Literatur bekannte Verfahren zur Lösung des OCST Problems nicht direkt angewandt werden, sondern müssen entsprechend angepasst werden. Häufig ist die Anzahl zu positionierender Knoten gegeben. Hingegen muss bei eingebetteten Kommunikationsnetzen die optimale Anzahl notwendiger Bündelungspunkte bestimmt und diese optimal positioniert werden.

Falls die Anzahl der Vermittlungsknoten nicht eingeschränkt und keine Ausfallsicherheit gefordert ist sowie Verkehrsanforderungen und der Bündelungseffekt vernachlässigt werden können, kann das Problem auf das Bestimmen eines Steinerbaumes abgebildet werden. Bei gleichen Einschränkungen und integrierten Vermittlungsknoten kann das Problem auf das Bestimmen eines MST abgebildet werden. In diesem Fall entsprechen die Verzweigungspunkte der Spannbäume den Positionen der Vermittlungsknoten. Sowohl für das Bestimmen eines Steinerbaumes als auch eines MST sind effiziente Verfahren bekannt [154–157, 159].

Im Folgenden werden neue, anwendungsreife und leistungsfähige Verfahren zur Kostenoptimierung von Topologien eingebetteter Kommunikationsnetze entwickelt. Neben den Verfahren zur Positionsoptimierung und zur Erstellung einer optimalen Vernetzungsstruktur wird auch ein Verfahren zum Bestimmen einer Kompromisslösung entwickelt. Letzteres löst den Zielkonflikt zwischen Minimierung der Anzahl der Vermittlungsknoten und Minimierung der Leitungskosten.

Gelegentlich sind in der praktischen Anwendung die Freiheitsgrade eingeschränkt. Die folgenden Lösungsverfahren sind auch für derartig eingeschränkte Probleme anwendbar. Beispielsweise können die Positionen der Vermittlungsknoten und/oder Bündelungspunkte gege-

ben sein. Dennoch kann mit den in Abschnitt 5.3 entwickelten Verfahren eine kostengünstige Vernetzungsstruktur bestimmt werden. Umgekehrt können die Verläufe der Leitungskanäle gegeben sein und es müssen die optimalen Positionen der Vermittlungsknoten bestimmt werden. Die Verfahren sind also modular und können beliebig miteinander kombiniert werden.

5.2 Positionierungsverfahren

In diesem Abschnitt werden exakte und heuristische Lösungsverfahren zur kostengünstigen Positionierung von Vermittlungsknoten und Bündelungspunkten entwickelt. Das exakte Lösungsverfahren ist jedoch nicht in der Lage, alle relevanten Eigenschaften eingebetteter Kommunikationsnetze zu berücksichtigen, findet dafür aber für eingeschränkte Probleminstanzen optimale Lösungen. Hingegen können die heuristischen Lösungsverfahren alle relevanten Eigenschaften berücksichtigen und finden auch für uneingeschränkte Probleminstanzen akzeptable Lösungen. Allerdings kann bei den heuristischen Lösungsverfahren keine Aussage über die Qualität einer gefundenen Lösung gemessen an der optimalen Lösung gemacht werden.

5.2.1 Exaktes Lösungsverfahren

Kombinatorische Optimierungsprobleme lassen sich häufig mathematisch als lineare Optimierungsprobleme formulieren und können mit geeigneten Verfahren gelöst werden (siehe Abschnitt 3.2.2). Allerdings können bei komplexen Problemen aus unterschiedlichen Gründen nicht alle Eigenschaften und Nebenbedingungen berücksichtigt werden. Zum einen können nicht alle Nebenbedingungen als lineare Gleichung oder Ungleichung formuliert werden. Beispielsweise enthält die in Abschnitt 4.3 hergeleitete Zielfunktion (4.11) nichtlineare Terme. Zum anderen ist die Zahl der Variablen und folglich des Lösungsraumes so groß, dass bekannte Lösungsverfahren keine Ergebnisse in akzeptabler Zeit finden.

Durch Vernachlässigung gewisser Eigenschaften und Nebenbedingungen kann das Positionierungsproblem der Vermittlungsknoten reduziert und als MILP formuliert werden. Eine solche Formulierung wird vom Autor et al. in [3, 5] vorgestellt. Gegeben ist lediglich eine endliche Menge invarianter Endknoten $N = \{n_1, n_2, \dots\}$ an den Positionen $\{(x_n, y_n) \mid n \in N\}$ und eine endliche Menge beliebig positionierbarer Vermittlungsknoten $S = \{s_1, s_2, \dots\}$. Mit der folgenden MILP Formulierung können die Leitungskosten minimiert werden. Jedoch werden die Verkehrsanforderungen und der Bündelungseffekt vernachlässigt. Die Endknoten werden baumförmig miteinander vernetzt. Es ist also jeder Endknoten direkt mit einem Vermittlungsknoten verbunden und es befindet sich in jedem Leitungskanal exakt eine vollständige physikalische Leitung. Folglich gilt:

$$\Psi = \Delta \quad \text{und} \quad \Upsilon = \Phi. \quad (5.1)$$

Weiter werden auch positionsabhängige Kosten vernachlässigt. Zur Vereinfachung wird in dieser Formulierung für einen Endknoten $n_i \in N$ oder Vermittlungsknoten $s_i \in S$ einfach nur dessen Index der Indexmenge $\{1, 2, \dots, |N|\}$ bzw. $\{1, 2, \dots, |S|\}$ verwendet; also Endknoten i anstatt n_i . Es wird die Manhattan Distanz $d_{i,j}$ als Distanzmetrik zwischen den Knoten i und j an

den Positionen (x_i, y_i) und (x_j, y_j) verwendet. Mit diesen Annahmen kann die Zielfunktion aus Gleichung (4.11) wie folgt vereinfacht werden:

$$z(\Delta, \Phi) = \sum_{i=1}^{|N|} \sum_{j=1}^{|S|} \delta_{i,j} \cdot d_{i,j} + \sum_{i=1}^{|S|-1} \sum_{j=i+1}^{|S|} \phi_{i,j} \cdot d_{i,j} \quad (5.2)$$

Die Distanz $d_{i,j}^x$ in x -Richtung zwischen den Knoten i und j ist definiert als:

$$d_{i,j}^x = |x_i - x_j| = \max \{ (x_i - x_j), (x_j - x_i) \} \quad (5.3)$$

Diese Gleichung kann in Form von zwei Ungleichungen formuliert werden:

$$d_{i,j}^x \geq (x_i - x_j) \quad (5.4a)$$

$$d_{i,j}^x \geq (x_j - x_i) \quad (5.4b)$$

Eine zulässige Lösung eines Optimierungsproblems muss alle Nebenbedingungen erfüllen (siehe Abschnitt 3.2.1). Folglich muss die Variable $d_{i,j}^x$ größer gleich dem Maximalwert der beiden Differenzen $(x_i - x_j)$ und $(x_j - x_i)$ sein. Allgemein führt ein größerer Wert von $d_{i,j}^x$ zu höheren Kosten. Da die Zielfunktion (5.2) minimiert wird, besitzt die Variable $d_{i,j}^x$ letztendlich genau den Maximalwert der beiden Differenzen.

Analog zu den Gleichungen (5.3) und (5.4) kann die Distanz $d_{i,j}^y$ in y -Richtung zwischen den Knoten i und j berechnet werden.

Für das Produkt $\delta_{i,j} \cdot d_{i,j}$ aus der vereinfachten Zielfunktion (5.2) zur Berechnung der Leitungskosten für die Anbindung der Endknoten gilt: $\forall i = 1, \dots, |N|, \forall j = 1, \dots, |S|$

$$\delta_{i,j} \cdot d_{i,j} = \min \{ d_{i,j}^x + d_{i,j}^y, \mathfrak{N} \cdot \delta_{i,j} \}, \quad (5.5)$$

mit dem problemspezifischen Parameter $\mathfrak{N} \gg 1$, der sicherstellt, dass bei Existenz einer Leitung zwischen den Knoten i und j die Distanz (als Kosten) zwischen diesen Knoten berücksichtigt wird. Falls keine Leitung zwischen den Knoten existiert, ist das Produkt $\mathfrak{N} \cdot \delta_{i,j} = 0$ und somit auch das Minimum.

Die Gleichung (5.5) kann in Form von drei Ungleichungen formuliert werden:

$$\delta_{i,j} \cdot d_{i,j} \leq d_{i,j}^x + d_{i,j}^y \quad (5.6a)$$

$$\delta_{i,j} \cdot d_{i,j} \leq \mathfrak{N} \cdot \delta_{i,j} \quad (5.6b)$$

$$\delta_{i,j} \cdot d_{i,j} \geq d_{i,j}^x + d_{i,j}^y + \mathfrak{N} \cdot (\delta_{i,j} - 1) \quad (5.6c)$$

Das Gleiche gilt für das Produkt $\phi_{i,j} \cdot d_{i,j}$ zur Berechnung der Leitungskosten zur baumförmigen Vernetzung der Vermittlungsknoten ($\forall i = 1, \dots, |S| - 1, \forall j = i + 1, \dots, |S|$).

Da die Endknoten nicht redundant angebunden werden, existiert zwischen jedem End- und jedem Vermittlungsknoten exakt eine Leitung und es gilt die Bedingung in Gleichung (4.6).

Mit dem Prinzip der Flusserhaltung (*engl. flow conservation*) [225, 239] kann eine baumförmige Vernetzungsstruktur zwischen den Vermittlungsknoten sichergestellt werden. Die Flusserhaltung besagt, dass Verkehr, der in ein Teilnetz hinein fließt, auch wieder aus diesem Teilnetz heraus fließen muss, es sei denn, es befindet sich in diesem Teilnetz die Senke. An dieser Stelle ist eine baumförmige Vernetzungsstruktur gefordert. Folglich muss lediglich sichergestellt sein, dass jeder Vermittlungsknoten mit einem bestimmten Vermittlungsknoten im Netz kommunizieren kann. Ist dies der Fall, können aufgrund der baumförmigen Vernetzungsstruktur alle Vermittlungsknoten miteinander kommunizieren. In folgender Bedingung wird angenommen, dass jeder Vermittlungsknoten mit dem Vermittlungsknoten 1 kommunizieren können muss: $\forall i = 2, \dots, |S|, \forall j, k = 1, \dots, |S|$

$$\sum_{\substack{j=1 \\ j \neq k}}^{|S|} \xi_{k,j}^{1,i} - \sum_{\substack{j=1 \\ j \neq k}}^{|S|} \xi_{j,k}^{1,i} = \begin{cases} 0 & \text{falls } k \neq 1 \text{ und } k \neq i, \\ +1 & \text{falls } k = 1, \\ -1 & \text{falls } k = i, \end{cases} \quad (5.7)$$

wobei die binäre Variable $\xi_{j,k}^{1,i}$ kennzeichnet, ob eine Leitung zwischen dem Vermittlungsknoten j und dem Vermittlungsknoten k zur Kommunikation zwischen dem Vermittlungsknoten 1 und dem Vermittlungsknoten i benötigt wird:

$$\xi_{j,k}^{1,i} = \begin{cases} 1 & \text{falls eine Leitung zwischen Vermittlungsknoten } j \text{ und } k \text{ zur Kommunikation} \\ & \text{zwischen Vermittlungsknoten 1 und } i \text{ benötigt wird,} \\ 0 & \text{sonst.} \end{cases} \quad (5.8)$$

Falls eine solche Leitung benötigt wird, muss durch folgende Bedingung sichergestellt sein, dass diese überhaupt existiert: $\forall i = 2, \dots, |S|, \forall j, k = 1, \dots, |S|$

$$\phi_{j,k} \geq \xi_{j,k}^{1,i} \quad (5.9)$$

Die Symmetrie der Matrix Φ mit ihren Elementen $\phi_{j,k}$ wurde bereits in Gleichung (4.10) erwähnt.

Unter Berücksichtigung dieser Nebenbedingungen kann die oben genannte Zielfunktion mit einem geeigneten Verfahren minimiert werden.

In [5] veröffentlichten der Autor et al. auch eine MILP Formulierung zur optimalen Positionierung integrierter Vermittlungsknoten. Diese Vermittlungsknoten dürfen nur dort positioniert werden, wo sich bereits Endknoten befinden. Daher ist die Anzahl möglicher Positionen und folglich der Lösungsraum stark eingeschränkt. Das MILP kann für diesen eingeschränkten Fall auch die positionsabhängigen Kosten berücksichtigen, indem die Kosten für eine Leitung zwischen den beiden Endknoten i und j in einer reduzierten $|N| \times |N|$ Kostenmatrix gespeichert werden. Die Zielfunktion aus (4.11) bleibt unverändert.

Zur Lösung dieses Optimierungsproblems wird zusätzlich die binäre Variable o_i ($i = 1, \dots, |N|$) eingeführt. Diese kennzeichnet, ob im Endknoten i ein Vermittlungsknoten integriert ist:

$$o_i = \begin{cases} 1 & \text{falls ein Vermittlungsknoten im Endknoten } i \text{ integriert ist,} \\ 0 & \text{sonst.} \end{cases} \quad (5.10)$$

Zudem hat die binäre Variable $\phi_{i,j}$ aus Gleichung (4.9) eine abgeänderte Bedeutung. Die Variable kennzeichnet nun, ob ein Vermittlungsknoten integriert im Endknoten i mit einem Vermittlungsknoten integriert im Endknoten j verbunden ist.

Bei einer baumförmigen Vernetzungsstruktur wird jeder Endknoten mit exakt einem Vermittlungsknoten verbunden (siehe Bedingung in Gleichung (4.6)). Da die Vermittlungsknoten in den Endknoten integriert sind, muss jeder Endknoten i mit exakt einem Endknoten j , in dem ein Vermittlungsknoten integriert ist, verbunden werden: $\forall i, j = 1, \dots, |N|$

$$\psi_{i,j} \leq o_j \quad (5.11)$$

Die binäre Variable $\xi_{k,l}^{i,j}$ ($i = 1, \dots, |N| - 1, j = i + 1, \dots, |N|, k, l = 1, \dots, |N|$) kennzeichnet nun, ob eine Leitung zwischen dem Endknoten k und l zur Kommunikation zwischen einem Vermittlungsknoten, der im Endknoten i integriert ist, und einem Vermittlungsknoten, der im Endknoten j integriert ist, benötigt wird.

Zwar muss auch diesmal die exakte Position der Vermittlungsknoten und die Vernetzungsstruktur zwischen diesen bestimmt werden, jedoch ist, wie bereits erwähnt, die Menge der möglichen Positionen stark eingeschränkt. Das bereits erwähnte Prinzip der Flusserhaltung wird modifiziert. Es wird angenommen, dass eine Kommunikation zwischen zwei Endknoten stattfindet, falls in zwei Endknoten Vermittlungsknoten integriert sind: $\forall i = 1, \dots, |N| - 1, \forall j = i + 1, \dots, |N|, \forall k = 1, \dots, |N|$

$$\sum_{\substack{l=1 \\ l \neq k}}^{|N|} \xi_{k,l}^{i,j} - \sum_{\substack{l=1 \\ l \neq k}}^{|N|} \xi_{l,k}^{i,j} = \begin{cases} = 0 & \text{falls } k \neq i \text{ und } k \neq j, \\ \geq o_i + o_j - 1 & \text{falls } k = i, \\ \leq -o_i - o_j + 1 & \text{falls } k = j. \end{cases} \quad (5.12)$$

Falls eine solche Leitung benötigt wird, muss wiederum durch folgende Bedingung sichergestellt sein, dass diese überhaupt existiert: $\forall i = 1, \dots, |N| - 1, \forall j = i + 1, \dots, |N|, \forall k, l = 1, \dots, |N|$

$$\phi_{k,l} \geq \xi_{k,l}^{i,j} \quad (5.13)$$

Es muss auch sichergestellt sein, dass, falls eine Leitung zwischen dem Vermittlungsknoten im Endknoten k und dem Vermittlungsknoten im Endknoten l benötigt wird, auch in beiden Endknoten ein Vermittlungsknoten integriert ist: $\forall k, l = 1, \dots, |N|$

$$\phi_{k,l} \leq o_k \quad \text{bzw.} \quad \phi_{k,l} \leq o_l \quad (5.14)$$

Des Weiteren ist, wie bereits erwähnt, die Anzahl zu installierender Vermittlungsknoten $|S|$ vorgegeben. Durch folgende Bedingung wird sichergestellt, dass nicht mehr Vermittlungsknoten installiert werden als zur Verfügung stehen:

$$\sum_{i=1}^{|N|} o_i \leq |S| \quad (5.15)$$

Unter Berücksichtigung dieser Modifikationen und Erweiterungen kann die oben genannte Zielfunktion nun ebenfalls mit einem geeigneten Verfahren minimiert werden.

Beide MILP Formulierungen (beliebig positionierbare und integrierte Vermittlungsknoten) sind in ihrer Struktur ähnlich zueinander. Wie bereits erwähnt haben sie den Nachteil, dass sie nur für eingeschränkte Probleminstanzen anwendbar sind. Jedoch können sie zur Bewertung heuristischer Lösungsverfahren herangezogen werden.

5.2.2 Heuristische Lösungsverfahren

In diesem Abschnitt werden problemangepasste Metaheuristiken zur kostengünstigen Positionierung von Vermittlungsknoten und Bündelungspunkten unter Berücksichtigung aller Nebenbedingungen entwickelt. Das erste Verfahren basiert auf Simulated Annealing (SA) und das zweite auf Tabu Search (TS). Für das SA-basierte Verfahren werden zusätzliche Schritte zur Erzeugung einer zulässigen Nachbarschaftslösung beschrieben. Mit diesen heuristischen Lösungsverfahren (Heuristiken) können nun im Gegensatz zum exakten Lösungsverfahren sämtliche Eigenschaften und Anforderungen eingebetteter Kommunikationsnetze berücksichtigt werden.

5.2.2.1 Verfahren basierend auf Simulated Annealing

Die Idee und das Prinzip des SA wurden bereits in Abschnitt 3.2.3.1 beschrieben. In Algorithmus 5.1 ist nun ein SA-basiertes, problemangepasstes Verfahren zur kostengünstigen Positionierung von Vermittlungsknoten und Bündelungspunkten beschrieben. Durch ein geeignetes Verfahren zur Erzeugung benachbarter Lösungen wird zudem gleichzeitig, d. h. innerhalb des Verfahrens, eine kostengünstige Anzahl von Bündelungspunkten bestimmt.

Gegeben sind die endliche Menge der invarianten Endknoten $N = \{n_1, n_2, \dots\}$, die endliche Menge der Vermittlungsknoten $S = \{s_1, s_2, \dots\}$, die Matrix \mathbf{R} mit den Verkehrsanforderungen und die Matrix $\mathbf{\Gamma}$ mit den Umgebungskosten. Zuerst werden die allgemeinen Parameter für den Cooling Schedule initialisiert. Danach wird mit einem Eröffnungsverfahren eine zulässige initiale Lösung l' erzeugt (Zeile 1). Zu Beginn des Algorithmus 5.1 ist die aktuell beste Lösung l^* gleich der initialen Lösung. Die Abbruchkriterien sind in den Zeilen 3 und 5 beschrieben. In Zeile 6 wird durch Aufruf der Funktion ERZEUGENACHBARSCHAFTSLÖSUNG eine zulässige Nachbarschaftslösung erzeugt. Das Verfahren zur Erzeugung einer zulässigen Nachbarschaftslösung wird im nächsten Abschnitt detailliert beschrieben.

Nachdem eine zulässige benachbarte Lösung l'' erzeugt wurde, wird diese bewertet. Falls eine Verbesserung (geringere Leitungskosten) erreicht wird, wird diese als aktuelle Lösung l' akzeptiert (Zeile 10). Falls die Lösung l'' besser als die aktuell beste Lösung l^* ist, wird diese als die aktuell beste Lösung akzeptiert (Zeile 13). Bei Gleichwertigkeit wird die Nachbarschaftslösung l'' mit einer Wahrscheinlichkeit von 50% akzeptiert (Zeile 17). Falls die Nachbarschaftslösung l'' schlechter als die aktuelle Lösung l' ist, entscheidet die Akzeptanzwahrscheinlichkeit

$$P_A = \exp\left(-\frac{z(l'') - z(l')}{z(l') \tau}\right) \quad (5.16)$$

Algorithmus 5.1 SA-basiertes Positionierungsverfahren**Input:** N, S, R, Γ **Output:** l^*

▷ Zulässige kostengünstige Lösung

define

mIter ▷ Maximale Anzahl aufeinander folgender Iterationen ohne Verbesserung

mImpr ▷ Maximale Anzahl von Verbesserungen (*engl. improvements*)mAtmp ▷ Maximale Anzahl von Versuchen (*engl. attempts*) τ und p ▷ (Anfangs-) Temperatur und Abkühlungsgröße1: $l^* \leftarrow l' \leftarrow \text{ERÖFFNUNGSVERFAHREN}(N, S, R, \Gamma)$ ▷ Erzeuge zulässige initiale Lösung2: iter \leftarrow 03: **while** iter < mIter **do**4: impr \leftarrow 0; atmp \leftarrow 05: **while** (impr < mImpr) \wedge (atmp < mAtmp) **do**6: $l'' \leftarrow \text{ERZEUGENACHBARSCHAFTSLÖSUNG}(l')$

7: atmp++

8: Erzeuge Zufallszahl $r \in [0,1]$ 9: **if** $z(l'') < z(l')$ **then** ▷ Verbesserte Lösung10: $l' \leftarrow l''$

11: impr++

12: **if** $z(l'') < z(l^*)$ **then**13: $l^* \leftarrow l''$ 14: iter \leftarrow 015: **else if** $z(l'') = z(l')$ **then** ▷ Gleichwertige Lösung16: **if** $r < 0,5$ **then**17: $l' \leftarrow l''$ 18: **else if** $r < \exp\left(-\frac{z(l'')-z(l')}{z(l') \tau}\right)$ **then** ▷ Schlechtere Lösung19: $l' \leftarrow l''$ 20: **if** atmp = mAtmp **then** ▷ Ausreichende Anzahl von Versuchen

21: iter++

22: $\tau \leftarrow p \cdot \tau$ 23: **return** l^*

über die Akzeptanz oder Ablehnung der Nachbarschaftslösung l'' , wobei der Nenner $z(l')$ τ dem Steuerparameter Ξ entspricht. Wie bereits in Abschnitt 3.2.3.1 erwähnt, wird die Nachbarschaftslösung akzeptiert, wenn die Wahrscheinlichkeit P_A größer als eine gleichförmig verteilte Zufallszahl $r \in [0,1]$ ist.

Neben dem SA-basierten Positionierungsverfahren wurde vom Autor auch ein Toleranzschwellwert- und Sintflut-basiertes Verfahren entwickelt und in der im Anhang A beschriebenen Softwarebibliothek implementiert. Beim Toleranzschwellwert-basierten Verfahren wird der initiale Schwellwert τ abhängig von den Kosten einer initialen Lösung gesetzt. Der Schwellwert wird dann im Verlaufe des Optimierungsprozesses weiter verringert, indem er nach einer bestimmten Anzahl von Iterationen mit einem zuvor festgelegten Wert (sinnvollerweise < 1) multipliziert wird. Beim Sintflut-basierten Verfahren wird initial der Wasserstand auf die Kosten einer initialen Lösung gesetzt und bei jeder Verbesserung verringert. In [12] sind beide Verfahren dokumentiert.

Algorithmus 5.2 Eröffnungsverfahren für SA-basiertes Positionierungsverfahren**Input:** N, S, R, Γ **Output:** l **ensure** $|N| \geq |S|$ **define** **init** **▷** Anzahl initialer Bündelungspunkte1: **for all** $s \in S$ **do**2: **switch** Positionierungsstrategie Vermittlungsknoten **do**3: **case** Beliebig positionierbar4: Positioniere s an zulässiger, zufällig ausgewählter Position (x,y) 5: **case** Integriert6: Wähle zufälligen Endknoten $n_i \in \{n_j \in N \mid o_j = 0\}$ 7: Positioniere s an der Position (x_{n_i}, y_{n_i}) 8: $o_i \leftarrow 1$ 9: $B \leftarrow \emptyset$ **▷** Erzeuge leere Menge von Bündelungspunkten10: **for** $i \leftarrow 1$ **to** init **do**11: Erzeuge Bündelungspunkt b 12: Positioniere b an zulässiger, zufällig ausgewählter Position (x,y) 13: $B \leftarrow B \cup \{b\}$ 14: $l \leftarrow \text{ERSTELLEVERNETZUNGSSTRUKTUR}(N, S, B, R, \Gamma)$ 15: **return** l **▷** Zulässige Lösung**Verfahren zur Erzeugung zulässiger Lösungen**

Bei einem Nachbarschaftssuchverfahren muss ein problemspezifisches Verfahren zur Erzeugung einer zulässigen initialen Lösung (Eröffnungsverfahren) sowie ein Verfahren zur Erzeugung einer zulässigen Nachbarschaftslösung entwickelt werden. Für die Positionierung der Vermittlungsknoten existieren zwei Strategien (siehe Abschnitt 4.1.2): (i) Beliebig positionierbare Vermittlungsknoten und (ii) in den Endknoten integrierte Vermittlungsknoten. Diese Strategien müssen bei den Verfahren zur Erzeugung zulässiger Lösungen entsprechend berücksichtigt werden.

Eröffnungsverfahren Algorithmus 5.2 beschreibt ein geeignetes Eröffnungsverfahren. In Abhängigkeit von der Positionierungsstrategie werden die Vermittlungsknoten zufällig positioniert (Zeile 1 ff). Bei integrierten Vermittlungsknoten wird sichergestellt, dass in einem Endknoten maximal ein Vermittlungsknoten integriert wird. Deshalb muss die Anzahl der Endknoten größer oder gleich der Anzahl der Vermittlungsknoten sein.

Anschließend werden die Bündelungspunkte erzeugt und ebenfalls zufällig positioniert (Zeile 9 ff). Die Anzahl initial zu positionierender Bündelungspunkte muss anfangs definiert werden. Als letztes werden die Endknoten, Vermittlungsknoten und Bündelungspunkte kostengünstig miteinander vernetzt (Zeile 14). Verfahren zur kostengünstigen Vernetzung werden später in Abschnitt 5.3 entwickelt.

Verfahren zur Erzeugung einer Nachbarschaftslösung Algorithmus 5.3 beschreibt ein Verfahren zur Erzeugung einer zulässigen Nachbarschaftslösung. Ausgangspunkt ist eine zulässige

Algorithmus 5.3 Verfahren zur Erzeugung einer Nachbarschaftslösung für das SA-basierte Positionierungsverfahren

Input: l ▷ Zulässige Lösung
Output: l' ▷ Zulässige zu l benachbarte Lösung
▷ Kopiere Ausgangslösung

- 1: $l' \leftarrow \text{KOPIERE}(l)$
- 2: Erzeuge Zufallszahl $r \in [0,1]$
- 3: **if** $r \geq \frac{|B|}{|B|+|S|}$ **then**
- 4: Wähle zufällig einen Vermittlungsknoten $s \in S(l')$ aus
- 5: **switch** Positionierungsstrategie Vermittlungsknoten **do**
- 6: **case** Beliebig positionierbar
- 7: Verschiebe s an zulässige, zufällig ausgewählte Position (x,y)
- 8: **case** Integriert
- 9: Verschiebe s auf zufällig ausgewählte Endknotenposition $\{(x_n, y_n) \mid n \in N\}$
- 10: **else**
- 11: Erzeuge Zufallszahl $r' \in [0,1]$
- 12: **if** $B(l') = \emptyset \vee r' > 0,9$ **then**
- 13: Erzeuge Bündelungspunkt b
- 14: Positioniere b an zulässiger, zufällig ausgewählter Position (x,y)
- 15: $B(l') \leftarrow B(l') \cup \{b\}$ ▷ Füge Bündelungspunkt hinzu
- 16: **else if** $r' > 0,8$ **then**
- 17: Wähle zufälligen Bündelungspunkt $b \in B(l')$
- 18: $B(l') \leftarrow B(l') \setminus \{b\}$ ▷ Lösche Bündelungspunkt
- 19: **else**
- 20: Wähle zufälligen Bündelungspunkt $b \in B(l')$
- 21: Verschiebe b an zulässige, zufällig ausgewählte Position (x,y)
- 22: $l' \leftarrow \text{ERSTELLEVERNETZUNGSSTRUKTUR}(N(l'), S(l'), B(l'), \mathbf{R}, \mathbf{\Gamma})$
- 23: **return** l'

Lösung l . Diese Lösung wird kopiert (Zeile 1). Beim Kopieren einer Lösung wird eine neue (Lösungs-) Instanz erstellt sowie alle *inneren* Elemente 1-zu-1 kopiert und entsprechend zugewiesen. Auf den Elementen der kopierten Lösung l' werden dann die Modifikationen durchgeführt und so eine Nachbarschaftslösung erzeugt. Die Menge $S(l')$ beschreibt die Menge der Vermittlungsknoten und $B(l')$ die Menge der Bündelungspunkte der Lösung l' . Nach dem Kopieren wird eine gleichförmig verteilte Zufallszahl $r \in [0,1]$ erzeugt. Im nächsten Schritt wird in Abhängigkeit von der Zufallszahl r und dem Verhältnis $\frac{|B|}{|B|+|S|}$ ein Vermittlungsknoten oder ein Bündelungspunkt bzw. die Menge der Bündelungspunkte verändert. Ist die Zufallszahl r größer oder gleich als $\frac{|B|}{|B|+|S|}$, so wird ein Vermittlungsknoten in Abhängigkeit von der Positionierungsstrategie verschoben. Diese Auswahlstrategie hat sich in mehreren Untersuchungen während der Entwicklung als geeignetste erwiesen. Es ist zulässig, dass ein Vermittlungsknoten an eine Endknotenposition verschoben wird, an der bereits ein Vermittlungsknoten existiert. Vermittlungsknoten, die unnötig sind, d. h. deren Entfernen die Leitungskosten nicht erhöhen, können in einem nachgelagerten Schritt entfernt werden (siehe Abschnitt 5.2.3).

Da Bündelungspunkte keine Kosten verursachen, ist deren maximale Anzahl nicht vorgegeben. Allerdings können, wie später gezeigt wird, zu viele Bündelungspunkte zu ungünstigen Lösungen führen. Zudem beeinflusst die Anzahl der Bündelungspunkte die Qualität und Leistungs-

fähigkeit des Verfahrens. Die optimale Anzahl der Bündelungspunkte muss also ebenfalls bestimmt werden. Deshalb sind unterschiedliche Operationen für die Bündelungspunkte definiert. Falls die Menge der Bündelungspunkte leer ist, wird ein neuer Bündelungspunkt hinzugefügt. Im anderen Fall wird mit einer Wahrscheinlichkeit von

- 10% ein neuer Bündelungspunkt hinzugefügt (Zeile 12 ff)
- 10% ein zufällig ausgewählter Bündelungspunkt gelöscht (Zeile 16 ff)
- 80% ein zufällig ausgewählter Bündelungspunkt an eine zulässige, zufällig ausgewählte Position verschoben (Zeile 20 ff)

Diese Zahlen haben sich ebenfalls in mehreren Untersuchungen während der Entwicklung als geeignetste erwiesen.

Als letzter Schritt werden die Endknoten, Vermittlungsknoten und Bündelungspunkte kostengünstig miteinander vernetzt und die zulässige, zu l benachbarte Lösung l' zurück gegeben.

Neben den bereits erwähnten Untersuchungen wurden weitere Verfahren zur Erzeugung von Nachbarschaftslösungen entwickelt und untersucht. Allerdings konnten durch deren Anwendung nur in Einzelfällen bessere Lösungen erzielt werden. Deshalb werden diese nur kurz zusammengefasst und im Folgenden nicht näher betrachtet:

- Ein zufällig ausgewählter, beliebig positionierbarer Vermittlungsknoten wird nur innerhalb eines definierten Rechtecks oder einer definierten Raute verschoben. Dabei kann die Größe des Rechtecks oder der Raute konstant sein oder sich im Verlauf des Optimierungsprozesses ändern. Der Mittelpunkt des Rechtecks oder der Raute entspricht der Position des zufällig ausgewählten Vermittlungsknotens.
- Anstatt nur eines Vermittlungsknotens wird eine definierte Anzahl zufällig ausgewählter Vermittlungsknoten an zufällig ausgewählte Positionen verschoben.
- Im Verlauf des Optimierungsprozesses wird die Wahrscheinlichkeit, ob ein Vermittlungsknoten oder Bündelungspunkt behandelt wird, über der Zeit so verändert, dass zu Beginn bevorzugt Vermittlungsknoten und gegen Ende bevorzugt Bündelungspunkte behandelt werden.
- Zufallsbasierte Auswahlstrategie, d. h. die Entscheidung, ob ein Vermittlungsknoten oder ein Bündelungspunkt verändert werden soll, erfolgt ohne Berücksichtigung des Verhältnisses $\frac{|B|}{|B|+|S|}$.
- Bei der Auswahl wird zusätzlich zur Anzahl der Vermittlungsknoten und Bündelungspunkte der Kostenfaktor λ berücksichtigt.
- Weitere Kombinationen für die Wahrscheinlichkeiten für das Verschieben, Entfernen und Hinzufügen von Bündelungspunkten werden in Betracht gezogen.
- Bei jeder Ausführung wird eine vorgegebene Anzahl von Bündelungspunkten zufällig positioniert und anschließend werden die unnötigen entfernt. Bei diesem Verfahren stellt sich die Frage, inwiefern die erzeugte Lösung noch als *benachbart* zur Ausgangslösung betrachtet werden kann.

- Neu erzeugte Bündelungspunkte werden intelligent positioniert, indem ein Endknoten zufällig ausgewählt, dessen beide Nachbarn über die minimale Distanz ermittelt und ein umschließendes Rechteck berechnet wird. Der hinzuzufügende Bündelungspunkt wird dann innerhalb dieses Rechtecks platziert. Dieses Verfahren entspricht einem *Clustering*-Ansatz und erzielt insbesondere bei einem niedrigen Kostenfaktor $\lambda \leq 0,7$ und einer heterogenen Kostenmatrix Γ gute Ergebnisse.
- Kombination(en) aus diesen Verfahren.

In [2] veröffentlichten der Autor et al. weitere SA-basierte Lösungsverfahren. Dabei werden das kostengünstige Positionieren der Vermittlungsknoten und das Optimieren der Bündelungspunkte in zwei Teilprobleme aufgeteilt und jeweils separierte Lösungsverfahren präsentiert. Für die Ausführungsreihenfolge der beiden Verfahren werden zwei Strategien diskutiert: Entweder eine Verschachtelung oder eine sequentielle Ausführung. Bei der Verschachtelung wird ein Verfahren zur Optimierung der Bündelungspunkte aufgerufen, nachdem ein Vermittlungsknoten verschoben wurde. Dies hat eine lange Ausführungsdauer zur Folge. Bei der sequentiellen Ausführung werden zuerst die Vermittlungsknoten kostengünstig positioniert und anschließend eine kostengünstige Anzahl von Bündelungspunkten bestimmt und diese positioniert.

5.2.2.2 Verfahren basierend auf Tabu Search

Das Prinzip des Tabu Search (TS) wurde bereits in Abschnitt 3.2.3.3 beschrieben. In Algorithmus 5.4 ist ein TS-basiertes, problemangepasstes Verfahren beschrieben. Im Gegensatz zum SA-basierten Verfahren ist dieses Verfahren deterministisch und arbeitet nicht zufallsbasiert; mit Ausnahme der initialen Positionierung der Vermittlungsknoten. Vor der Ausführung müssen die Steuerparameter $mAtmpS$, $mAtmpB$, $mIter$ und $tenure$ definiert werden. Anfangs werden dann zwei Tabulisten initialisiert (Zeile 1). In der Tabuliste T_S werden die Tabu-Positionen für die Vermittlungsknoten und in T_B die Tabu-Positionen für die Bündelungspunkte gespeichert. Ein Eintrag in der Tabuliste beinhaltet neben der Position auch die Verweildauer.

Gegeben sind wiederum die endliche Menge der invarianten Endknoten $N = \{n_1, n_2, \dots\}$, die endliche Menge der Vermittlungsknoten $S = \{s_1, s_2, \dots\}$ und die Matrizen \mathbf{R} und $\mathbf{\Gamma}$. Anfangs wird jeder Vermittlungsknoten bei einem zufällig ausgewählten Endknoten, an dem sich noch kein Vermittlungsknoten befindet, positioniert (Zeile 2 ff). Die binäre Variable o_i ($i = 1, \dots, |N|$) speichert, ob beim Endknoten mit dem Index i ein Vermittlungsknoten positioniert wurde. Diese Variable wurde in ähnlicher Weise in Gleichung (5.10) verwendet. Jeder Vermittlungsknoten wird zwar anfänglich bei einem Endknoten positioniert, ist jedoch nicht integriert und kann später verschoben werden. Im nächsten Schritt wird bei jedem Endknoten genau ein Bündelungspunkt positioniert (Zeilen 8 ff). Danach werden die Endknoten, Vermittlungsknoten und Bündelungspunkte kostengünstig miteinander vernetzt und als aktuelle Lösung l gespeichert. Wie bereits erwähnt werden geeignete Vernetzungsverfahren in Abschnitt 5.3 entwickelt.

Nun erfolgt eine intensive Suche (Intensivierung) nach verbesserten Vermittlungsknotenpositionen (Zeilen 14 ff). Dazu wird jeder Vermittlungsknoten, wie in Abbildung 5.1(a) dargestellt, ausgehend von seiner aktuellen Position, auf vier benachbarte (*north*, *east*, *south*, *west*) zulässige Positionen mit dem Abstand einer Längeneinheit verschoben. Für jede Verschiebung wird zuvor eine Kopie der aktuellen Lösung l erzeugt (Zeile 18). Das tatsächliche Verschieben

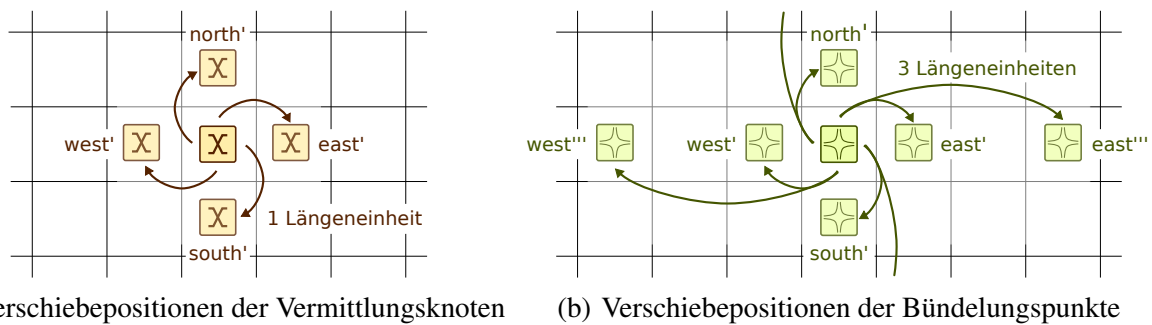


Abbildung 5.1: Intensivierung durch Verschieben der Vermittlungsknoten und Bündelungspunkte auf benachbarte und nahe gelegene Felder

des Vermittlungsknotens s_i erfolgt auf der Kopie l' . Somit bleibt die aktuelle Lösung l unverändert. Falls sich eine benachbarte Position außerhalb der definierten Ebene befindet (unzulässige Position), so wird der Vermittlungsknoten auf der Umrandung der Ebene positioniert. Zum Verschieben gehört auch das erneute Erstellen einer kostengünstigen Vernetzungsstruktur.

Falls die neue Position des Vermittlungsknotens $s_i \in S(l')$ nicht tabu ist oder durch die Verschiebung eine Verbesserung ($z(l') \leq z(l^*)$) gegenüber der aktuell besten Lösung l^* erzielt werden kann, wird die Lösung l' zur Menge zu betrachtender Lösungen L' hinzugefügt (Zeile 23). Eine Lösung kann also auch dann zur Menge L' hinzugefügt und ausgewertet werden, obwohl diese Tabu ist. Dies entspricht dem in Abschnitt 3.2.3.3 erwähnten Aspirationskriterium: Eine Lösung ist dann tabu, wenn die neue Position des verschobenen Vermittlungsknotens in der Tabuliste enthalten und die Verweildauer noch nicht abgelaufen ist.

Im nächsten Schritt wird die beste Lösung aus der Menge L' ausgewählt und der aktuellen Lösung l zugewiesen (Zeile 24). Unter Umständen kann durch die Verschiebungen keine Verbesserung erzielt werden. Um zu vermeiden, dass in diesem Fall keine Verschlechterung erfolgt, wurde zuvor auch die aktuelle Lösung l zur Menge zu betrachtender Lösungen L' hinzugefügt (Zeile 16). Nach der Auswahl der besten Lösung wird die (neue) Position des Vermittlungsknotens $s_i \in S(l)$ und die Verweildauer zur Tabuliste hinzugefügt (Zeile 26). Ist die aktuelle Lösung l besser als die aktuell beste Lösung l^* , so wird diese aktualisiert (Zeile 28).

Als nächstes erfolgt eine intensive Suche nach verbesserten Bündelungspunktpositionen (Zeilen 30 ff). Dazu wird jeder Bündelungspunkt verschoben. Das Vorgehen ist nahezu identisch zur intensiven Suche nach verbesserten Vermittlungsknotenpositionen. Es wird lediglich, wie in Abbildung 5.1(b) dargestellt, jeder Bündelungspunkt, ausgehend von seiner aktuellen Position, auf vier benachbarte ($north'$, $east'$, $south'$, $west'$) zulässige Positionen und zudem auf vier nahe gelegene ($north'''$, $east'''$, $south'''$, $west'''$) zulässige Positionen mit dem Abstand drei Längeneinheiten verschoben. Folglich werden acht benachbarte Lösungen erzeugt. Falls sich eine benachbarte Position außerhalb der definierten Ebene befindet, so wird der Bündelungspunkt wiederum auf der Umrandung der Ebene positioniert. Auch bei der Suche nach verbesserten Bündelungspunktpositionen wird die aktuelle Lösung l kopiert und das tatsächliche Verschieben erfolgt auf der Kopie l' . Erneut wird bei jeder Verschiebung die Vernetzungsstruktur neu erstellt. Nach der Intensivierung werden die Tabulisten T_S und T_B aktualisiert (Zeile 46), indem abgelaufene Einträge entfernt werden.

Algorithmus 5.4 TS-basiertes Positionierungsverfahren**Input:** N, S, Γ, R **Output:** l^*

▷ Zulässige kostengünstige Lösung

ensure $|N| \geq |S|$ **define**

mAtmpS ▷ Maximale Anzahl von Positionierungsversuchen für Vermittlungsknoten

mAtmpB ▷ Maximale Anzahl von Positionierungsversuchen für Bündelungspunkte

mIter

▷ Maximale Anzahl von Iterationen

tenure

▷ Dauer für einen Eintrag in der Tabuliste

1: Initialisiere Tabulisten $T_S \leftarrow \emptyset, T_B \leftarrow \emptyset$ 2: **for all** $s \in S$ **do**3: Wähle zufälligen Endknoten $n_i \in \{n_j \in N \mid o_j = 0\}$ 4: Positioniere s an der Endknotenposition (x_{n_i}, y_{n_i}) 5: $o_i \leftarrow 1$ 6: **for** cycle $\leftarrow 1$ **to** $|N|$ **do**7: $B \leftarrow \emptyset$

▷ Erzeuge leere Menge von Bündelungspunkten

8: **for all** $n \in N$ **do**9: Erzeuge Bündelungspunkt b und positioniere b an der Endknotenposition (x_n, y_n) 10: $B \leftarrow B \cup \{b\}$ 11: $l \leftarrow \text{ERSTELLEVERNETZUNGSSTRUKTUR}(N, S, B, R, \Gamma)$ 12: iter $\leftarrow 0$, sIter $\leftarrow 0$, bIter $\leftarrow 0$ 13: **while** (sIter + bIter) \leq mIter **do**14: **for** atmpS $\leftarrow 1$ **to** mAtmpS **do**

▷ Verbessere Vermittlungsknotenpositionen

15: **for** $i \leftarrow 1$ **to** $|S|$ **do**16: $L' \leftarrow \{l\}$ 17: **for all** $d \in \{\text{north}', \text{east}', \text{south}', \text{west}'\}$ **do**18: $l' \leftarrow \text{KOPIERE}(l)$ ▷ Kopiere aktuelle Lösung l 19: Verschiebe $s_i \in S(l')$ an Position d

20: sIter ++

21: $s' \leftarrow s_i \in S(l')$ 22: **if** ($\neg T_S.\text{ISTTABU}((x_{s'}, y_{s'}), \text{iter}) \vee (z(l') \leq z(l^*))$) **then**23: $L' \leftarrow L' \cup \{l'\}$ 24: $l \leftarrow \arg \min_{l'' \in L'} (z(l''))$

▷ Finde beste Lösung

25: $s' \leftarrow s_i \in S(l)$ 26: Füge $((x_{s'}, y_{s'}), \text{iter} + \text{tenure})$ zu T_S hinzu27: **if** $z(l) \leq z(l^*)$ **then**28: $l^* \leftarrow l$

29: iter++

30: **for** atmpB $\leftarrow 1$ **to** mAtmpB **do**

▷ Verbessere Bündelungspunktpositionen

31: **for** $i \leftarrow 1$ **to** $|B|$ **do**32: $L' \leftarrow \{l\}$ 33: **for all** $d \in \{\text{north}', \text{east}', \text{south}', \text{west}', \text{north}''', \text{east}''', \text{south}''', \text{west}'''\}$ **do**34: $l' \leftarrow \text{KOPIERE}(l)$ ▷ Kopiere aktuelle Lösung l 35: Verschiebe $b_i \in B(l')$ an Position d

36: bIter ++

Fortsetzung auf der nächsten Seite ...

... Fortsetzung der vorherigen Seite (Algorithmus 5.4)

```

37:            $b' \leftarrow b_i \in B(l')$ 
38:           if ( $\neg T_B.\text{ISTTABU}((x_{b'}, y_{b'}), \text{iter}) \vee (z(l') \leq z(l^*))$ ) then
39:                $L' \leftarrow L' \cup \{l'\}$ 
40:            $l \leftarrow \arg \min_{l'' \in L'} (z(l''))$  ▷ Finde beste Lösung
41:            $b' \leftarrow b_i \in B(l)$ 
42:           Füge  $((x_{b'}, y_{b'}), \text{iter} + \text{tenure})$  zu  $T_B$  hinzu
43:           if  $z(l) \leq z(l^*)$  then
44:                $l^* \leftarrow l$ 
45:           iter++
46:           Aktualisiere  $T_S$  und  $T_B$ 
47:            $S \leftarrow \text{DIVERSIFIZIEREVERMITTLUNGSKNOTENPOSITIONEN}(o_i, \dots, o_{|N|}, \text{cycle})$ 
48:           Lösche Einträge aus  $T_S$  und  $T_B$ 
49: return  $l^*$ 

```

Nach dem Aktualisieren der Tabulisten werden die initialen Positionen der Vermittlungsknoten aus den Zeilen 2 ff. verändert. Dadurch wird ein bereits intensiv untersuchter Bereich verlassen und die Suche in einen anderen Teil des Lösungsraumes gelenkt (Diversifizierung). Die Funktion `DIVERSIFIZIEREVERMITTLUNGSKNOTENPOSITIONEN` (Zeile 47) zur Diversifizierung verschiebt jeden Vermittlungsknoten $s \in S$ an die Position des Endknotens $((i + \text{cycle}) \bmod |N|)$, wobei i der Endknotenindex des anfänglichen Endknotens $n_i \in N$ von Vermittlungsknoten s und \bmod die Division mit Rest (Modulo) Operation sind. Da die Funktion `DIVERSIFIZIEREVERMITTLUNGSKNOTENPOSITIONEN` in Algorithmus 5.4 genau $|N|$ -mal aufgerufen wird, wird jeder Vermittlungsknoten genau einmal an jedem Endknoten positioniert und ausgehend davon mit der Intensivierung begonnen.

Im Vergleich zum SA-basierten Verfahren wird beim TS-basierten Verfahren nicht explizit zwischen beliebig positionierbaren und integrierten Vermittlungsknoten unterschieden. Dennoch ist das Verfahren bei Probleminstanzen mit integrierten Vermittlungsknoten anwendbar. Mit der Parametrisierung $\text{mAtmpS} = 0$ und dank der Diversifizierung können auch solche Probleminstanzen behandelt werden.

5.2.3 Ergänzende Verfahren

Ergänzungen zu den soeben entwickelten Positionierungsverfahren können teilweise eine zulässige Lösung weiter verbessern. Derartige Ergänzungen werden in diesem Abschnitt beschrieben. Im Wesentlichen basieren diese Ergänzungen auf Beobachtungen, die während der Entwicklung der einzelnen Verfahren und bei diversen Untersuchungen gemacht wurden.

Verfahren zum Entfernen von Vermittlungsknoten In einem Steinerbaum ist die maximale Anzahl der Steinerpunkte bzw. -knoten in einem Graphen mit n Knoten $n - 2$. Ähnlich hierzu existiert in dem vorliegenden Optimierungsproblem eine Obergrenze für die Anzahl sinnvoller Vermittlungsknoten. Oberhalb dieser können die Leitungskosten durch Installation zusätzlicher Vermittlungsknoten nicht weiter gesenkt werden (siehe Abbildung 4.1).

Nach Abschluss des Positionierungsverfahrens können unter Umständen in der gefundenen Lösung *irrelevante* Vermittlungsknoten entfernt werden. Ein Vermittlungsknoten wird dann als irrelevant bezeichnet, wenn durch das Entfernen desselben die Leitungskosten nicht erhöht werden und mehr als ein Vermittlungsknoten im Netz existiert. Ein Vermittlungsknoten kann genau dann entfernt werden, wenn dieser

- exakt an der Position eines anderen Vermittlungsknotens positioniert wurde, d. h. wenn für zwei beliebige Vermittlungsknoten mit dem Index i und j gilt: $(x_i, y_i) = (x_j, y_j) \wedge i \neq j$ bzw. $d_{i,j} = 0$,
- exakt mit einem Vermittlungsknoten verbunden ist, d. h. wenn für einen beliebigen Vermittlungsknoten mit dem Index i folgendes gilt:

$$\sum_{j=1}^{|S|} v_{i,j} = 1 \quad (5.17)$$

- oder exakt mit zwei Leitungen angebunden ist, d. h. wenn für einen beliebigen Vermittlungsknoten mit dem Index i folgendes gilt:

$$\sum_{j=1}^{|S|} \phi_{i,j} = 2 \quad (5.18)$$

Ein Vermittlungsknoten wird dadurch entfernt, dass die entsprechenden Zeilen und Spalten in den Matrizen Υ und Φ sowie die entsprechenden Spalten in den Matrizen Ψ und Δ gelöscht werden. Zudem müssen die Matrizen Υ und Φ angepasst werden, um so eventuell zwei Vermittlungsknoten oder Bündelungspunkte miteinander zu verbinden, die zuvor nicht verbunden waren.

Verfahren zum Entfernen von Bündelungspunkten Bündelungspunkte verursachen zwar keine direkten Kosten, jedoch können zu viele zu einer ungünstigen Vernetzungsstruktur führen. Dies ist in Abbildung 5.2 dargestellt. Werden zur Anbindung der Endknoten möglichst kurze Leitungskanäle verwendet, so werden die beiden Endknoten n_1 und n_2 mit dem Bündelungspunkt b_1 verbunden. Optimal wäre jedoch für jeden Endknoten eine direkte Verbindung mit dem Bündelungspunkt b_2 . Zudem beeinflusst die Anzahl der Bündelungspunkte auch die Leistungsfähigkeit der entwickelten Positionierungsverfahren. Beispielsweise sinkt beim SA-basierten Verfahren die Wahrscheinlichkeit, dass ein *sinnvoller* Bündelungspunkt ausgewählt und verschoben wird, mit steigender Anzahl von Bündelungspunkten. Die Lösungsqualität ist also unter anderem abhängig von der Anzahl der Bündelungspunkte. Aus diesen Gründen müssen *irrelevante* Bündelungspunkte zeitweise, beispielsweise beim SA-basierten Positionierungsverfahren, wenn die Anzahl der Bündelungspunkte die Anzahl der Endknoten übersteigt ($|B| > |N|$), entfernt und somit deren Anzahl minimiert werden.

Ein Bündelungspunkt wird dann als irrelevant bezeichnet, wenn durch das Entfernen desselben die Leitungskosten nicht erhöht werden. In einem ersten Schritt kann ein Bündelungspunkt genau dann entfernt werden, wenn dieser

- exakt an der Position eines anderen Bündelungspunktes positioniert wurde, d. h. wenn für zwei beliebige Bündelungspunkte mit dem Index i und j gilt: $(x_i, y_i) = (x_j, y_j) \wedge i \neq j$ bzw. $d_{i,j} = 0$

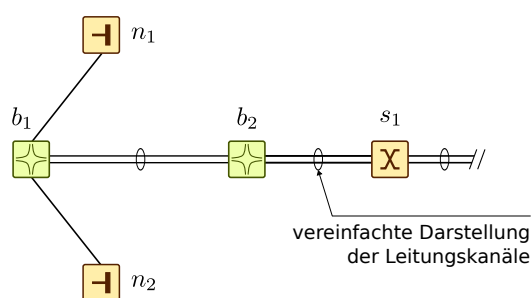


Abbildung 5.2: Teilnetz mit einem relevanten Bündelungspunkt mit zwei Nachbarn

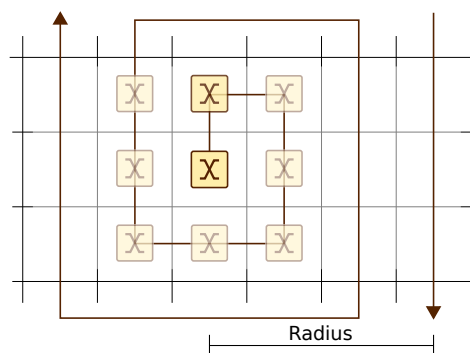


Abbildung 5.3: Verfahren zur abschließenden Positionsoptimierung

- oder mit keinem anderen Endknoten, Vermittlungsknoten oder Bündelungspunkt über ein Leitungssegment verbunden ist, d. h. ein Bündelungspunkt mit dem Index i ist lediglich mit *leeren* Leitungskanälen an das Netz angebunden:

$$\sum_{j=1}^{|S|+|B|} \phi_{i+|S|+1,j} + \sum_{j=1}^{|M|} \delta_{j,i+|S|+1} = 0 \quad (5.19)$$

Nach dem Entfernen der offensichtlich irrelevanten Bündelungspunkte können in einem zweiten Schritt eventuell weitere Bündelungspunkte entfernt werden. Ein Ansatz hierfür wäre, in Anlehnung an das Verfahren zum Entfernen von Vermittlungsknoten, die Bündelungspunkte zu entfernen, die exakt zwei Nachbarn besitzen. Jedoch ist dieser Ansatz nicht optimal. Im Teilnetz in Abbildung 5.2 würde es dazu führen, dass der Bündelungspunkt b_2 entfernt wird. Allerdings wäre es sinnvoller, den Bündelungspunkt b_1 zu entfernen und die Endknoten n_1 und n_2 mit dem Bündelungspunkt b_2 zu verbinden. Deshalb wird iterativ jeder Bündelungspunkt untersucht, ob durch dessen Entfernen die Leitungskosten verringert werden. Nur bei einer Verringerung, jedoch nicht wenn die Kosten unverändert bleiben, wird der betrachtete Bündelungspunkt auch tatsächlich entfernt. Das Entfernen von Bündelungspunkten erfolgt also *defensiv*. Dadurch wird, unabhängig von der Betrachtungsreihenfolge, im Teilnetz in Abbildung 5.2 korrekterweise b_1 und nicht b_2 entfernt.

Verfahren zur abschließenden Positionsoptimierung Untersuchungen haben gezeigt, dass insbesondere bei zufallsbasierten Heuristiken die Leitungskosten durch Verschieben der Vermittlungsknoten auf benachbarte Positionen in einem nachgelagerten Verfahren verringert werden können. In [3] veröffentlichten der Autor und Doumith ein Verfahren, bei dem die Positionen der Vermittlungsknoten an einem Raster ausgerichtet werden, das durch die Positionen der Endknoten bestimmt ist. Allerdings werden in dieser Veröffentlichung keine Umgebungskosten berücksichtigt. Eine Alternative ist, wie in Abbildung 5.3 dargestellt, jeden Vermittlungsknoten um seine aktuelle Position bis zu einem festgelegten Radius spiralförmig zu verschieben. Falls eine Verbesserung erreicht wird, wird der betrachtete Vermittlungsknoten neu positioniert. Dieses Verfahren ähnelt der Intensivierung des TS-basierten Verfahrens. Aus Laufzeitgründen ist es jedoch nicht empfehlenswert, bei jeder Verschiebung eines Vermittlungsknotens während der Optimierung dieses Verfahren anzuwenden.

5.3 Vernetzungsverfahren

Zur Minimierung der Leitungskosten müssen neben der kostengünstigen Positionierung die End- und Vermittlungsknoten sowie Bündelungspunkte kostengünstig miteinander vernetzt werden. Hierzu wird innerhalb der Positionierungsverfahren an mehreren Stellen die Funktion `ERSTELLEVERNETZUNGSSTRUKTUR` aufgerufen. Prinzipiell wird bei jeder Verschiebung eines Vermittlungsknotens oder Bündelungspunktes eine neue kostengünstige Vernetzungsstruktur erstellt. Während der Erstellung einer solchen werden die Knoten und Bündelungspunkte nicht verschoben. Zudem müssen neben den Umgebungskosten auch die Verkehrsanforderungen berücksichtigt werden, damit diese später erfüllt werden können.

Im Gegensatz zu den Positionierungsverfahren liegen den Vernetzungsverfahren keine bekannten Metaheuristiken zu Grunde. Vielmehr werden zur Erstellung einer kostengünstigen Vernetzungsstruktur problemspezifische Heuristiken entwickelt. Allerdings basieren diese ebenfalls auf dem Prinzip der Nachbarschaftssuche.

Zur Erstellung einer vollständigen Vernetzungsstruktur, d. h. sowohl von Leitungskanälen als auch Leitungen, sind folgende sequentielle Schritte notwendig (angelehnt an die Formulierung der Kostenfunktion aus Abschnitt 4.3):

1. Installiere Leitungskanäle

- 1.1. Für das Vermittlungsnetz, d. h. zwischen den Vermittlungsknoten und Bündelungspunkten, abhängig von Γ sowie den Positionen der Vermittlungsknoten und Bündelungspunkte (Berechne Variablen $v_{i,j}$).
- 1.2. Für das Zugangsnetz (Anbindung der Endknoten), d. h. zwischen den Endknoten und Vermittlungsknoten bzw. Bündelungspunkten, abhängig von Γ sowie den Positionen der End- und Vermittlungsknoten sowie der Bündelungspunkte (Berechne Variablen $\psi_{i,j}$).

2. Installiere Leitungen

- 2.1. Für das Vermittlungsnetz abhängig von Ψ , Υ und R (Berechne Variablen $\phi_{i,j}$).
- 2.2. Für das Zugangsnetz abhängig von Ψ , Υ und R (Berechne Variablen $\delta_{i,j}$).

In Abbildung 5.4 sind diese Schritte dargestellt. Zur Erstellung einer kostengünstigen Vernetzungsstruktur wird die Menge der Endknoten N , der Vermittlungsknoten S und der Bündelungspunkte B mit deren jeweiligen Position in der zweidimensionalen Ebene sowie die Matrix R mit den Verkehrsanforderungen und die Matrix Γ mit den Umgebungskosten benötigt. In dem Beispiel in Abbildung 5.4 befinden sich die vier Endknoten n_1, \dots, n_4 an den Positionen $(1,1)$, $(5,1)$, $(3,1)$, $(5,3)$, der Vermittlungsknoten s_1 an der Position $(3,2)$ und die beiden Bündelungspunkte b_1 und b_2 an den Positionen $(1,2)$ und $(5,2)$. Die Anzahl der Spalten in der Matrix Γ entspricht der Anzahl diskreter Felder in x -Richtung und die Anzahl der Zeilen entspricht der Anzahl diskreter Felder in y -Richtung.

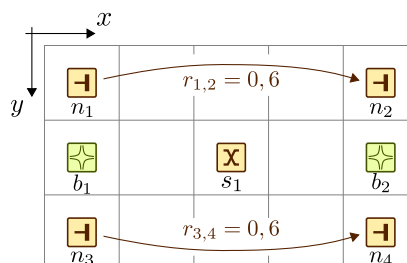
Eingabe N, S, B, R, Γ

- $N = \{n_1, n_2, n_3, n_4\}$

- $S = \{s_1\}$ und $B = \{b_1, b_2\}$

$$- R = \begin{matrix} & n_1 & n_2 & n_3 & n_4 \\ n_1 & \begin{pmatrix} 0,0 & 0,6 & 0,0 & 0,0 \end{pmatrix} \\ n_2 & \begin{pmatrix} 0,0 & 0,0 & 0,0 & 0,0 \end{pmatrix} \\ n_3 & \begin{pmatrix} 0,0 & 0,0 & 0,0 & 0,6 \end{pmatrix} \\ n_4 & \begin{pmatrix} 0,0 & 0,0 & 0,0 & 0,0 \end{pmatrix} \end{matrix}$$

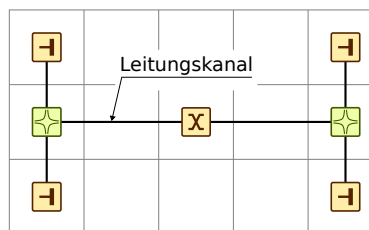
$$- \Gamma = \begin{matrix} & x_1 & x_2 & x_3 & x_4 & x_5 \\ y_1 & \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix} \\ y_2 & \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix} \\ y_3 & \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{matrix}$$



1. Installiere Leitungskanäle

1.1. Für das Vermittlungsnetz

$$\mathbf{Y} = \begin{matrix} & s_1 & b_1 & b_2 \\ s_1 & \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \\ b_1 & \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \\ b_2 & \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \end{matrix}$$



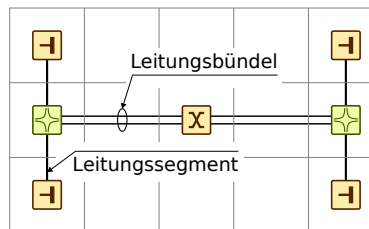
1.2. Für das Zugangsnetz

$$\mathbf{\Psi} = \begin{matrix} & s_1 & b_1 & b_2 \\ n_1 & \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \\ n_2 & \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \\ n_3 & \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \\ n_4 & \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

2. Installiere Leitungen (Leitungssegmente)

2.1. Für das Vermittlungsnetz

$$\mathbf{\Phi} = \begin{matrix} & s_1 & b_1 & b_2 \\ s_1 & \begin{pmatrix} 0 & 2 & 2 \end{pmatrix} \\ b_1 & \begin{pmatrix} 2 & 0 & 0 \end{pmatrix} \\ b_2 & \begin{pmatrix} 2 & 0 & 0 \end{pmatrix} \end{matrix}$$



2.2. Für das Zugangsnetz

$$\mathbf{\Delta} = \begin{matrix} & s_1 & b_1 & b_2 \\ n_1 & \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \\ n_2 & \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \\ n_3 & \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \\ n_4 & \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

Abbildung 5.4: Sequentielle Schritte zur Erstellung einer vollständigen Vernetzungsstruktur

Im Folgenden werden Heuristiken zur Erstellung einer vollständigen Vernetzungsstruktur beschrieben. Dabei wird unterschieden zwischen Heuristiken, die eine baumförmige Vernetzungsstruktur erstellen (Abschnitt 5.3.1) und solchen, die eine ausfallsichere Vernetzungsstruktur (Abschnitt 5.3.2) erstellen. Die Strategie der entwickelten Heuristiken ist, eine kostengünstige Vernetzungsstruktur in zwei Schritten zu erstellen. Zuerst wird mit einem *intelligenten* Eröffnungsverfahren eine zulässige Startlösung bestimmt. Anschließend wird solange versucht, eine zulässige Lösung iterativ durch lokale Strukturveränderungen zu verfeinern (Verfeinerungsverfahren), bis ein definiertes Abbruchkriterium erfüllt ist.

5.3.1 Baumförmige Vernetzungsstruktur

Für eine kostengünstige baumförmige Vernetzungsstruktur muss ein aufspannender Baum mit minimalen Kosten bestimmt werden. Der resultierende Baum muss die Verkehrsanforderungen erfüllen. Dieses Problem ist ein Sonderfall des bereits erwähnten OCST Problems; jedoch mit Fixkostenanteil (siehe Abbildung 4.5). In dem vorliegenden Problem sind die Kosten für einen Leitungskanal (Kante) abhängig von der Anzahl der Leitungen (siehe Gleichung (4.4)).

In [1] veröffentlichte der Autor mehrere Heuristiken zur Erstellung einer kostengünstigen baumförmigen Vernetzungsstruktur. In diesem Abschnitt werden die vielversprechendsten Heuristiken beschrieben.

5.3.1.1 Eröffnungsverfahren

Rothlauf [220] zeigt, dass ein MST als Anfangslösung zur Bestimmung eines OCST geeignet ist. In dem vorliegenden Optimierungsproblem ist der OCST gleich dem MST, falls der Bündelungseffekt und die Verkehrsanforderungen vernachlässigt werden. Folglich ist auch in diesem Fall das Erstellen eines MST durchaus ein geeignetes Eröffnungsverfahren.

Beim Erstellen eines MST werden alle Knoten eines Graphen miteinander verbunden. Da bei eingebetteten Kommunikationsnetzen nicht beliebige Leitungen zulässig sind, kann dieses Vorgehen nicht angewandt werden. So darf ein Endknoten nicht direkt mit einem anderen Endknoten verbunden werden. Aus diesem Grund werden beim Eröffnungsverfahren lediglich die Vermittlungsknoten und Bündelungspunkte in Form eines MST miteinander verbunden, d. h. die Leitungskanäle für das Vermittlungsnetz installiert. Das Eröffnungsverfahren führt somit den im vorigen Abschnitt erwähnten Schritt 1.1 durch. Die Schritte 1.2, 2.1 und 2.2 erfolgen durch das im Folgenden beschriebene Ergänzungsverfahren.

5.3.1.2 Ergänzungsverfahren

Ausgangspunkt des Ergänzungsverfahrens ist neben den endlichen Mengen N , S , B , den Matrizen \mathbf{R} und $\mathbf{\Gamma}$ eine Menge leerer, baumförmig vernetzter Leitungskanäle E zwischen den Vermittlungsknoten und den Bündelungspunkten. In Algorithmus 5.5 wird ein Verfahren beschrieben, das, basierend auf der Menge der Leitungskanäle E , die Vernetzungsstruktur um die fehlenden

Algorithmus 5.5 Ergänze Vernetzungsstruktur

Input: N, S, B, E, R, Γ ▷ E : Menge leerer, baumförmig vernetzter Leitungskanäle
Output: l ▷ Lösung mit baumförmiger Vernetzungsstruktur

- 1: Initialisiere l mit $\Psi^{(|N| \times (|S|+|B|))}$, $\Delta^{(|N| \times (|S|+|B|))}$, $\Upsilon^{(|S|+|B|) \times (|S|+|B|)}$ und $\Phi^{(|S|+|B|) \times (|S|+|B|)}$
- 2: **for all** $i \in E$ **do**
- 3: $v_{e_i, a_i} \leftarrow v_{a_i, e_i} \leftarrow 1$
- 4: $V \leftarrow S \cup B$
- 5: $E' \leftarrow \emptyset$
- 6: **for all** $n \in N$ **do**
- 7: $v \leftarrow \arg \min_{v \in V} (d_{n,v}(\Gamma))$
- 8: $\psi_{n,v} \leftarrow 1$
- 9: Berechne $\Delta(\theta, \Psi, \Upsilon, R)$
- 10: Berechne $\Phi(\theta, \Psi, \Upsilon, R)$
- 11: **return** l

Leitungskanäle zwischen den Endknoten und den Vermittlungsknoten bzw. Bündelungspunkten und um die benötigten physikalischen Leitungen ergänzt. Dieses Verfahren definiert somit die in Abschnitt 5.3 erwähnten Schritte 1.2 – 2.2.

Zu Beginn wird in Algorithmus 5.5 eine Lösungsinstanz l mit den notwendigen Matrizen und deren entsprechenden Größen initialisiert. Im nächsten Schritt werden die Elemente $v_{i,j}$ in Abhängigkeit von der Menge baumförmig vernetzter Leitungskanäle E bestimmt. Graphentheoretisch betrachtet ist E eine Kantenliste (siehe Abschnitt 3.1.2). Jede Kante $i \in E$ in der Liste besitzt einen eindeutigen Anfangs- a_i und Endknoten e_i . Diese Knoten entsprechen einem Element aus der Menge der Vermittlungsknoten und Bündelungspunkte ($a_i, e_i \in S \cup B$). Daraus folgt für die binären Variablen $v_{e_i, a_i} \leftarrow v_{a_i, e_i} \leftarrow 1$ (Zeile 3).

Im nächsten Schritt werden die Endknoten kostengünstig angebunden (Zeile 6 bis 8). Die Strategie bei der Anbindung ist, einen möglichst kurzen Leitungskanal zu installieren. Demzufolge wird zwischen jedem Endknoten und dessen nächsten (kostengünstigen) Vermittlungsknoten oder Bündelungspunkt ein Leitungskanal in Abhängigkeit von der Kostenmatrix Γ installiert. Die Strategie zielt darauf ab, dass möglichst frühzeitig ein Bündelungsgewinn genutzt werden kann. Basierend auf den installierten Leitungskanälen und unter Berücksichtigung der Verkehrsanforderungen wird die Anzahl benötigter physikalischer Leitungssegmente in jedem Leitungskanal berechnet und somit die erforderlichen Übertragungsraten sichergestellt.

Für die Berechnung der benötigten Leitungssegmente wird für jeden Endknoten das ein- und ausgehende Verkehrsaufkommen berechnet, der Maximalwert hieraus gewählt und dieser auf die nächste ganze positive Zahl aufgerundet. Dabei wird für das ein- und ausgehende Verkehrsaufkommen ein an der maximalen Übertragungsrate der gewählten Technologie normierter Wert genommen. Entsprechend des ganzzahligen Wertes werden die benötigten, unter Umständen parallelen physikalischen Leitungssegmente zwischen dem betrachteten Endknoten und seinem nächsten (kostengünstigsten) Vermittlungsknoten installiert. Der nächste Vermittlungsknoten wird mit einem geeigneten Algorithmus zum Finden eines kürzesten Weges bestimmt.

Abschließend wird die Anzahl benötigter physikalischer Leitungen zwischen den Vermittlungsknoten bestimmt (Zeile 10). Hierfür wird zuerst für jedes Paar von Vermittlungsknoten der kür-

Algorithmus 5.6 Lokales Suchverfahren**Input:** N, S, B, R, Γ **Output:** l

```

1:  $E \leftarrow \text{MST}(S, B)$  ▷  $E$ : Menge baumförmig vernetzter Leitungskanäle
2:  $l \leftarrow \text{ERGÄNZEVERNETHUNGSSTRUKTUR}(N, S, B, E, R, \Gamma)$  ▷ siehe Algorithmus 5.5
3: do
4:   improvement  $\leftarrow$  FALSE
5:    $E' \leftarrow E$ 
6:   for all  $e \in E'$  do
7:      $E'' \leftarrow E' \setminus \{e\}$ 
8:     Bestimme disjunkte Teilmengen  $U$  und  $V$ 
       des Graphen  $G_{E'} = (S \cup B, E') \mid U \cup V = S \cup B$ 
9:     for all  $(u, v) \in U \times V$  do
10:       $e' \leftarrow \{u, v\}$ 
11:       $E''' \leftarrow E'' \cup \{e'\}$ 
12:       $l' \leftarrow \text{ERGÄNZEVERNETHUNGSSTRUKTUR}(N, S, B, E''', R, \Gamma)$  ▷ siehe Algorithmus 5.5
13:      if  $z(l') < z(l)$  then
14:         $l \leftarrow l'$ 
15:         $E \leftarrow E'''$ 
16:        improvement  $\leftarrow$  TRUE
17: while improvement
18: return  $l$ 

```

zeste Weg berechnet. Anschließend wird das Verkehrsaufkommen zwischen benachbarten Vermittlungsknoten ermittelt. Basierend darauf kann die Anzahl benötigter, unter Umständen paralleler physikalischer Leitungssegmente zwischen den Vermittlungsknoten bestimmt werden.

Das Ergebnis aus Algorithmus 5.5 ist die zulässige Lösung l , deren Kosten mit Gleichung (4.11) berechnet werden können. Die Lösung l repräsentiert eine vollständige baumförmig vernetzte Topologie, die gegebene mittlere Verkehrsanforderungen erfüllt. Dabei werden keine Leistungskriterien berücksichtigt wie beispielsweise durch kurzzeitige, statistische Verkehrsschwankungen bedingte Pufferzeiten oder Rahmenverluste.

5.3.1.3 Verfeinerungsverfahren

Lokales Suchverfahren Nun wird ein Verfeinerungsverfahren beschrieben, das zuerst eine zulässige Startlösung mit dem Eröffnungsverfahren erzeugt und anschließend iterativ benachbarte Bäume untersucht. Allgemein gelten zwei Bäume dann als benachbart, wenn sie sich in genau einer Kante unterscheiden. Zu einem Baum mit n Knoten existieren abhängig von seiner Struktur zwischen $(n-1)(n-2)$ und $1/6 n(n-1)(n+1) - n + 1$ Nachbarschaftsbäume [220].

Das lokale Suchverfahren in Algorithmus 5.6 erstellt zuerst eine Anfangslösung nach dem soeben beschriebenen Verfahren. Es werden dann folgende Schritte solange wiederholt, bis keine Verbesserung erreicht werden konnte. Zuerst wird ein Leitungskanal e aus dem baumförmigen Vermittlungsnetz E' entfernt (Zeile 7). Dadurch wird der Baum in zwei Teilnetze getrennt. Im

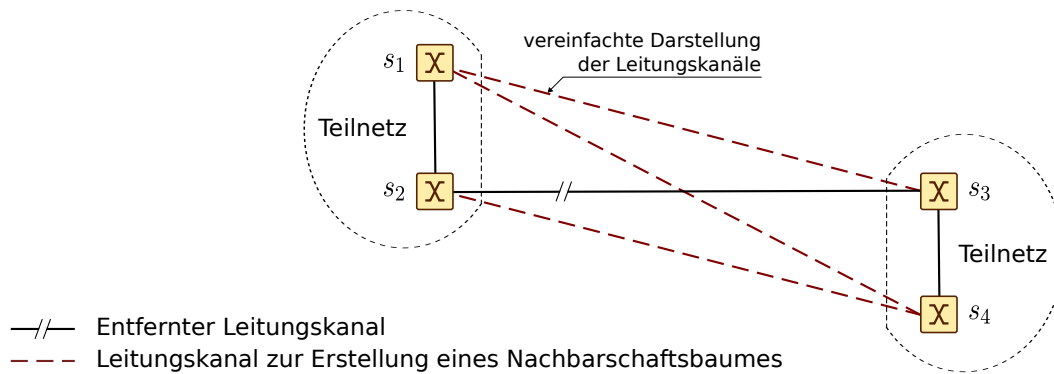


Abbildung 5.5: Erstellung eines Nachbarschaftsbaumes durch Entfernen und Hinzufügen eines Leitungskanals

nächsten Schritt werden die disjunkten Teilmengen U und V mit $U \cup V^1 = S \cup B$ ermittelt (Zeile 8). Die disjunkten Teilmengen U und V enthalten die Vermittlungsknoten und Bündelungspunkte der beiden Teilnetze. Ausgehend von den disjunkten Teilmengen werden alle möglichen Leitungskanäle, die die beiden Teilnetze miteinander verbinden, bestimmt.

Nach dem Bestimmen der möglichen Leitungskanäle werden sämtliche benachbarte Lösungen untersucht. Durch das Hinzufügen eines neuen Leitungskanals (in Zeile 10 und 11), der die beiden Teilnetze zusammenfügt, entsteht eine benachbarte Lösung (Nachbarschaftsbaum). Die soeben beschriebenen Schritte sind in Abbildung 5.5 beispielhaft dargestellt. Hier wird der Leitungskanal $\{s_2, s_3\}$ entfernt. Die disjunkten Teilmengen sind $U = \{s_1, s_2\}$ und $V = \{s_3, s_4\}$. Zur Verbindung der beiden Teilnetze kommen die Leitungskanäle $\{\{s_1, s_3\}, \{s_1, s_4\}, \{s_2, s_4\}\}$ in Frage.

Nachdem die beiden Teilnetze durch Hinzufügen jeweils eines neuen Leitungskanals zusammengefügt wurden, kann die vollständige Vernetzungsstruktur durch Aufruf des Algorithmus 5.5 ermittelt und die Kosten der benachbarten Lösung l' bestimmt werden. Falls diese Kosten geringer sind als die der aktuellen Lösung l (Zeile 13), wird die benachbarte Lösung als aktuelle Lösung akzeptiert (Zeile 14) und die Menge der baumförmig vernetzten Leitungskanäle E aktualisiert (Zeile 15). Zum Schluss wird die beste gefundene Lösung l zurück gegeben (Zeile 18).

Ein weiteres lokales Suchverfahren veröffentlichte der Autor in [1]. Dieses ist ähnlich zu dem soeben beschriebenen Verfahren und untersucht nur die benachbarten Bäume der Anfangslösung. Somit erfolgt bei einer Verbesserung keine wiederholte Ausführung. Folglich ist der potentiell untersuchte Lösungsraum wesentlich kleiner. Das Verfahren terminiert in der Regel schneller und findet Lösungen mit geringerer Qualität.

Gieriges Verfahren Zur Lösung eines Hub Location Problem werden häufig gierige (Such-) Verfahren eingesetzt. Diese Verfahren wählen in jedem Teilschritt die momentan beste Möglichkeit aus. Das im Algorithmus 5.7 beschriebene Verfahren zur Bestimmung einer kostengünstigen Vernetzungsstruktur basiert auf einem solchen gierigen Verfahren.

¹Das Symbol \cup steht für die Vereinigung disjunkter Mengen. Zwei Mengen U und V werden vereinigt, geschrieben $U \cup V$, wenn folgende Bedingung erfüllt ist: $U \cap V = \emptyset$ (die beiden Mengen sind disjunkt).

Algorithmus 5.7 Gieriges (Such-) Verfahren**Input:** N, S, B, R, Γ **Output:** l

```

define
   $d$  ▷ Suchtiefe
   $\beta_{\text{iter}}$  ▷ Anzahl zu untersuchender Nachbarschaftslösungen
1:  $E \leftarrow \text{MST}(S, B)$  ▷  $E$ : Menge baumförmig vernetzter Leitungskanäle
2:  $l \leftarrow \text{ERGÄNZEVERNETHUNGSSTRUKTUR}(N, S, B, E, R, \Gamma)$  ▷ siehe Algorithmus 5.5
3: do
4:   improvement  $\leftarrow$  FALSE
5:   for all  $e \in E$  do
6:     Bestimme Kosten  $c(e) \mid c : e \rightarrow \theta_{i,j}(\phi)$  ▷ siehe Gleichung (4.4)
7:     Sortiere Menge  $E$  in absteigender Reihenfolge  $c(e_1) \geq c(e_2) \geq \dots \geq c(e_{|E|})$ 
8:      $E' \leftarrow E$ 
9:     for  $i \leftarrow 1$  to  $\min(d, |E|)$  do
10:       $E'' \leftarrow E' \setminus \{e_i\}$ 
11:      Bestimme disjunkte Teilmengen  $U$  und  $V$ 
        des Graphen  $G_{E''} = (S \cup B, E'')$   $\mid U \cup V = S \cup B$ 
12:       $E^N \leftarrow \{u, v\} \mid (u, v) \in U \times V$ 
13:      for all  $n \in E^N$  do
14:        Bestimme Kosten  $c(n) \mid c : n \rightarrow c_{a_n, e_n}(\Gamma)$ 
15:        Sortiere Menge  $E^N$  in aufsteigender Reihenfolge  $c(n_1) \leq c(n_2) \leq \dots \leq c(n_{|E^N|})$ 
16:        for  $j \leftarrow 1$  to  $\min(\beta_{\text{iter}}, |E^N|)$  do
17:           $E''' \leftarrow E'' \cup \{n_j\}$ 
18:           $l' \leftarrow \text{ERGÄNZEVERNETHUNGSSTRUKTUR}(N, S, B, E''', R, \Gamma)$ 
▷ siehe Algorithmus 5.5
19:          if  $z(l') < z(l)$  then
20:             $l \leftarrow l'$ 
21:             $E \leftarrow E'''$ 
22:            improvement  $\leftarrow$  TRUE
23: while improvement
24: return  $l$ 

```

Vor der Ausführung des Algorithmus 5.7 müssen die beiden Steuerparameter d und β_{iter} definiert werden. Der Steuerparameter d legt die Suchtiefe und β_{iter} die Anzahl der in jedem Durchlauf zu untersuchenden Nachbarschaftslösungen fest.

Beim gierigen Verfahren wird zunächst die Menge der Leitungskanäle E den Kosten nach in absteigender Reihenfolge sortiert (Zeile 7). Dazu werden für jedes Element $e \in E$ die Kosten bestimmt. Als Kostenfunktion wird die Gleichung (4.4) verwendet. Im nächsten Schritt werden iterativ der i teuerste Leitungskanal entfernt und somit das Netz in zwei Teilnetze getrennt. Die Menge E^N beinhaltet die Leitungskanäle, mit denen die beiden Teilnetze verbunden werden können und somit eine Nachbarschaftslösung entsteht.

Nach dem Bestimmen der Menge E^N (Zeile 12) werden die Elemente dieser Menge den Kosten nach in aufsteigender Reihenfolge sortiert. Dazu werden für jedes Element $n \in E^N$ die Kosten

bestimmt. Da zu diesem Zeitpunkt die Anzahl in n benötigter physikalischer Leitungssegmente unbekannt und der Leitungskanal leer ist, kann nicht die Gleichung (4.4) als Kostenfunktion angewandt werden. Deshalb werden die minimalen Kosten c_{a_n, e_n} für den Weg vom Anfangspunkt a_n zum Endpunkt e_n des Leitungskanals n berechnet und als Kriterium verwendet. Anschließend wird die vollständige Vernetzungsstruktur mit den jeweils β_{iter} billigsten Leitungskanälen bestimmt und l' zugewiesen. Falls eine der untersuchten Nachbarschaftslösungen l' zu einer Verbesserung führt, wird diese als die aktuell beste Lösung akzeptiert (Zeile 20) und die Menge E (Zeile 21) aktualisiert. Dieses Vorgehen wird solange wiederholt, bis keine Verbesserung erzielt wird. Zum Schluss wird die beste gefundene Lösung l zurück gegeben.

Die beiden Verfeinerungsverfahren optimieren die Vernetzungsstruktur zwischen den Vermittlungsknoten und den Bündelungspunkten. Die Vernetzungsstruktur zwischen den Endknoten und den Vermittlungsknoten bzw. Bündelungspunkten wurde nicht optimiert. Diese werden, wie im Algorithmus 5.5 beschrieben, ebenfalls nach einem gierigen Verfahren angebunden. Um eine möglichst kostengünstige Anbindung der Endknoten zu erreichen, könnte auch im Verfeinerungsverfahren jeder Endknoten an die β_{iter} -nächsten Vermittlungsknoten bzw. Bündelungspunkte angebunden werden.

Ein weiterer Ansatz wäre, die Vermittlungsknoten und Bündelungspunkte so zu clustern, dass jeder Bündelungspunkt seinem jeweils nächsten Vermittlungsknoten zugeordnet wird, der auf einem Pfad über die bereits vorhandenen Leitungskanäle erreicht werden kann. Anschließend wird dann für jeden Endknoten der nächste (kostengünstigste) Vermittlungsknoten ermittelt und mit diesem unter Ausnutzung bereits vorhandener Leitungskanäle verbunden. Ein ähnlicher Ansatz wird in [13] für die Wahl geeigneter Zugangspunkte für die Anbindung eines Endknotens bei einer ausfallsicheren Vernetzungsstruktur vorgeschlagen.

5.3.2 Ausfallsichere Vernetzungsstruktur

Bisher wurden Verfahren zur Erstellung einer kostengünstigen bäumförmigen Vernetzungsstruktur behandelt. Wie bereits erwähnt ist die Ausfallsicherheit hinsichtlich eines Einfachfehlers eine wesentliche Anforderung an sicherheitskritische eingebettete Kommunikationsnetze. Damit diese Anforderung erfüllt werden kann, muss ein solches Netz mindestens 2-fach zusammenhängend sein. Dies bedeutet, dass bei Ausfall eines beliebigen Vermittlungsknotens oder bei Beschädigung eines Leitungskanals inklusive aller darin befindlichen Leitungssegmente noch mindestens ein Pfad zwischen allen Endknotenpaaren existiert.

Ein trivialer Ansatz wäre, das Kommunikationsnetz quasi vollständig, mit Ausnahme der Endknoten, zu klonen und vollständig redundant zu installieren. Einen solchen Ansatz verfolgt AFDX (siehe Abschnitt 2.7). Allerdings muss hier sichergestellt werden, dass Leitungen nicht im gleichen Kanal installiert werden. Dieser Ansatz kann zwar einen Ausfall eines Vermittlungsknotens oder einer Leitung kompensieren, erzielt jedoch kostenunünstige Lösungen.

Die Herausforderung bei der Erstellung einer ausfallsicheren Vernetzungsstruktur ist, dass zwischen allen Endknotenpaaren mindestens zwei kanten- und knotendisjunkte Pfade existieren – bei gleichzeitiger Minimierung der Leitungskosten. Mit dieser Anforderung allein lassen sich ausfallsichere Lösungen finden, die eine hohe Konnektivität der Endknoten erfordern [13]. Aus Kostengründen wird jedoch zusätzlich gefordert, dass jeder Endknoten nur zweifach, d. h. über

zwei physikalische Schnittstellen, angebunden werden soll. Folglich können bekannte Verfahren der Graphentheorie nicht direkt angewandt werden.

In Abschnitt 4.4.2 wurde erwähnt, dass eine 1+1 Absicherung erfolgen soll. Somit muss für eine ausfallsichere Vernetzungsstruktur die benötigte Übertragungsrate für eine 1+1 Absicherung unter Berücksichtigung der Verkehrsanforderungen ermittelt werden. Analog zur Bestimmung eines OCST existieren auch für das vorliegende Optimierungsproblem keine exakten Lösungsverfahren. Deshalb wird zur Lösung dieses Problems erneut eine zweistufige Heuristik, bestehend aus Eröffnungs- und Verfeinerungsverfahren, angewandt.

5.3.2.1 Eröffnungsverfahren

Das Eröffnungsverfahren soll eine möglichst kostengünstige zulässige Startlösung erzeugen. Eine zulässige Lösung ist eine Topologie mit struktureller Redundanz. Dadurch kann ein Ausfall eines beliebigen Vermittlungsknotens oder Leitungskanals kompensiert werden.

Da die Endknoten genau zweifach angebunden werden sollen, müssen die End- und Vermittlungsknoten unterschiedlich behandelt werden. Daraus ergibt sich ein zweistufiges Verfahren: (i) Aufbau einer ausfallsicheren Vernetzungsstruktur zwischen den Vermittlungsknoten (Vermittlungsnetz) und (ii) redundante Anbindung der Endknoten (Zugangsnetz). Im Vermittlungsnetz müssen zwischen allen Vermittlungsknoten zwei disjunkte Pfade existieren, wobei Bündelungspunkte genutzt werden können. Bündelungspunkte selbst müssen nicht redundant angebunden werden. Aus der Literatur [240] bekannte Algorithmen basieren im Wesentlichen auf zwei Grundideen: (i) In einem bidirektionalen Ring ist die geforderte Ausfallsicherheit stets gegeben. (ii) Zwei minimale Spannbäume können, wie vom Autor in [6] beschrieben, so miteinander kombiniert werden, dass die Segmente des einen Spannbaumes den anderen derart ergänzen, dass zu jedem Pfad innerhalb eines Spannbaumes ein zweiter, disjunkter Pfad durch den anderen entsteht. Somit ist die geforderte strukturelle Redundanz gewährleistet. Verschiedene Strategien, die Eigenschaften von Kreisen bzw. Zyklen ausnutzen, werden in [13] diskutiert. In der vorliegenden Arbeit wird ein Ansatz, basierend auf der Zusammenführung und Kombination zweier minimaler Spannbäume (2-MST), verfolgt. In Algorithmus 5.8 ist ein Verfahren beschrieben, das eine vollständige, d. h. sowohl für das Vermittlungs- als auch für das Zugangsnetz ausfallsichere Vernetzungsstruktur erstellt. Im ersten Teil (Zeile 2 – 4) wird das ausfallsichere Vermittlungsnetz und im zweiten Teil (Zeile 5 – 18) ein redundantes Zugangsnetz erstellt. Im dritten Teil wird die vollständige Vernetzungsstruktur berechnet (Zeile 19 – 24).

Der Aufbau eines ausfallsicheren Vermittlungsnetzes ist in Abbildung 5.6 dargestellt. Zunächst wird ein erster MST G'_T über alle Vermittlungsknoten und Bündelungspunkte ermittelt und die entsprechenden Leitungskanäle hinzugefügt. Anschließend wird der Konnektivitätsgrad für jeden Vermittlungsknoten bestimmt. Der Konnektivitätsgrad ist definiert als die Anzahl benachbarter Vermittlungsknoten, die auf knotendisjunkten Pfaden erreicht werden können. Danach wird ein zweiter MST G''_T über alle Vermittlungsknoten mit einem Konnektivitätsgrad $\epsilon < 2$ erstellt (Zeile 3). Beim Aufbau der Leitungskanalstruktur kann noch keine Aussage über den Bündelungsgewinn eines einzelnen Bündelungspunktes gemacht werden. Daher würde eine Untersuchung auf die optimale Zuordnung eines Bündelungspunktes die Komplexität steigern, ohne einen nennenswerten Gewinn zu erzielen. Demzufolge werden beim zweiten MST keine

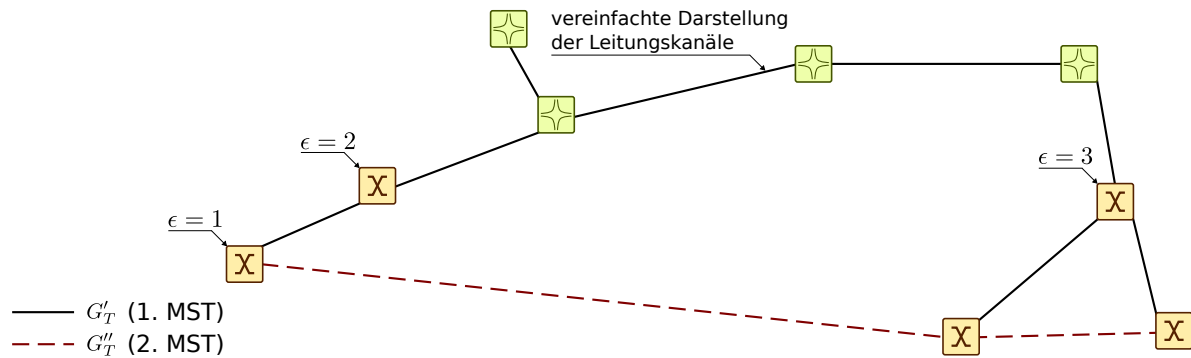


Abbildung 5.6: Kombination zweier minimaler Spannäume G'_T und G''_T zur Erzeugung einer ausfallsicheren Vernetzungsstruktur

Bündelungspunkte berücksichtigt. Die Zuordnungen der Bündelungspunkte können im nachgelagerten Verfeinerungsverfahren geeignet optimiert werden.

Im zweiten Teil werden die Endknoten redundant angebunden. Hierfür müssen diese mit zwei unabhängigen Vermittlungsknoten verbunden werden. Unabhängig bedeutet, dass keine gemeinsamen Leitungskanäle oder Bündelungspunkte verwendet werden dürfen. Jede Zugangsleitung kann beliebig viele Bündelungspunkte durchlaufen. Analog zur einfachen Anbindung der Endknoten beim OCST werden für die Anbindung der Endknoten möglichst kurze Leitungskanäle verwendet, da auch hier zu diesem Zeitpunkt die Anzahl in n benötigter physikalischer Leitungssegmente unbekannt ist. Es werden für jeden Endknoten $n \in N$ die möglichen Zugangspunkte (Vermittlungsknoten oder Bündelungspunkt) nach ihrem kürzesten Weg in aufsteigender Reihenfolge sortiert. Anschließend wird ein Leitungskanal zwischen dem anzubindenden Endknoten und dem nächsten Zugangspunkt hinzugefügt (Zeile 11). Ausgehend von diesem Zugangspunkt wird der kürzeste Pfad zu einem Vermittlungsknoten $s \in S$ in dem Graphen $G(S \cup B, E)$ und in Abhängigkeit von der Kostenmatrix Γ gesucht.

Als nächstes wird ein zweiter Pfad P' von einem der nächsten möglichen Zugangspunkte zu einem Vermittlungsknoten $s' \in S$ gesucht. Dabei darf der zweite Pfad nicht den Vermittlungsknoten und keine Bündelungspunkte des ersten Pfades nutzen ($E \setminus P$), so dass die beiden Zugangsleitungen über die gefundenen Pfade sowohl kanten- als auch knotendisjunkt sind. In [13] werden unterschiedliche Verfahren (Tiefensuche, Breitensuche und Uniforme-Kosten-Suche) für die Suche der Vermittlungsknoten und Pfadsuche diskutiert. Zudem wird noch ein sogenanntes Clustering-Verfahren zur redundanten Anbindung der Endknoten beschrieben. Bei diesem Verfahren werden die Bündelungspunkte und Vermittlungsknoten so zusammengefasst, dass bei zwei Zugangspunkten aus unterschiedlichen Clustern die beiden Pfade zwischen den Zugangspunkten und den Vermittlungsknoten disjunkt sind.

Im dritten Teil von Algorithmus 5.8 werden die Matricelemente der Lösungsinstanz l berechnet und somit die vollständige Vernetzungsstruktur festgelegt. Die Elemente $v_{i,j}(E)$ und $\psi_{i,j}(E')$ werden anhand der Anfangs- und Endpunkte der Elemente i der Leitungskanalmenge E bzw. E' berechnet. Anschließend werden die in den einzelnen Leitungskanälen benötigten Leitungssegmente unter Berücksichtigung der Verkehrsanforderungen berechnet. In [13] werden hierfür geeignete Verfahren beschrieben. Diese erzeugen bereits bei der Berechnung der beiden MSTs und beim Aufbau des Zugangsnetzes jeweils eine zulässige initiale Leitungsstruktur mit

Algorithmus 5.8 Eröffnungsverfahren zur Erstellung einer ausfallsicheren Vernetzungsstruktur**Input:** N, S, B, R, Γ **Output:** l

▷ Lösung mit ausfallsicherer Vernetzungsstruktur

- 1: Initialisiere l mit $\Psi^{[N] \times (|S|+|B|)}$, $\Delta^{[N] \times (|S|+|B|)}$, $\Upsilon^{(|S|+|B|) \times (|S|+|B|)}$ und $\Phi^{(|S|+|B|) \times (|S|+|B|)}$
- 2: $E_{G'_T} \leftarrow \text{MST}(S \cup B)$ ▷ $E_{G'_T}$: Menge baumförmig vernetzter Leitungskanäle des 1. MST
- 3: $E_{G''_T} \leftarrow \text{MST}(\{\forall s \in S \mid \epsilon(s) = 1\})$
 ▷ $E_{G''_T}$: Menge baumförmig vernetzter Leitungskanäle des 2. MST
- 4: $E \leftarrow E_{G'_T} \cup E_{G''_T}$
- 5: $E' \leftarrow \emptyset$ ▷ E' : Menge der Leitungskanäle für das Zugangsnetz
- 6: **for all** $n \in N$ **do**
- 7: $G = (S \cup B \cup \{n\}, E, c_{i,j})$
- 8: $A \leftarrow S \cup B$ ▷ A : Menge der möglichen Zugangspunkte
- 9: Sortiere Menge A nach ihrem kürzesten Weg zu n in Abhängigkeit von Γ
 in aufsteigender Reihenfolge: $c_{n,a_1} \leq c_{n,a_2} \leq \dots \leq c_{n,a_{|S|+|B|}} \mid a_i \in A$
- 10: $e \leftarrow \{n, a_1\}$
- 11: $E' \leftarrow E' \cup \{e\}$ ▷ Hinzufügen eines Leitungskanals
- 12: $P \leftarrow \arg \min_{s \in S} (\text{KÜRZESTERWEG}(a_1, s, G(S \cup B, E), \Gamma))$
- 13: **for** $i \leftarrow 2$ **to** $|A|$ **do**
- 14: $P' \leftarrow \arg \min_{s' \in S} (\text{KÜRZESTERWEG}(a_i, s', G(S \cup B, E \setminus P), \Gamma))$
- 15: **if** $(P' \cup P) = \emptyset$ **then**
- 16: $e' \leftarrow \{n, a_i\}$
- 17: $E' \leftarrow E' \cup \{e'\}$ ▷ Hinzufügen eines Leitungskanals
- 18: **break**
- 19: **for all** $i \in E'$ **do**
- 20: $\psi_{a_i, e_i} \leftarrow \psi_{e_i, a_i} \leftarrow 1$
- 21: **for all** $i \in E$ **do**
- 22: $v_{a_i, e_i} \leftarrow v_{e_i, a_i} \leftarrow 1$
- 23: Berechne $\Delta(\theta, \Psi, \Upsilon, R)$
- 24: Berechne $\Phi(\theta, \Psi, \Upsilon, R)$
- 25: **return** l

minimaler Anzahl von Leitungssegmenten. Später wird dann diese Struktur entsprechend den Verkehrsanforderungen angepasst. Dazu werden die kürzesten Wege zwischen allen Endknoten ermittelt und auf diesen eine ausreichende Anzahl von Leitungen installiert.

Bei einer ausfallsicheren Vernetzungsstruktur mit einer 1 + 1 Absicherung müssen hierzu zwei disjunkte Pfade ermittelt und entsprechend den Verkehrsanforderungen Leitungen hinzugefügt werden. Ansonsten kann ein Ausfall eines Pfades nicht kompensiert werden. Die Suche nach den disjunkten Pfadpaaren basiert auf dem Suurballe-Algorithmus (siehe Abschnitt 3.1.4). Das Verfahren berücksichtigt zudem, dass zwar ein einzelner Pfad Bündelungspunkte mehrfach überschreiten kann, gleichzeitig die beiden gefundenen Pfade jedoch keine gemeinsamen Bündelungspunkte aufweisen. In Abbildung 5.7 ist ein Beispiel mit einem mehrfach überschrittenem Bündelungspunkt dargestellt. Der Bündelungspunkt b_1 wird vom Arbeitspfad P_{s_1, s_2} mehrfach überschritten. Jedoch wird dieser Bündelungspunkt nur vom Arbeitspfad genutzt und nicht vom Ersatzpfad P'_{s_1, s_2} . Fällt also der Bündelungspunkt b_1 aus, besteht weiterhin ein Pfad zwi-

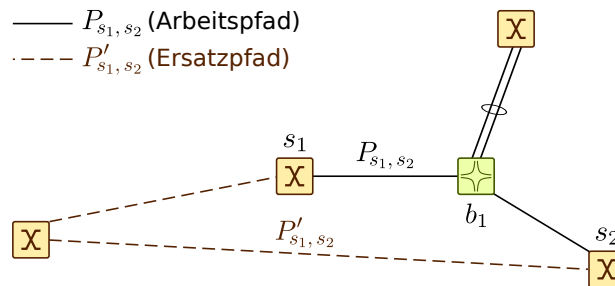


Abbildung 5.7: Beispiel zweier disjunkter Pfade mit mehrfach überschrittenem Bündelungspunkt

schen den beiden Vermittlungsknoten s_1 und s_2 . In [13] werden weitere Verfahren beschrieben, um mögliche Sonderfälle zu detektieren und aufzulösen.

5.3.2.2 Verfeinerungsverfahren

Durch ein geeignetes Verfeinerungsverfahren soll nun die mit dem Eröffnungsverfahren erzeugte Startlösung in ihrer Qualität schrittweise verbessert werden. Eine Verfeinerung kann entweder durch kantenzahl-bewahrende Kreismodifikationen, Brezel- und Quezelmodifikationen oder Reduktion von Konnektivitätsgraden erreicht werden [240]. Bei einer Brezelmodifikation wird ein Leitungskanal eines Kreises entfernt und die offenen Knoten mit zwei zufällig ausgewählten Knoten auf dem Kreisrest verbunden. Umgekehrt werden bei einer Quezelmodifikation zwei Kreise zu einem größeren zusammengefasst. Voraussetzung ist die Existenz von zwei Kreisen, die einen gemeinsamen Teilpfad besitzen. Das Auffinden von zwei Kreisen mit einem gemeinsamen Teilpfad ist jedoch aufwändig.

Das im Folgenden beschriebene Verfahren modifiziert existierende Kreise im Vermittlungsnetz und basiert darauf, dass jeder Kreis über eine Menge von Knoten durch einen anderen Kreis ersetzt werden kann. Folglich wird die Konnektivität der Knoten nicht verändert. Bevor existierende Kreise modifiziert werden können, müssen diese detektiert werden. Hierfür werden in [13] zwei Strategien zum Finden von Kreisen mit einer definierten minimalen Länge vorgestellt: (i) Gierige und (ii) zufällige Strategie. Bei der gierigen Strategie werden, ausgehend von einem zufällig ausgewählten Startknoten $s \in S$, mit einem uniformen Kostensuchverfahren solange die benachbarten Vermittlungsknoten untersucht, bis die minimale Länge und der Ausgangsknoten erreicht werden. Im Gegensatz zur Breitensuche wird bei der uniformen Kostensuche jeweils stets der kostengünstigste Pfad zum nächsten noch nicht besuchten Knoten weiter verfolgt. Demzufolge wird schließlich der kürzeste Kreis mit der geforderten Länge gefunden. Hingegen wird bei der zufälligen Strategie, wie es der Name schon assoziiert, jeweils ein zufällig ausgewählter Pfad weiter verfolgt.

Bei der Kreisdetektion werden die benachbarten Vermittlungsknoten betrachtet. Daher erfolgt die Kreisdetektion auf Leitungsebene. Es werden also keine Bündelungspunkte betrachtet. Die Leitungen bestehen aus mehreren Leitungssegmenten, die sich in unterschiedlichen Leitungskanälen befinden. Bei der Entfernung und Ersetzung zweier vollständiger Leitungen gehen unter Umständen bereits erzielte Bündelungsgewinne verloren. Um dennoch eine Verbesserung zu erzielen, werden, nachdem ein Kreis detektiert wurde, lediglich zwei Leitungskanäle, wie in

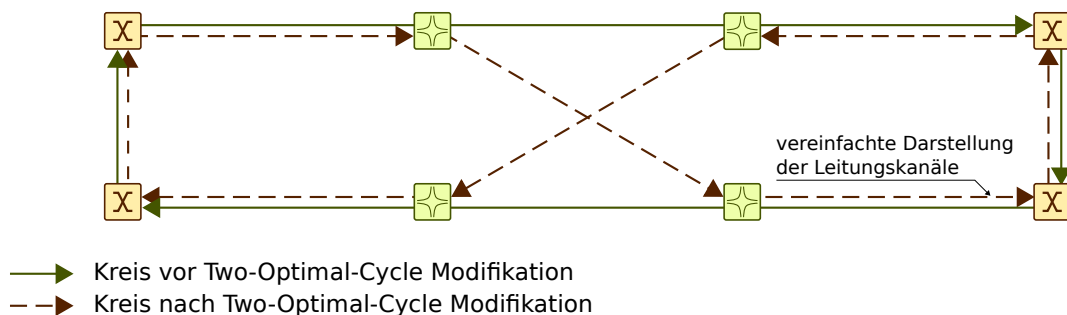


Abbildung 5.8: Modifikation der Leitungskanäle beim Two-Optimal-Cycle Verfahren [13]

Abbildung 5.8 dargestellt, durch zwei neue ersetzt (*Two-Optimal-Cycle*). Dabei verbinden die beiden neuen Leitungskanäle die nach der Entfernung offenen Enden der Kreissegmente kreuzförmig miteinander. Durch dieses Vorgehen können auch Kreuzungen, die vor der Modifikation des Kreises vorlagen, aufgelöst werden.

Basierend auf dem Prinzip der gierigen Vorgehensweise wird das teuerste Leitungskanalpaar zur Entfernung ausgewählt, da hier die größte Verbesserung zu erwarten ist. Eine ebenfalls geeignete Vorgehensweise, die in [13] beschrieben und untersucht wird, ist eine zufällige Auswahl eines Leitungskanalpaares. Bei beiden Auswahlstrategien muss sichergestellt werden, dass die ausgewählten Leitungskanäle keine gemeinsamen Endpunkte (Vermittlungsknoten und Bündelungspunkte) besitzen. Anderenfalls werden die Leitungskanäle faktisch nur vertauscht, so dass keine Veränderung der bestehenden Struktur erfolgt. Nach dem Vertauschen der Leitungskanäle muss eine neue zulässige Leitungsstruktur berechnet werden.

Ähnlich wie beim gierigen Verfahren in Abschnitt 5.3.1 ist auch bei diesem Verfahren die Anzahl zu betrachtender Leitungskanalpaare parametrisierbar. Somit wird nicht nur die Ersetzung eines Leitungskanalpaares betrachtet, sondern die Ersetzung von d Paaren. Zudem kann die Qualität des Verfeinerungsverfahrens erhöht werden, indem die soeben beschriebenen Schritte bei jedem Aufruf mehrmals durchgeführt werden. Dadurch werden nicht nur ein Kreis, sondern mehrere untersucht. Der Steuerparameter mI_{ter} legt die maximale Anzahl aufeinander folgender Iterationen ohne Verbesserung fest. Folglich bricht das Verfeinerungsverfahren ab, sobald in mI_{ter} Durchläufen keine Verbesserung erzielt wurde. Wird in einem Durchlauf eine Verbesserung erzielt, wird die Anzahl bisher untersuchter Kreise auf null zurückgesetzt.

In [13] wird auch ein *Three-Optimal-Cycle* Verfahren diskutiert, bei dem drei Kreiskanten durch drei neue Kanten ausgetauscht werden, wobei eine minimale Kreislänge von sechs Leitungskanälen vorausgesetzt wird. Die beiden Verfahren Two- und Three-Optimal-Cycle ersetzen existierende Kreise durch andere. Folglich wird bei diesen Verfahren die grundlegende Struktur im Gegensatz zu einer Brezel- oder Quezelmodifikation nicht verändert.

Darüber hinaus wird in [13] ein Verfahren zum Ausschluss von Nachbarschaftsbeziehungen beschrieben. Bisher wurden beim Erstellen einer ausfallsicheren Vernetzungsstruktur keine räumlichen Distanzen zwischen den Leitungskanälen berücksichtigt. Pfade können sich also kreuzen oder nahe beieinanderliegen. Allerdings ist die Wahrscheinlichkeit hoch, dass bei Beschädigungen durch eine mechanische Zerstörung räumlich benachbarte Leitungskanäle ebenfalls beschädigt werden. Deshalb kann beim Entwurf eine definierte minimale Distanz zwischen

sich gegenseitig absichernden Pfaden gefordert werden. Beim Verfahren zum Ausschluss von Nachbarschaftsbeziehungen werden während des Verfeinerungsverfahrens die geforderten Mindestabstände erreicht, indem die Kosten für die Bewertung einer Lösung erhöht werden, falls bestimmte Leitungskanäle die Abstandsanforderungen nicht erfüllen. Folglich werden diese Lösungen mit einer hohen Wahrscheinlichkeit nicht akzeptiert und verworfen.

5.4 Verfahren zur Bestimmung einer Kompromisslösung

Bisher wurden Verfahren zur Optimierung der Leitungskosten beschrieben. In Abschnitt 4.1.2 wurde der Zielkonflikt zwischen Minimierung der Anzahl der Vermittlungsknoten und Minimierung der Gesamtlänge der Leitungen bzw. den Leitungskosten beschrieben. Ein etabliertes Vorgehen zur Lösung multikriterieller Optimierungsprobleme ist das Bestimmen einer Kompromisslösung. In diesem Abschnitt wird nun abschließend ein iteratives Verfahren zur Bestimmung einer solchen entwickelt.

Die Kompromisslösung mit den minimalen Vernetzungskosten ist abhängig von den Kosten für einen Vermittlungsknoten c_S , deren Anzahl x sowie den Leitungskosten $c_{\Sigma L}$ (siehe Abbildung 5.9). Die Leitungskosten wiederum sind unter anderem vom Kostenfaktor λ , den Verkehrsanforderungen, der Zielumgebung sowie den Positionen der Vermittlungsknoten und Bündelungspunkte abhängig. Im Grunde genommen können sämtliche Kosten nur vom Hersteller bzw. Ausrüster quantifiziert und somit die optimale Kompromisslösung bestimmt werden.

Selbst wenn sämtliche Kosten bekannt sind, kann die Kompromisslösung nicht direkt bestimmt werden, da zuvor die minimalen Leitungskosten für eine gegebene Anzahl von Vermittlungsknoten bestimmt werden müssen. Dies bedeutet, dass das Kommunikationsnetz zuvor entworfen und optimiert werden muss. Ein mögliches Vorgehen ist, für jede Anzahl von Vermittlungsknoten die minimalen Leitungskosten unter Anwendung der entwickelten Optimierungsverfahren zu bestimmen. Dies würde einer vollständigen Enumeration entsprechen. Wie bereits erwähnt liegt die Obergrenze für die Anzahl sinnvoller Vermittlungsknoten bei n Endknoten bei $n - 2$. Es müssen also $n - 2$ Probleminstanzen gelöst werden. Folglich ist das Bestimmen der Kompromisslösung bei einer großen Anzahl von Endknoten aufwändig. Allerdings können die Probleminstanzen unabhängig voneinander gelöst und somit das Verfahren parallelisiert werden. Nachdem für jede Anzahl ($x = 1, \dots, n - 2$) von Vermittlungsknoten die minimalen Leitungskosten $c_{\Sigma L}^{\min}(x)$ (vgl. Abschnitt 4.3) mit den entwickelten Verfahren² bestimmt wurden, können durch Minimierung der Zielfunktion $z(x)$

$$\min(z(x)) \quad \text{mit} \quad z(x) = \underbrace{x \cdot c_S}_{\text{Kosten für die Vermittlungsknoten}} + \underbrace{c_{\Sigma L}^{\min}(x)}_{\text{Leitungskosten}} \quad (5.20)$$

die Vernetzungskosten optimiert werden. Die Vernetzungskosten setzen sich aus den Kosten für die Vermittlungsknoten und den Leitungskosten zusammen.

²Die in Abschnitt 5.2 entwickelten Verfahren gehen neben einer Menge von Endknoten auch von einer Menge von Vermittlungsknoten aus. In diesem Abschnitt ist lediglich die Anzahl der Vermittlungsknoten x gegeben. Deshalb muss zur Bestimmung der minimalen Leitungskosten in Abhängigkeit von x trivialerweise eine Menge mit entsprechender Anzahl von Vermittlungsknoten erzeugt werden.

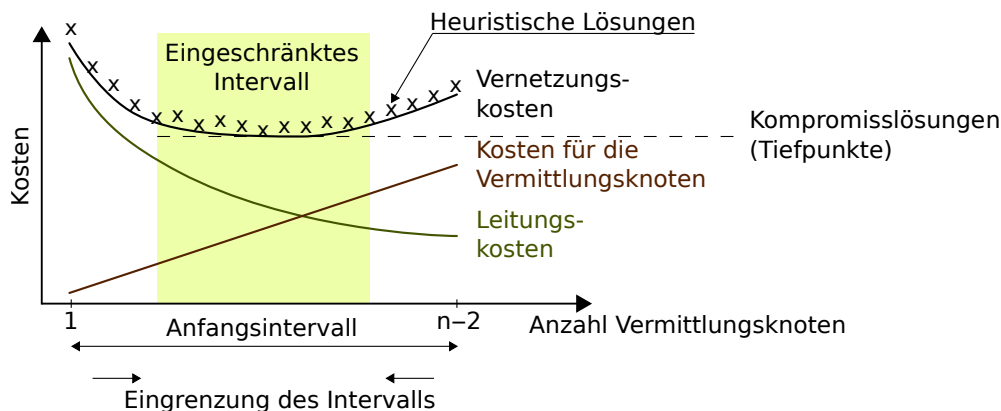


Abbildung 5.9: Prinzip zur Bestimmung einer Kompromisslösung

In Gleichung 5.20 werden für alle Vermittlungsknoten identische Kosten angenommen. Folglich werden im Vergleich zu den Leitungskosten keine Umgebungskosten berücksichtigt. Es ist aber durchaus möglich, auch für die Kosten der Vermittlungsknoten deren Position und somit die Umgebungskosten mit einzubeziehen. Die Modellierung und das Vorgehen wären analog zu den Leitungskosten. Das im Folgenden beschriebene Lösungsverfahren wäre trotzdem in der dargestellten Form uneingeschränkt anwendbar.

Das vollständige Enumerieren zur Bestimmung einer Kompromisslösung kann durch Berücksichtigung von Eigenschaften der Kostenfunktionen eingeschränkt werden. Unter der Annahme, dass alle Vermittlungsknoten gleich teuer sind und die Einzelkosten unabhängig von der Ausstattung (Anzahl der Anschlüsse, etc.) sind, ist die Kostenfunktion für die Vermittlungsknoten linear steigend. Die Funktion für die Leitungskosten ist mit zunehmender Anzahl der Vermittlungsknoten monoton fallend, d. h. sie wird nur kleiner oder bleibt konstant (siehe Abbildungen 4.1 und 5.9). Aufgrund dieser Eigenschaften kann der Tiefpunkt der Funktion für die Vernetzungskosten (Kompromisslösung) mittels Intervallschachtelung iterativ bestimmt werden. Begonnen wird mit dem Intervall $[1, n - 2]$. Anschließend wird dann das Intervall schrittweise eingegrenzt. In jedem Schritt wird untersucht, ob sich die Kompromisslösung links oder rechts von der aktuell unteren oder oberen Intervallgrenze befindet und entsprechend die Intervallgrenzen angepasst. Wenn in einem Durchlauf keine Verbesserung erzielt wurde, entspricht entweder der untere oder obere Grenzwert der Kompromisslösung, d. h. der optimalen Anzahl von Vermittlungsknoten. Die Intervallschachtelung entspricht im Prinzip der unvollständigen (begrenzten) Enumeration (siehe Abschnitt 3.2.2). Bei der Intervallschachtelung wird das Problem ebenfalls in mehrere Teilprobleme zerlegt, um herauszufinden, ob eine optimale Lösung in einem Zweig von Teilproblemen (Intervall) enthalten ist.

Die beschriebene Intervallschachtelung setzt jedoch voraus, dass für jede beliebige Anzahl von Vermittlungsknoten die minimalen Leitungskosten (optimale Lösung) ermittelt werden können. Allerdings können in akzeptabler Zeit nur Lösungen gefunden werden, die sehr nahe am Optimum sind (heuristische Lösungen). Die Qualität einer heuristischen Lösung, d. h. wie weit sie von der optimalen Lösung entfernt ist, kann jedoch nicht bestimmt werden. Somit ist die Intervallschachtelung in Kombination mit heuristischen Lösungsverfahren nur begrenzt anwendbar, da der tatsächliche Kurvenverlauf der Leitungskosten nicht ermittelt werden kann. Experimentelle Untersuchungen haben jedoch gezeigt, dass zumindest die Tendenz erkennbar ist. Deshalb kann mit den heuristischen Lösungsverfahren das Anfangsintervall $[1, n - 2]$ zumindest einge-

grenzt werden. Das eingegrenzte Intervall (siehe Abbildung 5.9) kann dann mittels einer vollständigen Enumeration untersucht werden. Dieses Vorgehen minimiert zwar nicht den Aufwand zur Bestimmung einer Kompromisslösung, verringert ihn jedoch erheblich. Auch die vollständige Enumeration des eingeschränkten Intervalls erfolgt mittels einer Heuristik. Folglich ist das Ergebnis eine *heuristische Kompromisslösung*.

6 Evaluierung der Verfahren

In diesem Kapitel erfolgt eine Evaluierung der neu entwickelten Verfahren. Nach der Beschreibung der Evaluierungsstrategie werden zuerst die Positionierungsverfahren und anschließend die Vernetzungsverfahren evaluiert. Hierfür werden die Optimierungsverfahren mit gleichem Anwendungszweck miteinander verglichen. Es werden stets mehrere Probleminstanzen mit gleicher Größe (Anzahl der Endknoten, Anzahl der Vermittlungsknoten, etc.) untersucht. Sämtliche Ergebnisse werden ausführlich diskutiert und bewertet. Bei Vernetzungsverfahren, die über Steuerparameter verfügen, wird deren Einfluss auf die Lösungsqualität und den Berechnungsaufwand untersucht. Für die Positionierungsverfahren wurden bereits umfassende Arbeiten [15, 14] zur optimalen Wahl und den Einflüssen der Steuerparameter (Cooling Schedule u. ä.) veröffentlicht. Auch für die vom Autor veröffentlichten Arbeiten [2, 3, 5] wurden im Vorfeld diverse Untersuchungen zur Parameterjustierung durchgeführt.

6.1 Evaluierungsstrategie

Die Evaluierung der Optimierungsverfahren soll unabhängig von einem konkreten Anwendungsszenario erfolgen. Es sollen vielmehr deren allgemeine Leistungsfähigkeit bewertet werden. Üblicherweise ist die Qualität der gefundenen Lösung und die benötigte Rechenzeit von Interesse. Allerdings ist Letzteres von untergeordneter Bedeutung. Während bei Anwendungen mit zeitlichen Anforderungen hinsichtlich der Lösungsfindung teilweise mehrere Sekunden inakzeptabel sind, kann bei der Kostenoptimierung von Topologien eingebetteter Kommunikationsnetze auch eine Rechenzeit von mehreren Tagen bis noch Wochen adäquat sein. Aufgrund der Kostensensitivität in eingebetteten Bereichen spielt die Qualität der gefundenen Lösung im Vergleich zur Rechenzeit eine wesentlich größere Rolle.

Dennoch werden die Optimierungsverfahren auch hinsichtlich ihres Berechnungsaufwandes miteinander verglichen. Der Berechnungsaufwand kann in verbrauchter Rechenzeit oder Anzahl der Operationen gemessen werden. Letzteres hat den Vorteil, dass es unabhängig von der Leistungsfähigkeit eines Rechners und dessen Auslastung ist. Im Folgenden wird überwiegend die Anzahl untersuchter Nachbarschaftslösungen als Metrik für den Berechnungsaufwand bewertet. Immerhin wird für das Erzeugen einer Nachbarschaftslösung die meiste Rechenzeit verwendet. Im Vergleich dazu können Operationen wie das Erzeugen einer Zufallszahl und Berechnen der Akzeptanzwahrscheinlichkeit vernachlässigt werden. Zudem kann die Rechenzeit durch Optimierungen des Quellcodes, Parallelisierung, etc. verbessert werden und ist somit immer eine relative Aussage. Hingegen ist die Qualität der gefundenen Lösung in Abhängigkeit von den untersuchten Nachbarschaftslösungen wesentlich aussagekräftiger.

Zur Bewertung der Leistungsfähigkeit werden, wo möglich, Optimierungsverfahren mit gleichem Anwendungszweck miteinander verglichen. Hierfür werden Probleminstanzen zufällig erzeugt, d. h. die Knoten und Bündelungspunkte werden anfänglich zufällig positioniert und die Verkehrsanforderungen zufällig erzeugt. Um eine stochastische Unabhängigkeit zu erreichen und allgemeingültige Aussagen treffen zu können, werden stets 100 zufällig erzeugte Probleminstanzen untersucht. Es werden also mehrere unabhängige Zufallsexperimente durchgeführt (quasi Monte-Carlo Simulation).

Im nächsten Schritt wird jedes Optimierungsverfahren auf dieselben 100 zufällig erzeugten Probleminstanzen angewandt, d. h. die zu vergleichenden Verfahren haben identische Probleminstanzen zu lösen bzw. zu optimieren. Dieses Vorgehen gewährleistet einen fairen Vergleich. Danach wird für jedes Verfahren aus den $n = 100$ Zielfunktionswerten (Einzelmessungen) z_i der arithmetische Mittelwert $\bar{z} = 1/n \sum_{i=1}^n z_i$ berechnet. Für den Vergleich zweier Verfahren k und l wird dann die prozentuale Differenz Δz aus den beiden Mittelwerten berechnet:

$$\Delta z = \left(\frac{\frac{1}{n} \sum_{i=1}^n z_i^k}{\frac{1}{n} \sum_{j=1}^n z_j^l} - 1 \right) \cdot 100 \quad (6.1)$$

Aus diesem Quotient kann dann abgeleitet werden, welches Optimierungsverfahren im Mittel besser ist. Zur Ergänzung zu den in diesem Kapitel aufgeführten arithmetischen Mittelwerten \bar{z} werden im Anhang C die dazugehörigen 95%-Konfidenzintervalle (cint) aufgeführt. Ein *schmales* Konfidenzintervall deutet auf eine geringe Streuung der Zielfunktionswerte hin. Daraus lässt sich schließen, dass die Messergebnisse nicht von Ausreißern (*Extrem-Probleminstanzen*) dominiert werden. Eine solche Analyse kann auch als probabilistische Analyse bezeichnet werden [189].

Sämtliche Optimierungsverfahren wurden, soweit nichts anderes erwähnt, in einer eigens entwickelten Softwarebibliothek implementiert. Die Architektur und der Funktionsumfang dieser Bibliothek sind im Anhang A umfassend beschrieben.

6.2 Positionierungsverfahren

6.2.1 Eingeschränkte Probleminstanzen

Zur Evaluierung der Heuristiken wäre es wünschenswert, von möglichst vielen Probleminstanzen die exakte Lösung zu kennen. In Abschnitt 6.1 wurde erwähnt, dass für eingeschränkte Probleminstanzen die exakte Lösung bestimmt werden kann. Die verschiedenen Positionierungsverfahren und deren Anwendbarkeit bzw. Einschränkungen sind in Tabelle 6.1 übersichtlich aufgelistet. Die exakten Verfahren (MST, ST und MILP) berücksichtigen nicht alle Eigenschaften eingebetteter Kommunikationsnetze. Zudem kann bei den beiden exakten Verfahren MST und ST die Anzahl der Vermittlungsknoten nicht konfiguriert werden. Aufgrund dieser Nachteile können die exakten Verfahren lediglich zur Bewertung der Heuristiken und nicht zur Lösung praxisrelevanter Probleme angewandt werden.

Bei dem folgenden Vergleich werden keine positionsabhängigen Kosten berücksichtigt. Als Distanzmetrik wird die Manhattan Distanz (siehe Abschnitt 4.2.1) verwendet. Zudem wird der

Tabelle 6.1: Vergleich der Verfahren zur Positionierung von Vermittlungsknoten

Verfahren	Optimale Lösung	Anzahl Vermittlungsknoten konfigurierbar	Positionierungsstrategie ¹		Bündelungseffekt	Positionsabhängige Kosten
			B	I		
MST	+	–	–	+	–	–
ST	+	–	+	–	–	–
MILP	+	+	+	+	–	○
SA-basiert	○	+	+	+	+	+
TS-basiert	○	+	+	+	+	+

+ möglich

○ teilweise möglich

– nicht möglich

¹ B Beliebig positionierbare Vermittlungsknoten

I Integrierte Vermittlungsknoten

Bündelungseffekt vernachlässigt. Folglich sind die Leitungskosten gleich der Gesamtlänge der Leitungen.

Die optimalen Lösungen für 1, 2 und 3 Vermittlungsknoten werden mit dem in Abschnitt 5.2.1 beschriebenen MILP bestimmt. Zur Lösung linearer Optimierungsprobleme gibt es mehrere, teilweise frei verfügbare Softwarebibliotheken. Eine bekannte, nicht kommerzielle Bibliothek zum Lösen gemischt-ganzzahliger linearer Optimierungsprobleme ist Solving Constraint Integer Programs (SCIP) vom Konrad-Zuse-Zentrum für Informationstechnik in Berlin [241, 242]. Diese Bibliothek wird auch in dieser Arbeit zum Lösen des MILP verwendet.

Zur Bestimmung eines ST und eines MST wird die Softwarebibliothek GeoSteiner [243] verwendet. Bei einem ST und einem MST entspricht die Anzahl der Steiner- bzw. Verzweigungspunkte der Anzahl der Vermittlungsknoten und kann nicht konfiguriert werden. Aus diesem Grund sind die Kosten für diesen Fall in den Tabellen 6.2 und 6.3 in der Zeile [1, |N|] enthalten; wobei die maximale Anzahl $|N| - 2$ nicht überschreitet. Für diesen speziellen Vergleich werden die Heuristiken mit $|S| = |N|$ initialisiert. Anschließend können mit dem in Abschnitt 5.2.3 beschriebenen Verfahren die irrelevanten Vermittlungsknoten entfernt werden.

Die Heuristiken werden wie folgt parametrisiert: (i) Cooling Schedule für das SA-basierte Verfahren mit $\tau = 1,0$, $p = 0,9$, $mIter = 500$, $mAtmp = 500$ und $mImpr = 100$. (ii) TS-basiertes Verfahren mit $mAtmpB = 0$ (da kein Bündelungseffekt berücksichtigt wird), $mIter = 10^5$ und $tenure = 13$. Die maximale Anzahl der Positionierungsversuche für Vermittlungsknoten wurde in Abhängigkeit von der Strategie (beliebig positionierbare und integrierte Vermittlungsknoten) parametrisiert. Für beliebig positionierbare Vermittlungsknoten wurde $mAtmpS = 20$ gewählt. Für integrierte Vermittlungsknoten wurde $mAtmpS = 0$ gewählt. Dadurch, dass $mAtmpB = 0$ und $mAtmpS = 0$ gewählt wurden, werden die Variablen $sIter$ und $jIter$ (siehe Algorithmus 5.4) niemals erhöht, was dazu führt, dass das Verfahren nicht terminiert. Deshalb muss für diesen Sonderfall die Ausführung der *while*-Schleife in Zeile 13 des Algorithmus 5.4 geeignet vermieden werden (beispielsweise durch Ergänzung einer vorangestellten *if*-Bedingung). Dadurch entfällt die Intensivierung und es erfolgt lediglich eine Diversifizierung. Da bei diesen eingeschränkten Probleminstanzen kein Bündelungseffekt berücksichtigt wird, werden sowohl beim SA- als auch beim TS-basierten Verfahren keine Bündelungspunkte erzeugt, positioniert und verschoben.

Tabelle 6.2: Vergleich der Positionierungsverfahren mit beliebig positionierbaren Vermittlungsknoten und eingeschränkten Probleminstanzen

Problemgröße		optimale Lösung		SA-basiert			TS-basiert		
N	S	\bar{z} [KE]	\bar{t} [min]	\bar{z} [KE]	\bar{t} [min]	Δz [%]	\bar{z} [KE]	\bar{t} [min]	Δz [%]
10	1	4 507,2	< 1	4 507,2	< 1	0,00	4 507,2	< 1	0,00
	2	3 347,8	< 1	3 349,3	< 1	0,04	3 348,3	< 1	0,01
	3	2 886,6	< 1	2 890,1	< 1	0,12	2 896,6	< 1	0,35
	[1,10]	2 294,0	< 1	2 313,1	< 1	0,83	2 356,8	< 1	2,74
15	1	6 956,8	< 1	6 957,6	< 1	0,01	6 956,8	< 1	0,00
	2	5 121,4	< 1	5 122,9	< 1	0,03	5 121,9	< 1	0,01
	3	4 325,9	≈ 105	4 330,5	< 1	0,11	4 328,1	< 1	0,05
	[1,15]	2 890,1	< 1	2 919,0	< 1	1,00	2 992,6	≈ 2	3,55
20	1	9 481,0	< 1	9 481,0	< 1	0,00	9 481,0	< 1	0,00
	2	6 945,7	< 1	6 950,2	< 1	0,06	6 946,9	< 1	0,02
	3	5 721,9	≈ 105	5 784,9	< 1	1,10	5 780,4	< 1	1,02
	[1,20]	3 376,2	< 1	3 417,5	≈ 1	1,22	3 492,2	≈ 3	3,44

Bei der zufälligen Erzeugung der 100 Probleminstanzen werden Endknoten zufällig in einer Ebene mit der Größe 1000×1000 Längeneinheiten verteilt. Neben der Qualität der gefundenen Lösung wird auch der Berechnungsaufwand bewertet. Da die exakten Verfahren nicht auf dem Prinzip der Nachbarschaftssuche basieren, wird die Rechenzeit und nicht die Anzahl betrachteter Nachbarschaftslösungen gemessen.

In den Tabellen 6.2 und 6.3 werden die von den beiden Heuristiken gefundenen Lösungen mit den optimalen Lösungen hinsichtlich den Leitungskosten sowie die benötigte mittlere Rechenzeit \bar{t} , gemessen in Minuten, für unterschiedliche Problemgrößen verglichen. Bei der Berechnung der prozentualen Differenz Δz entsprechen die mittleren Kosten der optimalen Lösung dem Grundwert (Hundert).

An der mittleren Rechenzeit wird deutlich, dass lediglich für kleine Probleminstanzen (≤ 20 Endknoten) die optimale Lösung bestimmt werden kann. Die Bestimmung der optimalen Lösung bei beliebig positionierbaren Vermittlungsknoten mit nur 15 und 20 Endknoten und 3 Vermittlungsknoten dauert für eine Probleminstanz im Mittel nahezu 2 Stunden. Dabei werden nur sehr eingeschränkte Probleminstanzen betrachtet. Die Bestimmung der optimalen Lösung bei integrierten Vermittlungsknoten mit nur 20 Endknoten und 3 Vermittlungsknoten dauert im Mittel mehr als 11 Stunden. Bei integrierten Vermittlungsknoten werden, wie in Abschnitt 5.2 beschrieben, die Kosten zur Verbindung zweier Endknoten in einer Kostenmatrix gespeichert, um somit auch positionsabhängige Kosten berücksichtigen zu können. Dadurch wird die Matrix mit den Nebenbedingungen für das lineare Programm größer und die Rechenzeit steigt an, obwohl der Lösungsraum mit integrierten Vermittlungsknoten im Vergleich zum Lösungsraum mit beliebig positionierbaren Vermittlungsknoten normalerweise kleiner ist.

In [5] wurde vom Autor et al. mit dem gleichen Optimierungsverfahren eine Probleminstanz mit 20 Endknoten und 5 Vermittlungsknoten gelöst. Dies dauerte für eine Probleminstanz mit beliebig positionierbaren Vermittlungsknoten länger als 7 Tage und mit integrierten Vermitt-

Tabelle 6.3: Vergleich der Positionierungsverfahren mit integrierten Vermittlungsknoten und eingeschränkten Probleminstanzen

Problemgröße		optimale Lösung		SA-basiert			TS-basiert		
N	S	\bar{z} [KE]	\bar{t} [min]	\bar{z} [KE]	\bar{t} [min]	Δz [%]	\bar{z} [KE]	\bar{t} [min]	Δz [%]
10	1	4 616,0	< 1	4 616,0	< 1	0,00	4 616,0	< 1	0,00
	2	3 462,1	≈ 12	3 462,1	< 1	0,00	3 607,4	< 1	4,20
	3	2 995,6	$\approx 2\frac{1}{2}$	2 995,6	< 1	0,00	3 229,4	< 1	7,80
	[1,10]	2 553,7	< 1	2 553,7	< 1	0,00	2 553,7	< 1	0,00
15	1	7 114,5	< 1	7 114,5	< 1	0,00	7 114,5	< 1	0,00
	2	5 235,1	≈ 15	5 235,1	< 1	0,00	5 510,3	< 1	5,26
	3	4 368,7	≈ 58	4 455,5	< 1	1,99	4 832,7	< 1	10,62
	[1,15]	3 236,8	< 1	3 236,8	< 1	0,00	3 236,8	< 1	0,00
20	1	9 653,8	< 1	9 653,8	< 1	0,00	9 653,8	< 1	0,00
	2	7 081,1	≈ 460	7 081,1	< 1	0,00	7 366,6	< 1	4,03
	3	5 923,6	≈ 700	5 923,6	< 1	0,00	6 359,1	< 1	7,35
	[1,20]	3 793,6	< 1	3 793,6	< 1	0,00	3 793,6	< 1	0,00

lungsknoten ebenfalls ungefähr 7 Stunden. Zudem wurden in [5] die optimalen Lösungen für Probleminstanzen mit integrierten Vermittlungsknoten unter Berücksichtigung positionsabhängiger Kosten bestimmt. Die Rechenzeiten hierfür waren ähnlich.

Aus den Tabellen 6.2 und 6.3 wird die Leistungsfähigkeit der beiden Heuristiken deutlich. Beide finden Ergebnisse, die häufig optimal oder zumindest sehr nahe am Optimum sind. Für das SA-basierte Verfahren beträgt die maximale Abweichung der Mittelwerte 1,99%; beim TS-basierten Verfahren 10,62%. An dieser Stelle sei noch erwähnt, dass das TS-basierte Verfahren nicht mit Blick auf integrierte Vermittlungsknoten entwickelt wurde. Lediglich durch eine geeignete Parametrisierung ($mAtmpS = 0$) wird es für diesen Spezialfall angepasst. Eine ausschließliche Diversifizierung ist also nicht immer ausreichend, um Lösungen mit einer sehr hohen Qualität zu bestimmen.

Die Rechenzeiten der Heuristiken zur Bestimmung einer kostengünstigen oder -optimalen Lösung sind bei den untersuchten Problemgrößen überwiegend kleiner als 1 Minute. Folglich behalten die Heuristiken ihre Geschwindigkeit auch für größere Probleminstanzen bei. Zusammenfassend sind beide Heuristiken zur Lösung des vorliegenden Optimierungsproblems geeignet. Das SA-basierte Verfahren findet durchweg Lösungen mit hoher Qualität. Auch das TS-basierte Verfahren erzielt gute Lösungen. Lediglich bei der Positionierung integrierter Vermittlungsknoten und bei einer großen Anzahl beliebig positionierbarer Vermittlungsknoten weichen die Ergebnisse etwas stärker vom Optimum ab.

Zusammenfassend liefern die Heuristiken hervorragende Ergebnisse bei geringem Berechnungsaufwand. In Abschnitt 3.2.3.4 wurde bereits erwähnt, dass in der Praxis auftretende Optimierungsprobleme häufig so in ihrer Struktur geartet sind, dass sie über sehr viele gute Lösungen verfügen. Es scheint so, als würde das vorliegende Optimierungsproblem zu regelmäßigen Strukturen tendieren und wäre *gutartig genug* [199], um es mit Heuristiken behandeln zu kön-

nen. Das Problem exakt zu lösen ist extrem (rechen-) aufwändig, jedoch eine nahezu optimale Lösung zu finden hingegen ist zeitlich gesehen wesentlich *einfacher*.

6.2.2 Uneingeschränkte Probleminstanzen

Nach dem Vergleich mit eingeschränkten Probleminstanzen werden nun die beiden Heuristiken auf uneingeschränkte Probleminstanzen angewandt und deren Leistungsfähigkeit miteinander verglichen. Die beiden Heuristiken sollen die optimalen Positionen der Vermittlungsknoten und Bündelungspunkte bestimmen. Die optimale Vernetzungsstruktur wird mit anderen Verfahren bestimmt. Deshalb werden bei diesem Vergleich die Vermittlungsknoten und Bündelungspunkte in Form eines MST miteinander vernetzt. Es werden also zur Erstellung der Vernetzungsstruktur lediglich die in den Abschnitten 5.3.1.1 und 5.3.1.2 beschriebenen Verfahren angewandt. Erneut werden 100 Probleminstanzen zufällig erzeugt, d. h. zufällige Positionierung der Endknoten und beim SA-basierten Verfahren zufällige, initiale Positionierung der Bündelungspunkte. Sämtlichen Probleminstanzen liegen zur Berücksichtigung der positionsabhängigen Kosten zwei verschiedene Matrizen Γ zugrunde: (i) Eine homogene Kostenmatrix, bei der alle Teilbereiche $\gamma_{x,y}$ den gleichen Wert besitzen; in dieser Untersuchung 0,5 und (ii) eine heterogene, zufällig erzeugte Kostenmatrix mit Autokorrelationsstruktur. Der arithmetische Mittelwert $\bar{\gamma}$ der Elemente der Matrix Γ beträgt bei dieser Kostenmatrix ebenfalls 0,5. In den Tabellen 6.4 und 6.5 sind die homogene Kostenmatrix mit den identischen Werten mit = und die heterogene Kostenmatrix mit \neq gekennzeichnet. Beide Kostenmatrizen haben die Größe 100×100 Längeneinheiten. Verkehrsanforderungen werden erst später bei der Bewertung der Vernetzungsverfahren berücksichtigt.

Die Heuristiken werden wie folgt parametrisiert: (i) Cooling Schedule für das SA-basierte Verfahren mit $\tau = 1,0$, $p = 0,9$, $mIter = 100$, $mAtmp = 100$, $mImpr = 25$ und initiale Anzahl der Bündelungspunkte $|N| - |S| - 2$ (siehe Algorithmus 5.2). (ii) TS-basiertes Verfahren mit $mAtmpB = 2$, $mIter = 2500$. Die maximale Anzahl der Positionierungsversuche für Vermittlungsknoten wurde in Abhängigkeit von der Strategie (beliebig positionierbare und integrierte Vermittlungsknoten) parametrisiert. Für beliebig positionierbare Vermittlungsknoten wurde $mAtmpS = 8$ und für integrierte Vermittlungsknoten $mAtmpS = 0$ gewählt.

Ein Vergleich der beiden Heuristiken ist nur dann aussagekräftig, wenn die Anzahl untersuchter Nachbarschaftslösungen und folglich der betrachtete Anteil am Lösungsraum in etwa in der gleichen Größenordnung liegen. Die Parameter für das TS-basierte Verfahren werden so gewählt, dass diese Bedingung weitestgehend erfüllt ist. Wie bereits erwähnt ist im Vergleich zum SA-basierten Verfahren das TS-basierte Verfahren deterministisch. Daher kann die Anzahl erzeugter Nachbarschaftslösungen berechnet werden. Die Diversifizierung und folglich die äußere Schleife (in Algorithmus 5.4 ab Zeile 6) wird $|N|$ -mal ausgeführt. Da während der Intensivierung jeder Vermittlungsknoten $mAtmpS$ -mal auf vier Felder und jeder Bündelungspunkt $mAtmpB$ -mal auf acht Felder verschoben werden, gilt für die Anzahl der Ausführungen der inneren Schleife (in Algorithmus 5.4 ab Zeile 13):

$$\text{iter} = \left\lceil \frac{mIter}{mAtmpS \cdot |S| \cdot 4 + mAtmpB \cdot |B| \cdot 8} \right\rceil \quad (6.2)$$

Es gilt also für die Anzahl der Nachbarschaftslösungen, die durch ein Verschieben eines Vermittlungsknotens erzeugt werden:

$$\beta_S = (\text{iter} \cdot \text{mAtmpS} \cdot |S| \cdot 4) \cdot |N| \quad (6.3)$$

Bei der Diversifizierung wird jeder Vermittlungsknoten an eine andere Position verschoben. Diese neu erzeugte Lösung wird nicht als Nachbarschaftslösung betrachtet und daher nicht zur Anzahl der Nachbarschaftslösungen β_S dazu addiert.

Für die Anzahl der Nachbarschaftslösungen, die durch ein Verschieben eines Bündelungspunktes erzeugt werden, gilt:

$$\beta_B = (\text{iter} \cdot \text{mAtmpB} \cdot |B| \cdot 8) \cdot |N| \quad (6.4)$$

Beim TS-basierten Verfahren wird zu Beginn bei jedem Endknoten genau ein Bündelungspunkt positioniert und während des Optimierungsprozesses keiner entfernt. Demzufolge gilt $|B| = |N|$ und die Gleichung (6.4) kann vereinfacht werden:

$$\beta_B = \text{iter} \cdot \text{mAtmpB} \cdot |N|^2 \cdot 8 \quad (6.5)$$

In den Tabellen 6.4 und 6.5 sind die mittleren Zielfunktionswerte \bar{z} (mittlere Leitungskosten) der beiden Heuristiken sowie die mittlere Anzahl betrachteter Nachbarschaftslösungen aufgeführt. Dabei wird unterschieden, ob eine Nachbarschaftslösung durch Verschieben eines Vermittlungsknotens β_S oder eines Bündelungspunktes β_B (bzw. beim SA-basierten Verfahren durch Hinzufügen oder Entfernen) erzeugt wurde. Zur Berechnung der prozentualen Differenzen Δz werden die Ergebnisse des SA-basierten Verfahrens als Grundwert angenommen. Sämtliche Werte in den beiden Tabellen sind arithmetische Mittelwerte über 100 Probleminstanzen. In allen Szenarien findet das SA-basierte Verfahren Lösungen mit einer höheren Qualität, obwohl meistens die Anzahl betrachteter Nachbarschaftslösungen kleiner und somit der Berechnungsaufwand geringer ist. Die Differenz zwischen den Zielfunktionswerten der beiden Verfahren ist unabhängig von der Problemgröße. Bei Problemgrößen mit 100 Endknoten und beliebig positionierbaren Vermittlungsknoten beträgt die Differenz maximal 10,91%.

Erkennbar ist wiederum, dass das TS-basierte Verfahren nicht für Probleminstanzen mit integrierten Vermittlungsknoten ausgerichtet wurde. Hier ist die Differenz zwischen den beiden Verfahren im Vergleich zu Probleminstanzen mit beliebig positionierbaren Vermittlungsknoten noch größer und beträgt maximal sogar 21,33%.

In den Tabellen 6.4 und 6.5 ist auch ersichtlich, dass mit geringerem Kostenfaktor für zusätzliche Leitungssegmente λ geringere Kosten erzielt werden. Die Ursache dafür ist, dass bei einem geringeren Faktor zusätzliche Leitungen günstiger sind und die Bündelung von Leitungen einen verhältnismäßig großen Kosteneinspareffekt hat. Im Grunde genommen können Ergebnisse mit unterschiedlichen Kostenfaktoren nicht miteinander verglichen werden.

Tabelle 6.4: Vergleich der Positionierungsverfahren mit beliebig positionierbaren Vermittlungsknoten

Problemgröße		Parameter		SA-basiert			TS-basiert			Δz [%]		
$ N $	$ S $	λ	Γ	\bar{z} [KE]	$\bar{\beta}_S$	$\bar{\beta}_B$	\bar{z} [KE]	$\bar{\beta}_S$	$\bar{\beta}_B$			
50	5	0,6	=	478,6	5 722	90 518	498,7	24 000	120 000	4,20		
			≠	427,3	10 197	107 212	449,8	24 000	120 000	5,27		
		0,8	=	526,3	8 823	85 208	567,5	24 000	120 000	7,83		
			≠	465,4	22 410	130 166	505,9	24 000	120 000	8,70		
		10	0,6	=	380,2	14 002	75 801	394,1	48 000	120 000	3,66	
				≠	341,3	24 354	91 063	358,5	48 000	120 000	5,04	
		0,8	=	402,3	19 260	78 370	429,7	48 000	120 000	6,81		
			≠	358,8	40 415	93 423	386,8	48 000	120 000	7,80		
	25	0,6	=	307,7	41 970	56 308	313,2	80 000	80 000	1,79		
			≠	277,9	70 295	55 284	287,5	80 000	80 000	3,45		
		0,8	=	314,3	46 966	55 357	324,8	80 000	80 000	3,34		
			≠	281,9	83 365	48 904	296,6	80 000	80 000	5,21		
75	10	0,6	=	540,7	8 106	94 965	562,4	48 000	180 000	4,01		
			≠	486,7	15 405	130 145	512,1	48 000	180 000	5,22		
		0,8	=	589,7	12 608	99 487	631,2	48 000	180 000	7,04		
			≠	514,4	35 094	140 323	570,5	48 000	180 000	10,91		
		20	0,6	=	438,1	19 607	83 976	450,3	96 000	180 000	2,78	
				≠	394,5	37 980	106 160	414,3	96 000	180 000	5,02	
	0,8		=	457,7	27 458	85 288	484,0	96 000	180 000	5,75		
			≠	406,0	63 062	103 300	443,3	96 000	180 000	9,19		
	30	0,6	=	398,0	33 544	75 176	403,6	144 000	180 000	1,41		
			≠	359,3	64 559	86 379	372,7	144 000	180 000	3,73		
		0,8	=	409,1	42 575	76 658	424,4	144 000	180 000	3,74		
			≠	366,3	89 548	75 521	389,3	144 000	180 000	6,28		
		100	10	0,6	=	701,3	6 432	119 826	716,2	64 000	320 000	2,12
					≠	629,1	11 546	152 151	658,1	64 000	320 000	4,61
	0,8			=	791,2	7 656	130 198	819,8	64 000	320 000	3,61	
				≠	689,0	23 802	183 251	743,1	64 000	320 000	7,85	
	25			0,6	=	520,3	18 877	100 741	525,3	160 000	320 000	0,96
					≠	467,6	35 096	123 416	486,1	160 000	320 000	3,96
0,8			=	546,7	24 799	97 288	567,2	160 000	320 000	3,75		
			≠	481,3	66 007	123 100	521,5	160 000	320 000	8,35		
50	0,6		=	436,3	50 395	73 523	446,8	160 000	160 000	2,41		
			≠	398,5	85 416	87 626	414,2	160 000	160 000	3,94		
	0,8		=	447,0	57 099	76 444	465,2	160 000	160 000	4,07		
			≠	402,9	117 308	74 211	429,2	160 000	160 000	6,53		

Tabelle 6.5: Vergleich der Positionierungsverfahren mit integrierten Vermittlungsknoten

Problemgröße		Parameter		SA-basiert			TS-basiert			Δz [%]
$ N $	$ S $	λ	Γ	\bar{z} [KE]	$\bar{\beta}_S$	$\bar{\beta}_B$	\bar{z} [KE]	$\bar{\beta}_S$	$\bar{\beta}_B$	
50	5	0,6	=	470,4	6 059	82 506	536,8	0	160 000	14,12
			≠	428,3	10 560	102 956	483,5	0	160 000	12,89
		0,8	=	527,1	8 904	86 375	617,4	0	160 000	17,13
			≠	472,6	22 264	113 633	551,2	0	160 000	16,63
	10	0,6	=	374,0	13 917	68 858	431,8	0	160 000	15,45
			≠	342,0	25 069	80 134	393,9	0	160 000	15,18
		0,8	=	400,7	18 677	71 669	479,0	0	160 000	19,54
			≠	364,4	37 980	67 890	430,9	0	160 000	18,25
	25	0,6	=	306,0	40 172	31 061	337,0	0	160 000	10,13
			≠	280,6	44 163	17 712	306,8	0	160 000	9,34
		0,8	=	312,1	44 162	28 916	352,4	0	160 000	12,91
			≠	283,8	28 887	5 519	317,7	0	160 000	11,95
75	10	0,6	=	524,8	8 687	89 908	609,4	0	270 000	16,12
			≠	481,1	15 640	111 067	556,4	0	270 000	15,65
		0,8	=	576,2	12 465	91 436	692,7	0	270 000	20,22
			≠	519,5	32 093	121 391	623,9	0	270 000	20,10
	20	0,6	=	423,4	22 499	76 023	492,2	0	270 000	16,25
			≠	390,7	39 775	83 617	451,5	0	270 000	15,56
		0,8	=	446,1	28 074	73 809	537,1	0	270 000	20,40
			≠	408,8	56 543	66 178	487,5	0	270 000	19,25
	30	0,6	=	387,3	38 378	56 188	438,9	0	270 000	13,32
			≠	356,8	63 356	57 521	402,6	0	270 000	12,84
		0,8	=	399,9	46 370	53 722	466,5	0	270 000	16,65
			≠	367,5	52 234	25 726	424,0	0	270 000	15,37
100	10	0,6	=	678,5	6 543	110 014	786,0	0	320 000	15,84
			≠	618,5	11 196	131 655	719,4	0	320 000	16,31
		0,8	=	758,6	9 038	120 745	907,2	0	320 000	19,59
			≠	682,3	24 392	158 987	822,5	0	320 000	20,55
	25	0,6	=	496,9	21 016	87 386	580,7	0	320 000	16,86
			≠	457,8	39 769	104 722	535,4	0	320 000	16,95
		0,8	=	523,6	27 183	85 415	635,3	0	320 000	21,33
			≠	479,2	60 368	90 895	580,0	0	320 000	21,04
	50	0,6	=	425,7	55 110	48 329	475,9	0	320 000	11,79
			≠	395,8	71 524	36 877	439,0	0	320 000	10,91
		0,8	=	435,9	57 383	45 316	497,9	0	320 000	14,22
			≠	401,8	51 166	14 287	455,4	0	320 000	13,34

Zusammenfassend ist das SA-basierte Verfahren bei uneingeschränkten Probleminstanzen auf jeden Fall wegen der höheren Lösungsqualität bei gleichzeitig geringerem Berechnungsaufwand zu bevorzugen. Diese Erkenntnis wird auch in den Untersuchungen in [14] bestätigt.

Bei diesem Vergleich wird quasi ein deterministisches mit einem stochastischen Verfahren verglichen. Eine zufällige Auswahl der Veränderungsschritte, zufällige Veränderungsschritte und eine zufällige Akzeptanz von Lösungen scheint zu besseren Ergebnissen zu führen. Anscheinend werden bei einem systematischen Vorgehen wie beim TS-basierten Verfahren gewisse, Erfolg versprechende Möglichkeiten nicht betrachtet. Diese Erkenntnis legt die Vermutung nahe, dass bei Optimierungsproblemen mit sehr großen Lösungsräumen stochastische Verfahren geeigneter erscheinen. Ähnliches wurde auch bei der Entwicklung und Bewertung der Verfahren zur Erzeugung einer Nachbarschaftslösung (siehe Abschnitt 5.2.2.1 Seite 81) beobachtet. Systematische Verfahren haben im Vergleich zu den stochastischen überwiegend schlechter abgeschnitten.

In Abschnitt 4.1.2 wurde erwähnt, dass bei integrierten Vermittlungsknoten ein geringeres Optimierungspotential im Vergleich zu beliebig positionierbaren Vermittlungsknoten besteht, da die Anzahl möglicher Positionen und folglich der Lösungsraum geringer sind. Werden jedoch die vom SA-basierten Verfahren erreichten Zielfunktionswerte aus den Tabellen 6.4 und 6.5 miteinander verglichen, so ist erkennbar, dass teilweise mit integrierten Vermittlungsknoten bessere Lösungen (geringere Leitungskosten) erreicht werden. Dieses *Phänomen* wurde vom Autor et al. bereits in [2] beschrieben. Die Ursache ist, dass in beiden Untersuchungen ungefähr die gleiche Anzahl der Nachbarschaftslösungen betrachtet wird, wobei der Lösungsraum für integrierte Vermittlungsknoten wesentlich geringer ist. Folglich wird in den Untersuchungen mit integrierten Vermittlungsknoten ein wesentlich größerer Teil des Lösungsraumes durchsucht. Die Ergebnisse aus den Tabellen 6.4 und 6.4 dürfen also nicht miteinander verglichen werden.

Neben dem SA-basierten Positionierungsverfahren wurde auch, wie in Abschnitt 5.2.2.1 erwähnt, ein Sintflut- und Toleranzschwellwert-basiertes Verfahren entwickelt und implementiert. In [12] werden diese beiden Verfahren mit dem SA-basierten Verfahren hinsichtlich ihrer Leistungsfähigkeit verglichen und evaluiert. Es wird gezeigt, dass das Toleranzschwellwert-basierte Verfahren ähnlich gute Lösungen wie das SA-basierte Verfahren findet. Hingegen schnitt das Sintflut-basierte Verfahren in allen Untersuchungen wesentlich schlechter ab.

6.3 Vernetzungsverfahren

Die positionierten Knoten und Bündelungspunkte müssen kostengünstig miteinander vernetzt werden. Hierfür wurden in Abschnitt 5.3 mehrere Verfahren entwickelt, welche im Folgenden evaluiert werden. Zu diesem Zweck wird für jede Probleminstanz eine vorgegebene Anzahl von Knoten und Bündelungspunkten in der Ebene mit der Größe 100×100 Längeneinheiten zufällig positioniert und dann mit dem ausgewählten Verfahren eine kostengünstige Vernetzungsstruktur ermittelt. Die Positionen sowohl der Knoten als auch der Bündelungspunkte werden währenddessen nicht verändert. Zudem werden keine positionsabhängigen Kosten berücksichtigt. Als Distanzmetrik wird also die Manhattan Distanz verwendet.

Für jede Problem Instanz wird eine zufällig erzeugte Verkehrsmatrix \mathbf{R} erstellt. Die Verkehrsanforderungen $r_{i,j}$ zwischen den beiden Endknoten mit dem Index i und j werden nach dem Prinzip der Peer-to-Peer-Kommunikation (siehe Abschnitt 2.3) erzeugt. Hierfür wird der Verkehrsanforderungsgenerator, der vom Autor in [1] veröffentlicht wurde, verwendet. Dieser erzeugt eine gegebene Anzahl zufälliger Kommunikationsbeziehungen. Für jede Beziehung wird ein Sender $n_i \in N$ und ein Empfänger $n_j \in N$ zufällig gleichverteilt ausgewählt. Der Verkehrsanforderungsgenerator stellt zudem sicher, dass Sender und Empfänger nicht identisch sind ($i \neq j$). Die Verkehrsanforderung bzw. das Verkehrsaufkommen zwischen Sender und Empfänger wird nach einer negativ-exponentiellen Verteilungsfunktion (arithmetischer Mittelwert μ) ausgewürfelt und zu dem entsprechenden Element $r_{i,j}$ in der Verkehrsanforderungsmatrix \mathbf{R} dazu addiert. Um die Verkehrsanforderungen möglichst *real* abzubilden, wird bei einer ausgewürfelten Anforderung > 1 erneut gewürfelt. Denn eine derartige Anforderung würde bedeuten, dass bereits ein einzelner Verkehrsfluss eine Leitung überlastet und die zu installierende Technologie wegen ihrer zu geringen Übertragungsrates falsch gewählt worden wäre. Häufig werden vom Empfänger ankommende Nachrichten bestätigt, was eine bidirektionale Kommunikation zur Folge hat. Somit ist die Verkehrsanforderung in Hin- und Rückrichtung asymmetrisch. Um dies zu berücksichtigen, wird in Rückrichtung die Verkehrsanforderung $r_{j,i}$ um 5% der zuvor ausgewürfelten Anforderung erhöht. In allen Untersuchungen zu den Vernetzungsstrukturen wurde die Anzahl der Kommunikationsbeziehungen im Netz gleich der Anzahl der Endknoten gewählt.

6.3.1 Baumförmige Vernetzungsstruktur

In diesem Abschnitt wird untersucht, inwieweit die entwickelten Verfahren zur Verfeinerung einer baumförmigen Vernetzungsstruktur die vom Eröffnungsverfahren erstellte Lösung hinsichtlich der Leitungskosten verbessern können. Ergänzend dazu wird der notwendige Berechnungsaufwand betrachtet.

Bevor das lokale Suchverfahren und das gierige Verfahren miteinander verglichen werden, müssen für das gierige Verfahren geeignete Einstellungen für die beiden Steuerparameter d und β_{iter} bzw. deren Einfluss auf die Lösungsqualität und den Berechnungsaufwand bestimmt werden. Zu diesem Zweck wird eine Problemgröße mit 50 Endknoten, 10 Vermittlungsknoten und 25 Bündelungspunkten gewählt. Weiter wird der Mittelwert für zufällig erzeugte Verkehrsanforderungen $\mu = 0,1$ und $0,01$ sowie der Kostenfaktor für zusätzliche Leitungssegmente $\lambda = 0,6$ und $0,8$ gewählt.

Die Einflüsse der beiden Steuerparameter sind in den Abbildungen 6.1 und 6.2 dargestellt. Die schwarzen Kurven (von links nach rechts absteigend) zeigen die mittleren Zielfunktionswerte (Leitungskosten) und die roten Kurven (von links nach rechts ansteigend) die mittlere Gesamtanzahl untersuchter Nachbarschaftslösungen $\bar{\beta}_{\Sigma}$. Wie in Abbildung 6.1 erkennbar ist, werden mit einem Wert $\beta_{\text{iter}} > 10$ nur noch geringfügige Verbesserungen erzielt, während der Berechnungsaufwand nahezu linear ansteigt. Dies gilt bis zu einem gewissen Schwellwert auch für die Suchtiefe d . Wie aus Abbildung 6.2 ersichtlich ist, werden bei der betrachteten Problemgröße nur bis zu einer Suchtiefe von $d = 35$ Verbesserungen erzielt. Eine Suchtiefe > 35 ist in diesem Beispiel also nicht sinnvoll. In Zeile 9 in Algorithmus 5.7 wird der kleinere Wert aus der Anzahl der Leitungskanäle und der Suchtiefe ausgewählt. Bei einer baumförmigen Vernetzungsstruktur ist die maximale Anzahl der Leitungskanäle gleich $|S| + |B| - 1$; bei dieser

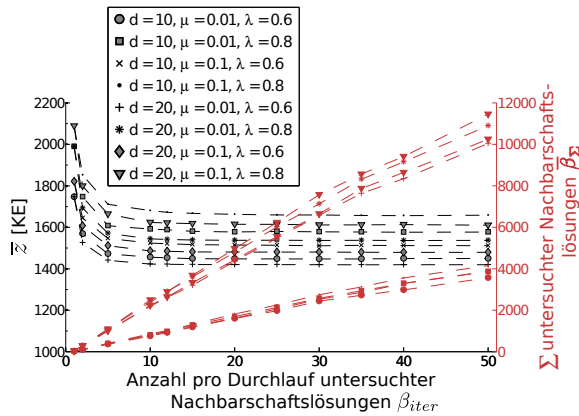


Abbildung 6.1: Einfluss der Anzahl pro Durchlauf zu untersuchender Nachbarschaftslösungen β_{iter} auf die Lösungsqualität und den Berechnungsaufwand (in Gesamtanzahl untersuchter Nachbarschaftslösungen $\bar{\beta}_{\Sigma}$)

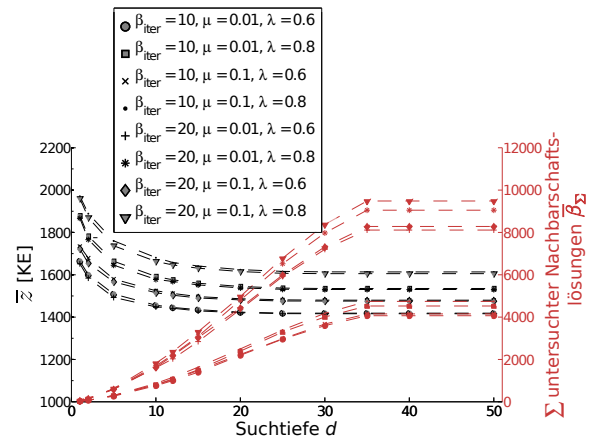


Abbildung 6.2: Einfluss der Suchtiefe d auf die Lösungsqualität und den Berechnungsaufwand (in Gesamtanzahl betrachteter Nachbarschaftslösungen $\bar{\beta}_{\Sigma}$)

Problemgröße folglich 34. Somit ergibt sich folgende geeignete Wahl für die beiden Steuerparameter: $n = 10$ und $d = |S| + |B| - 1$.

Nach der Festlegung der Steuerparameter kann die Leistungsfähigkeit der beiden Verfeinerungsverfahren untersucht werden. Für diese Untersuchungen werden folgende Parameter gewählt: $\lambda = 0,4, 0,6$ und $0,8$ sowie $\mu = 0,01$ und $0,1$.

Die Ergebnisse sind in Tabelle 6.6 (und Tabelle B.1 im Anhang) dargestellt. Die mittleren Zielfunktionswerte \bar{z} (mittlere Leitungskosten) in der Spalte *Eröffnungsverfahren* entsprechen dem initial erstellten MST (in Algorithmus 5.6 und 5.7 Zeile 1). Die Spalte $\bar{\beta}_{\Sigma}$ enthält die mittlere Anzahl betrachteter Nachbarschaftsbäume. In der Spalte Δz werden die prozentualen Differenzen der mittleren Zielfunktionswerte zwischen dem gierigen Verfahren und dem lokalen Suchverfahren berechnet, wobei die mittleren Zielfunktionswerte des lokalen Suchverfahrens als Grundwert angenommen werden. Im Anhang in der Tabelle B.1 sind weitere Ergebnisse aufgeführt. Diesen Untersuchungen liegt eine heterogene, zufällig erzeugte Kostenmatrix mit Autokorrelationsstruktur und dem arithmetischen Mittelwert $\bar{\gamma} = 1$ der Elemente der Matrix Γ zugrunde.

Wie aus Tabelle 6.6 (und Tabelle B.1 im Anhang) ersichtlich, wird die vom Eröffnungsverfahren erstellte Ausgangslösung durch die Verfeinerungsverfahren deutlich verbessert. Des Weiteren nimmt der Berechnungsaufwand mit der Problemgröße zu. Allerdings fällt dieser Einfluss beim gierigen Verfahren wesentlich schwächer aus als beim lokalen Suchverfahren. Aus diesem Grund ist das gierige Verfahren auch für große Netze praxistauglich. Das Verfahren untersucht beispielsweise bei Probleminstanzen mit $|N| = 100$, $|S| = 25$, $|B| = 50$, $\mu = 0,1$ und $\lambda = 0,8$ lediglich einen Bruchteil der Lösungen (ca. 2,5% bzw. 2,8% unter Berücksichtigung einer Kostenmatrix) und erzielt dabei im Mittel nur um 2,61% bzw. 1,87% schlechtere Lösungen. In den

Tabelle 6.6: Vergleich der Verfahren zur Erstellung einer baumförmigen Vernetzungsstruktur

Problemgröße			Parameter		Eröffnungs- verfahren	Lokales Suchver- fahren	Gieriges Verfahren		Δz	
$ N $	$ S $	$ B $	μ	λ	\bar{z} [KE]	\bar{z} [KE]	$\bar{\beta}_z$	\bar{z} [KE]	$\bar{\beta}_z$	[%]
50	10	10	0,01	0,4	1 394,3	1 314,0	4 659	1 314,0	937	0,00
				0,6	1 536,7	1 408,5	5 247	1 408,9	1 055	0,03
				0,8	1 679,1	1 499,3	5 731	1 500,3	1 178	0,07
			0,1	0,4	1 448,6	1 362,0	5 094	1 362,6	1 053	0,04
				0,6	1 618,1	1 475,8	5 690	1 477,0	1 216	0,08
				0,8	1 787,6	1 583,2	6 178	1 586,0	1 330	0,18
50	10	25	0,01	0,4	1 503,6	1 290,9	42 957	1 293,1	3 431	0,17
				0,6	1 746,4	1 414,4	49 463	1 417,9	4 060	0,25
				0,8	1 989,3	1 528,3	53 823	1 534,6	4 519	0,41
			0,1	0,4	1 553,5	1 332,5	42 326	1 336,9	3 577	0,33
				0,6	1 821,3	1 473,2	47 364	1 479,3	4 145	0,41
				0,8	2 089,1	1 601,4	52 060	1 612,9	4 736	0,72
75	10	25	0,01	0,4	2 004,4	1 764,5	45 408	1 766,9	3 570	0,14
				0,6	2 349,3	1 964,3	50 540	1 968,1	4 182	0,19
				0,8	2 694,1	2 154,3	56 478	2 159,8	4 787	0,26
			0,1	0,4	2 095,6	1 848,9	43 426	1 853,3	3 692	0,24
				0,6	2 486,1	2 083,8	48 765	2 092,8	4 345	0,43
				0,8	2 876,5	2 308,7	52 703	2 323,8	4 814	0,65
75	25	35	0,01	0,4	1 627,5	1 408,3	257 002	1 412,3	8 000	0,28
				0,6	1 820,9	1 496,7	282 419	1 500,9	9 192	0,28
				0,8	2 014,4	1 576,8	308 715	1 583,9	10 325	0,45
			0,1	0,4	1 745,3	1 499,8	243 750	1 507,7	8 478	0,53
				0,6	1 997,6	1 622,9	273 825	1 637,5	9 824	0,90
				0,8	2 249,9	1 738,0	291 158	1 761,4	10 697	1,35
100	25	25	0,01	0,4	1 899,8	1 741,2	131 926	1 741,4	5 674	0,01
				0,6	2 099,5	1 851,9	150 768	1 853,2	6 542	0,07
				0,8	2 299,2	1 956,0	165 569	1 958,6	7 424	0,13
			0,1	0,4	2 079,9	1 884,6	130 754	1 888,4	6 233	0,20
				0,6	2 369,6	2 052,2	148 552	2 064,1	7 247	0,58
				0,8	2 659,3	2 210,4	158 790	2 229,7	8 046	0,87
100	25	50	0,01	0,4	2 056,9	1 723,4	635 126	1 730,0	13 986	0,38
				0,6	2 370,2	1 860,1	715 569	1 870,4	16 073	0,55
				0,8	2 683,5	1 986,0	772 443	2 002,3	17 812	0,82
			0,1	0,4	2 230,9	1 862,4	596 240	1 883,3	14 186	1,12
				0,6	2 631,2	2 056,6	640 899	2 091,8	16 080	1,71
				0,8	3 031,5	2 237,2	677 256	2 295,5	17 309	2,61

meisten Fällen ist die Differenz zwischen den beiden mittleren Zielfunktionswerten sogar $< 1\%$. Dies gilt sowohl ohne Berücksichtigung einer Kostenmatrix (Manhattan Distanz) als auch mit einer Kostenmatrix.

Auch wenn der Berechnungsaufwand bei der Optimierung von Topologien eingebetteter Kommunikationsnetze nur eine untergeordnete Rolle spielt, so ist doch in Anbetracht dessen, dass bei den Positionierungsverfahren für jede neue Nachbarschaftslösung auch eine neue kostengünstige Vernetzungsstruktur erstellt werden muss, das gierige Verfahren zu bevorzugen. In Fällen, in denen die Positionen der Knoten und Bündelungspunkte festgelegt sind und lediglich eine sehr kostengünstige Vernetzungsstruktur gesucht wird, kann das lokale Suchverfahren angewandt werden.

6.3.2 Ausfallsichere Vernetzungsstruktur

In diesem Abschnitt wird nun untersucht, inwieweit die entwickelten Verfeinerungsverfahren bzw. Auswahlstrategien zur Erstellung einer ausfallsicheren Vernetzungsstruktur die vom Eröffnungsverfahren erstellte Lösung hinsichtlich der Leitungskosten verbessern können. Ergänzend dazu wird der notwendige Berechnungsaufwand betrachtet.

Vor der Evaluierung der beiden in Abschnitt 5.3.2.2 beschriebenen Auswahlstrategien müssen geeignete Einstellungen für die Steuerparameter bestimmt werden. Für beide Strategien können sowohl die maximale Anzahl aufeinander folgender Iterationen ohne Verbesserung $mIter$ als auch die Anzahl der Ersetzungen pro detektiertem Kreis d eingestellt werden. Zur Bestimmung einer geeigneten Parametrisierung wird erneut eine Problemgröße mit 50 Endknoten, 10 Vermittlungsknoten und 25 Bündelungspunkten gewählt. Weiter wird der Kostenfaktor für zusätzliche Leitungssegmente $\lambda = 0,8$ sowie der Mittelwert für die zufällig erzeugten Verkehrsanforderungen $\mu = 0,1$ gewählt.

Die Einflüsse der beiden Steuerparameter für die beiden Auswahlstrategien sind in den Abbildungen 6.3 bis 6.6 dargestellt. Die Ergebnisse zeigen, dass bereits das Eröffnungsverfahren, welches im Vermittlungsnetz einen 2-MST erstellt und die Endknoten mit möglichst kurzen Leitungskanälen anbindet, Lösungen mit einer hohen Qualität (niedrigen Leitungskosten) findet. Immerhin werden mit den Verfeinerungsverfahren nur Verbesserungen von $< 5\%$ erzielt.

Wie in den Abbildungen 6.3 und 6.4 erkennbar, werden bei mehr als 15 Ersetzungen pro detektiertem Kreis nur geringfügige Verbesserungen erzielt. Interessant ist bei einer zufälligen Auswahlstrategie der Effekt, dass insbesondere im Bereich $d = [0,10]$ eine größere Kantentiefe nicht zwangsläufig zu besseren Ergebnissen führt. Die Ursache dafür ist, dass sich bei unterschiedlichen Werten bereits nach der ersten Iteration die Vernetzungsstrukturen, d. h. Zwischenlösungen, unterscheiden und folglich ein anderer Weg durch den Lösungsraum durchlaufen wird. Bei der zufälligen Auswahlstrategie führt eine größere Anzahl von Ersetzungen pro detektiertem Kreis also nicht zwangsläufig zu besseren Ergebnissen. Mit einer größeren Anzahl ist die Wahrscheinlichkeit höher, dass eine bessere Lösung im Vergleich zu einer geringeren Anzahl gefunden wird.

Auch bei der maximalen Anzahl aufeinander folgender Iterationen ohne Verbesserung $mIter$ führt bei beiden Auswahlstrategien eine größere Anzahl, wie aus den Abbildungen 6.5 und 6.6

ersichtlich wird, nicht zwangsläufig zu besseren Ergebnissen. Jedoch kann es bei mehr als 50 Iterationen nur zu marginalen Verschlechterungen kommen. Des Weiteren wird bei mehr als 25 Iterationen die Ausgangslösung immer verbessert.

Wie im Abschnitt 5.3.2.2 bereits erwähnt, legt der Steuerparameter $mIter$ die maximale Anzahl aufeinander folgender Iterationen ohne Verbesserung fest. Dies bedeutet, dass bei einer Verbesserung die Anzahl bisher untersuchter Kreise auf null zurückgesetzt wird. Folglich ist die Anzahl tatsächlich untersuchter Kreise $\sum iter \geq mIter$. In den Abbildungen 6.7 und 6.8 sind die Anzahl tatsächlich untersuchter Kreise in Abhängigkeit von $\sum iter$ dargestellt. Erkennbar ist, dass bei der gierigen Auswahlstrategie die Anzahl tatsächlich untersuchter Kreise nur geringfügig größer als $mIter$ ist. Folglich werden nach den ersten Iterationen keine weiteren Verbesserungen erzielt. Hingegen ist bei der zufälligen Auswahlstrategie die Anzahl tatsächlich untersuchter Kreise ungefähr doppelt so groß wie $mIter$. Es werden also durch die zufällige Auswahl häufiger Verbesserungen erzielt und somit die Anzahl bisher untersuchter Kreise auf null zurückgesetzt.

Auf der Grundlage der soeben diskutierten Ergebnisse werden für die Bewertung des Verfeinerungsverfahrens mit gieriger und zufälliger Auswahlstrategie der Steuerparameter $d = 15$ gewählt. Für einen fairen Vergleich sollte die Anzahl tatsächlich untersuchter Kreise ungefähr gleich groß sein. Aus diesem Grund wird für das gierige Verfahren $mIter = 100$ und für das zufällige Verfahren $mIter = 50$ gewählt.

Nach der Festlegung der Steuerparameter kann die Leistungsfähigkeit der beiden Auswahlstrategien untersucht werden. Für diese Untersuchungen werden erneut folgende Parameter gewählt: $\lambda = 0,4, 0,6$ und $0,8$ sowie $\mu = 0,01$ und $0,1$.

Die Ergebnisse sind in Tabelle 6.7 (und Tabelle B.2 im Anhang) dargestellt. Die mittleren Zielfunktionswerte \bar{z} (mittlere Leitungskosten) in der Spalte *Eröffnungsverfahren* entsprechen der initial erstellten Vernetzungsstruktur (siehe Abschnitt 5.3.2.1). Die Spalten $\Sigma iter$ enthalten die mittlere Anzahl tatsächlich durchgeführter Iterationen. In der Spalte Δz werden erneut die prozentualen Differenzen der mittleren Zielfunktionswerte (mittlere Leitungskosten) zwischen dem gierigen Verfahren und dem lokalen Suchverfahren berechnet, wobei die mittleren Zielfunktionswerte der zufälligen Auswahlstrategie als Grundwert angenommen werden. Im Anhang sind in Tabelle B.2 weitere Ergebnisse aufgeführt. Diesen liegen Untersuchungen mit einer heterogenen, zufällig erzeugten Kostenmatrix mit Autokorrelationsstruktur und dem arithmetischen Mittelwert $\bar{\gamma} = 1$ der Elemente der Matrix Γ zugrunde.

Wie aus Tabelle 6.7 (und Tabelle B.2 im Anhang) ersichtlich, wird die vom Eröffnungsverfahren erstellte Ausgangslösung durch die Verfeinerungsverfahren stets verbessert. Die Beobachtung, die bereits bei den Untersuchungen zu den Einflüssen der Steuerparameter gemacht wurde, dass die Verfeinerungsverfahren nur Verbesserungen von $< 5\%$ erzielen, bestätigt sich auch bei dieser umfangreichen Untersuchung. Das zeigt erneut, dass bereits das Eröffnungsverfahren Lösungen mit einer hohen Qualität findet. Zwischen den beiden Auswahlstrategien ist kein signifikanter Unterschied hinsichtlich der mittleren Zielfunktionswerte zu erkennen. Die prozentuale Differenz ist stets $< 1\%$. In den meisten Fällen findet die zufällige Auswahlstrategie bessere Lösungen. Aufgrund der geringeren Anzahl tatsächlich durchgeführter Iterationen und somit des geringeren Berechnungsaufwandes ist die zufällige Auswahlstrategie sicherlich zu bevorzugen.

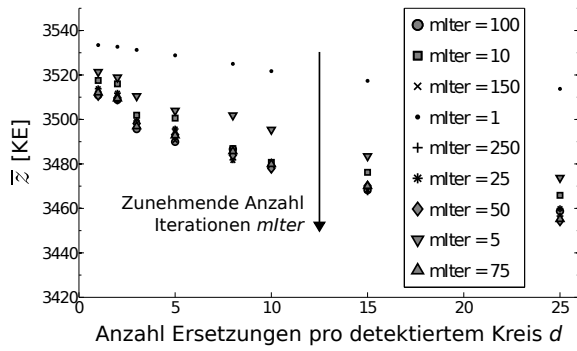


Abbildung 6.3: Einfluss der Anzahl der Ersetzungen pro detektiertem Kreis auf die Lösungsqualität bei der gierigen Auswahlstrategie

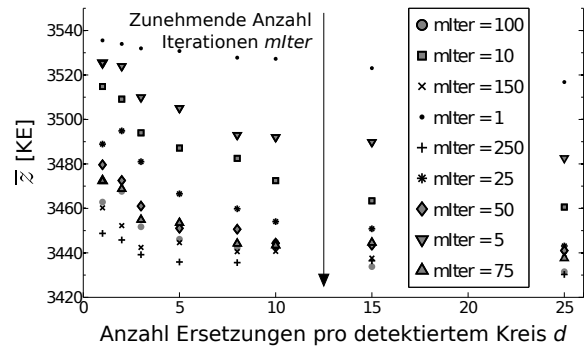


Abbildung 6.4: Einfluss der Anzahl der Ersetzungen pro detektiertem Kreis auf die Lösungsqualität bei der zufälligen Auswahlstrategie

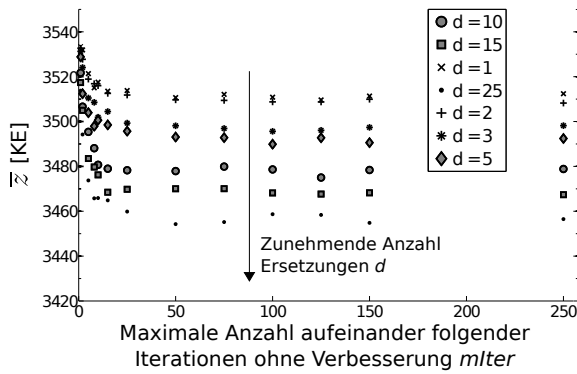


Abbildung 6.5: Einfluss der maximalen Anzahl aufeinander folgender Iterationen ohne Verbesserung auf die Lösungsqualität bei der gierigen Auswahlstrategie

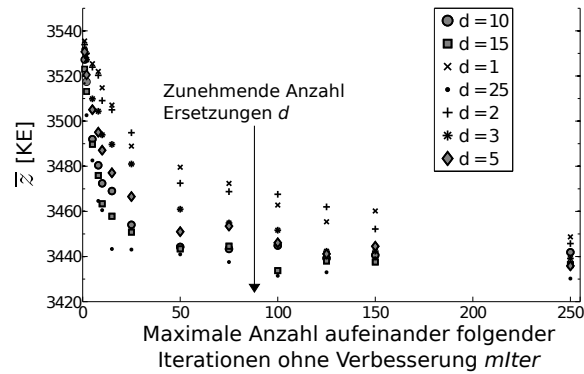


Abbildung 6.6: Einfluss der maximalen Anzahl aufeinander folgender Iterationen ohne Verbesserung auf die Lösungsqualität bei der zufälligen Auswahlstrategie

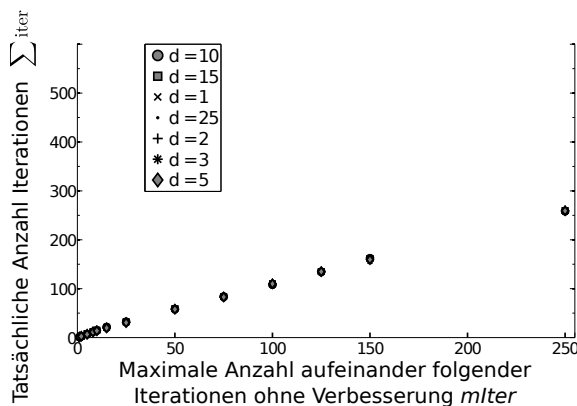


Abbildung 6.7: Anzahl tatsächlich durchgeführter Iterationen in Abhängigkeit von der maximalen Anzahl aufeinander folgender Iterationen ohne Verbesserung bei der gierigen Auswahlstrategie

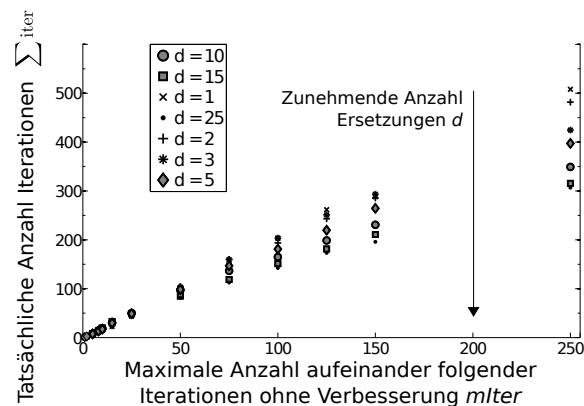


Abbildung 6.8: Anzahl tatsächlich durchgeführter Iterationen in Abhängigkeit von der maximalen Anzahl aufeinander folgender Iterationen ohne Verbesserung bei der zufälligen Auswahlstrategie

Tabelle 6.7: Vergleich der Verfahren zur Erstellung einer ausfallsicheren Vernetzungsstruktur

Problemgröße			Parameter		Eröffnungs- verfahren \bar{z} [KE]	Auswahlstrategie		Verfeinerungsverfahren		Δz [%]
$ N $	$ S $	$ B $	μ	λ		Zufällig \bar{z} [KE]	Σ iter	Gierig \bar{z} [KE]	Σ iter	
50	10	10	0,01	0,4	2 609,0	2 576,6	75,3	2 580,7	112,1	0,16
				0,6	2 824,2	2 782,3	79,7	2 787,1	113,8	0,17
				0,8	3 039,3	2 985,9	83,4	2 996,1	115,2	0,34
			0,1	0,4	2 716,0	2 682,9	72,6	2 688,9	108,9	0,22
				0,6	2 984,6	2 936,8	71,6	2 942,3	109,9	0,19
				0,8	3 253,2	3 177,8	79,0	3 194,4	107,9	0,52
50	10	25	0,01	0,4	2 613,2	2 583,5	76,9	2 588,2	112,5	0,18
				0,6	2 964,2	2 920,1	83,4	2 932,1	112,6	0,41
				0,8	3 315,2	3 260,1	85,7	3 273,0	110,9	0,40
			0,1	0,4	2 725,3	2 689,9	78,2	2 698,8	106,7	0,33
				0,6	3 132,4	3 068,3	83,8	3 087,7	108,9	0,63
				0,8	3 539,3	3 443,4	85,1	3 468,2	110,1	0,72
75	10	25	0,01	0,4	3 690,6	3 662,7	82,8	3 665,8	110,8	0,08
				0,6	4 226,3	4 186,5	86,3	4 195,4	113,8	0,21
				0,8	4 762,0	4 701,2	91,7	4 719,5	115,8	0,39
			0,1	0,4	3 881,9	3 840,0	83,2	3 851,3	108,8	0,29
				0,6	4 513,3	4 438,0	85,1	4 461,1	109,4	0,52
				0,8	5 144,6	5 020,9	90,5	5 061,7	109,4	0,81
75	25	35	0,01	0,4	2 694,7	2 658,1	108,7	2 649,8	161,1	-0,31
				0,6	2 942,6	2 894,5	121,0	2 890,4	164,3	-0,14
				0,8	3 190,4	3 124,3	141,7	3 126,6	151,9	0,07
			0,1	0,4	2 937,6	2 893,9	99,9	2 893,1	138,0	-0,03
				0,6	3 306,9	3 231,4	112,5	3 234,7	145,7	0,10
				0,8	3 676,2	3 564,4	115,2	3 573,0	143,7	0,24
100	25	25	0,01	0,4	3 358,1	3 329,2	109,5	3 317,4	154,5	-0,35
				0,6	3 633,5	3 590,6	131,9	3 587,5	149,7	-0,09
				0,8	3 908,9	3 850,7	136,6	3 851,0	150,3	0,01
			0,1	0,4	3 714,0	3 667,5	94,5	3 669,5	131,4	0,05
				0,6	4 167,3	4 087,5	109,2	4 095,6	138,9	0,20
				0,8	4 620,7	4 491,2	119,8	4 507,8	143,4	0,37
100	25	50	0,01	0,4	3 419,4	3 388,0	108,2	3 382,0	150,3	-0,18
				0,6	3 838,3	3 793,1	126,5	3 790,9	153,6	-0,06
				0,8	4 257,2	4 191,2	143,3	4 192,2	166,7	0,02
			0,1	0,4	3 786,6	3 736,8	115,6	3 740,0	134,0	0,09
				0,6	4 389,1	4 296,0	127,1	4 309,4	145,2	0,31
				0,8	4 991,6	4 832,0	148,2	4 866,0	147,4	0,70

7 Anwendung der Verfahren

In diesem Kapitel wird die Anwendungsreife der neu entwickelten Verfahren gezeigt. Es werden unter Anwendung der Verfahren exemplarische, praxisrelevante Fragestellungen beantwortet. Gleichzeitig werden die Mächtigkeit, Anwendungsreife und Flexibilität verdeutlicht. Im ersten Teil wird für ein eingebettetes Kommunikationsnetz die maximale Anzahl sinnvoller Vermittlungsknoten und Bündelungspunkte bestimmt. Im zweiten Teil werden eine baumförmige und eine ausfallsichere Vernetzungsstruktur hinsichtlich ihrer Kosten miteinander verglichen. Im letzten Teil werden dann für zwei Problem instanzen die Kompromisslösungen bestimmt.

7.1 Maximalausstattung

In dieser Untersuchung wird experimentell die Frage beantwortet, ob für ein baumförmig strukturiertes Kommunikationsnetz eine sinnvolle Maximalausstattung benannt werden kann. Das soll heißen, ob ähnlich zur maximalen Anzahl von Steinerpunkten auch eine maximale Anzahl sinnvoller Vermittlungsknoten und Bündelungspunkte existiert. Um diese Frage zu beantworten, werden die beiden Verfahren zum Entfernen von Vermittlungsknoten und Bündelungspunkten aus Abschnitt 5.2.3 angewandt.

Für die Untersuchung werden wiederum 100 zufällig erzeugte Problem instanzen unterschiedlicher Größen betrachtet. Initial werden eine große Anzahl von Vermittlungsknoten und Bündelungspunkten zufällig platziert. Die initiale Anzahl der Vermittlungsknoten $|S|$ und Bündelungspunkte $|B|$ wird abhängig von der Anzahl der Endknoten $|N|$ gewählt: $|S| = 1$ und $|S| = |N|$ sowie $|B| = 0$ und $|B| = |N|$. Mit dem Wert $|S| = 1$ soll die maximale Anzahl sinnvoller Bündelungspunkte unabhängig von der Anzahl der Vermittlungsknoten untersucht werden. Umgekehrt soll mit dem Wert $|B| = 0$ die maximale Anzahl sinnvoller Vermittlungsknoten unabhängig von der Anzahl der Bündelungspunkte untersucht werden. Mit der Kombination $|S| = |N|$ und $|B| = |N|$ soll untersucht werden, ob ein Zusammenhang zwischen einer maximalen Anzahl sinnvoller Vermittlungsknoten und einer maximalen Anzahl sinnvoller Bündelungspunkte existiert.

Nachdem die Vermittlungsknoten und Bündelungspunkte zufällig platziert wurden, werden diese und die Endknoten nach den in den Abschnitten 5.3.1.1 und 5.3.1.2 entwickelten Verfahren miteinander vernetzt (Kombination aus MST im Vermittlungsnetz und möglichst kurze Leitungskanäle für die Anbindung der Endknoten). Anschließend werden zuerst die irrelevanten Bündelungspunkte und dann die irrelevanten Vermittlungsknoten entfernt. Es sei nochmals darauf hingewiesen, dass das Entfernen von Bündelungspunkten eher defensiv erfolgt und dadurch unter Umständen nicht alle irrelevanten Bündelungspunkte entfernt werden.

Tabelle 7.1: Anzahl sinnvoller Vermittlungsknoten und Bündelungspunkte

Initiale Problemgröße			Parameter λ	Anzahl Vermittlungsknoten			Anzahl Bündelungspunkte			Anzahl Summe		
$ N $	$ S $	$ B $		min	mean	max	min	mean	max	min	mean	max
25	25	0	*	8	12,9	17	0	0,0	0	8	12,9	17
			0,2	9	13,8	18	1	5,6	17	12	19,4	32
			0,8	10	14,0	18	1	4,0	12	12	18,0	29
	1	25	0,2	1	1,0	1	7	10,6	17	8	11,6	18
			0,8	1	1,0	1	4	9,1	20	5	10,1	21
50	50	0	*	20	26,3	31	0	0,0	0	20	26,3	31
			0,2	21	28,1	41	5	12,5	32	30	40,6	73
			0,8	21	28,4	41	3	9,9	34	24	38,3	75
	1	50	0,2	1	1,0	1	16	22,7	31	17	23,7	32
			0,8	1	1,0	1	13	20,7	32	14	21,7	33
75	75	0	*	31	38,8	47	0	0,0	0	31	38,8	47
			0,2	33	41,7	55	9	18,9	32	43	60,6	76
			0,8	33	42,3	52	3	14,7	33	41	56,9	76
	1	75	0,2	1	1,0	1	25	34,7	48	26	35,7	49
			0,8	1	1,0	1	22	32,4	55	23	33,4	56
100	100	0	*	41	52,3	60	0	0,0	0	41	52,3	60
			0,2	47	55,7	69	13	24,9	39	66	80,6	103
			0,8	50	56,7	66	4	20,4	34	58	77,1	96
	1	100	0,2	1	1,0	1	38	48,9	73	39	49,9	74
			0,8	1	1,0	1	32	45,5	74	33	46,5	75

In dieser Untersuchung werden keine Verkehrsanforderungen berücksichtigt. Es wird angenommen, dass die maximale Übertragungsrate der gewählten Technologie ausreichend für die Verkehrsanforderungen ist und keine zusätzlichen parallelen Leitungen installiert werden müssen.

Die Ergebnisse zur Bestimmung der Maximalausstattung sind in Tabelle 7.1 aufgeführt. In dieser Tabelle sind neben den arithmetischen Mittelwerten (mean) auch die Minimal- (min) und Maximalwerte (max) über die 100 zufällig erzeugten Probleminstanzen enthalten. Falls keine Bündelungspunkte ($|B| = 0$) installiert werden, wird niemals ein Bündelungsgewinn erzielt und das Ergebnis ist unabhängig vom Kostenfaktor λ . Bei allen betrachteten Probleminstanzen ohne Bündelungspunkte ist die maximale Anzahl sinnvoller Vermittlungsknoten kleiner $|N| - 2$. Das lässt den Schluss zu, dass ähnlich zur maximalen Anzahl von Steinerpunkten auch eine maximale Anzahl sinnvoller Vermittlungsknoten existiert. Das Gleiche gilt für die maximale Anzahl sinnvoller Bündelungspunkte. Selbst wenn durch einen geringen Kostenfaktor für zusätzliche Leitungssegmente ($\lambda = 0,2$) ein hoher Bündelungsgewinn erzielt werden könnte, ist die maximale Anzahl sinnvoller Bündelungspunkte bei allen betrachteten Probleminstanzen kleiner $|N| - 2$. Jedoch gilt diese Regel nicht für die Summe aus der Anzahl sinnvoller Vermittlungsknoten und Bündelungspunkte. Der Maximalwert der Summe ist stets größer als die Anzahl der Endknoten. Allerdings ist der Mittelwert der Summe stets kleiner als die Anzahl der Endknoten.

Trotzdem existiert keine Regel für die maximale Summenanzahl ähnlich zur maximalen Anzahl von Steinerpunkten. Interessant ist allerdings auch, dass der Durchschnitts- und Maximalwert der Summe nahezu unabhängig vom Kostenfaktor λ ist. Sowohl für $\lambda = 0,2$ als auch für $\lambda = 0,8$ ergeben sich ähnliche Ergebnisse.

7.2 Kostenvergleich Vernetzungsstrukturen

In Bereichen, in denen bei Ausfall Menschenleben gefährdet sind, kann nicht abgewogen werden, ob sich eine ausfallsichere Vernetzungsstruktur *rechnet*. Anders hingegen sieht es in Bereichen aus, in denen ein Ausfall einfach nur Kosten, beispielsweise durch einen Produktionsstillstand, verursacht. In diesen Bereichen wird häufig eine Wertanalyse zur Abwägung, ob ein Kommunikationsnetz ausfallsicher ausgelegt werden soll, durchgeführt. Bei einer solchen Analyse wird das Verhältnis Nutzen zu Aufwand berechnet. Falls dieses Verhältnis größer als 1 ist, ist es sinnvoll, das Kommunikationsnetz ausfallsicher auszulegen. Allerdings müssen für eine Wertanalyse die zusätzlichen Kosten einer ausfallsicheren Vernetzungsstruktur bestimmt werden. In diesem Abschnitt wird gezeigt, wie diese zusätzlichen Kosten (Mehrkosten) mit den entwickelten Vernetzungsverfahren bestimmt werden und als Grundlage einer Wertanalyse dienen können.

In Tabelle 7.2 werden die mittleren Zielfunktionswerte (mittlere Leitungskosten) für eine baumförmige \bar{z}_B (nicht ausfallsichere) und eine ausfallsichere \bar{z}_A Vernetzungsstruktur für unterschiedliche Probleminstanzen miteinander verglichen. Die Ergebnisse sind den in Abschnitt 6.3 durchgeführten Untersuchungen entnommen. Es sind also arithmetische Mittelwerte aus 100 zufällig erzeugten Probleminstanzen. Für die Kosten einer baumförmigen Vernetzungsstruktur wurden die Ergebnisse, die mit dem gierigen Verfahren (siehe Tabelle 6.6) bestimmt wurden, herangezogen. Für die Kosten einer ausfallsicheren Vernetzungsstruktur wurden die Ergebnisse, die mit der zufälligen Auswahlstrategie (siehe Tabelle 6.7) bestimmt wurden, herangezogen. In der letzten Spalte der Tabelle 7.2 sind die Teuerungsquotienten \bar{z}_A/\bar{z}_B für eine ausfallsichere Vernetzungsstruktur aufgelistet.

Aus dem Vergleich in Tabelle 7.2 wird ersichtlich, dass eine ausfallsichere Vernetzungsstruktur ungefähr doppelt so teuer ist im Vergleich zu einer baumförmigen. Dieser Quotient ist *quasi* unabhängig von der Problemgröße, dem Kostenfaktor λ und den mittleren Verkehrsanforderungen μ .

Obwohl eine ausfallsichere Vernetzungsstruktur ungefähr doppelt so teuer ist, ergibt sich eine erhebliche Kostenersparnis im Vergleich zu einer vollständig redundanten Installation. Bei der optimierten Lösung sichern sich die Vermittlungsknoten gegenseitig ab und es sind keine zusätzlichen Vermittlungsknoten notwendig. Hingegen erhöhen sich bei einer vollständig redundanten Installation auch die Kosten für die Vermittlungsknoten.

Zur genaueren Betrachtung der Mehrkosten einer ausfallsicheren Vernetzungsstruktur werden diese in Tabelle 7.3 auf das Vermittlungsnetz \bar{z}_V und auf das Zugangsnetz \bar{z}_Z aufgeteilt. Falls ein Leitungskanal sowohl zur redundanten Vernetzung der Vermittlungsknoten als auch zur redundanten Anbindung der Endknoten genutzt wird, muss definiert werden, welche Anteile der Fixkosten des Leitungskanals auf den jeweiligen Teil entfallen. Im Folgenden werden die

Tabelle 7.2: Kostenvergleich baumförmiger und ausfallsicherer Vernetzungsstrukturen

Problemgröße			Parameter		Baumförmig	Ausfallsicher	
$ N $	$ S $	$ B $	μ	λ	\bar{z}_B [KE]	\bar{z}_A [KE]	\bar{z}_A/\bar{z}_B
50	10	10	0,01	0,4	1 314,0	2 576,6	1,96
				0,6	1 408,5	2 782,3	1,98
				0,8	1 499,3	2 985,9	1,99
			0,1	0,4	1 362,0	2 682,9	1,97
				0,6	1 475,8	2 936,8	1,99
				0,8	1 583,2	3 177,8	2,01
50	10	25	0,01	0,4	1 290,9	2 583,5	2,00
				0,6	1 414,4	2 920,1	2,06
				0,8	1 528,3	3 260,1	2,13
			0,1	0,4	1 332,5	2 689,9	2,02
				0,6	1 473,2	3 068,3	2,08
				0,8	1 601,4	3 443,4	2,15
75	10	25	0,01	0,4	1 764,5	3 662,7	2,08
				0,6	1 964,3	4 186,5	2,13
				0,8	2 154,3	4 701,2	2,18
			0,1	0,4	1 848,9	3 840,0	2,08
				0,6	2 083,8	4 438,0	2,13
				0,8	2 308,7	5 020,9	2,17
75	25	35	0,01	0,4	1 408,3	2 658,1	1,89
				0,6	1 496,7	2 894,5	1,93
				0,8	1 576,8	3 124,3	1,98
			0,1	0,4	1 499,8	2 893,9	1,93
				0,6	1 622,9	3 231,4	1,99
				0,8	1 738,0	3 564,4	2,05
100	25	25	0,01	0,4	1 741,2	3 329,2	1,91
				0,6	1 851,9	3 590,6	1,94
				0,8	1 956,0	3 850,7	1,97
			0,1	0,4	1 884,6	3 667,5	1,95
				0,6	2 052,2	4 087,5	1,99
				0,8	2 210,4	4 491,2	2,03
100	25	50	0,01	0,4	1 723,4	3 388,0	1,97
				0,6	1 860,1	3 793,1	2,04
				0,8	1 986,0	4 191,2	2,11
			0,1	0,4	1 862,4	3 736,8	2,01
				0,6	2 056,6	4 296,0	2,09
				0,8	2 237,2	4 832,0	2,16
Mittlerer Teuerungsquotient \triangleright							2,03

Tabelle 7.3: Vergleich der Kostenanteile von Vermittlungsnetz \bar{z}_V und Zugangsnetz \bar{z}_Z bei einer ausfallsicheren Vernetzungsstruktur

Problemgröße			Parameter		\bar{z}_A [KE]	\bar{z}_Z [KE]	\bar{z}_V [KE]	\bar{z}_V/\bar{z}_Z
$ N $	$ S $	$ B $	μ	λ				
50	10	10	0,01	0,4	2 576,6	2 289,9	286,7	7,99
				0,6	2 782,3	2 483,9	298,4	8,32
				0,8	2 985,9	2 676,8	309,1	8,66
			0,1	0,4	2 682,9	2 286,2	396,7	5,76
				0,6	2 936,8	2 482,0	454,8	5,46
				0,8	3 177,8	2 676,1	501,7	5,33
50	10	25	0,01	0,4	2 583,5	2 304,1	279,4	8,25
				0,6	2 920,1	2 621,3	298,8	8,77
				0,8	3 260,1	2 937,2	322,9	9,10
			0,1	0,4	2 689,9	2 298,1	391,8	5,87
				0,6	3 068,3	2 619,0	449,3	5,83
				0,8	3 443,4	2 936,4	507,0	5,79
75	10	25	0,01	0,4	3 662,7	3 383,0	279,7	12,10
				0,6	4 186,5	3 881,7	304,8	12,74
				0,8	4 701,2	4 379,1	322,1	13,60
			0,1	0,4	3 840,0	3 374,9	465,1	7,26
				0,6	4 438,0	3 877,7	560,3	6,92
				0,8	5 020,9	4 377,7	643,2	6,81
75	25	35	0,01	0,4	2 658,1	2 139,2	518,9	4,12
				0,6	2 894,5	2 348,0	546,5	4,30
				0,8	3 124,3	2 555,8	568,5	4,50
			0,1	0,4	2 893,9	2 126,4	767,5	2,77
				0,6	3 231,4	2 340,8	890,6	2,63
				0,8	3 564,4	2 552,4	1 012,0	2,52
100	25	25	0,01	0,4	3 329,2	2 804,2	525,0	5,34
				0,6	3 590,6	3 044,2	546,4	5,57
				0,8	3 850,7	3 283,2	567,5	5,79
			0,1	0,4	3 667,5	2 789,6	877,9	3,18
				0,6	4 087,5	3 035,6	1 051,9	2,89
				0,8	4 491,2	3 279,5	1 211,7	2,71
100	25	50	0,01	0,4	3 388,0	2 868,1	519,9	5,52
				0,6	3 793,1	3 235,7	557,4	5,80
				0,8	4 191,2	3 601,7	589,5	6,11
			0,1	0,4	3 736,8	2 850,2	886,6	3,21
				0,6	4 296,0	3 225,6	1 070,4	3,01
				0,8	4 832,0	3 597,2	1 234,8	2,91

Fixkosten eines von mehreren Leitungssegmenten genutzten Leitungskanals entsprechend den Anteilen des Vermittlungs- und des Zugangsnetzes aufgeteilt. Dies bedeutet, dass bei gleicher Anzahl von Leitungssegmenten die Fixkosten je zur Hälfte dem Vermittlungsnetz und dem Zugangsnetz zugeordnet werden. In [13] werden weitere Möglichkeiten für die Aufteilung der Fixkosten diskutiert.

In der letzten Spalte der Tabelle 7.3 sind die Quotienten aus \bar{z}_v/\bar{z}_z aufgelistet. Daraus wird ersichtlich, dass die Mehrkosten einer ausfallsicheren Vernetzungsstruktur im Wesentlichen durch die Absicherung des Zugangsnetzes entstehen, wohingegen die Ausfallsicherheit im Vermittlungsnetz wesentlich *günstiger* ist. Die Kosten für eine ausfallsichere Anbindung der Endknoten sind in den meisten Fällen um ein Vielfaches höher als die Kosten für das ausfallsichere Vermittlungsnetz. Im Vermittlungsnetz sind anscheinend nur wenige zusätzliche Leitungen und Leitungskanäle notwendig, um eine ausfallsichere Vernetzungsstruktur herzustellen bzw. es können bestehende Leitungskanäle weitestgehend genutzt werden. Während beim Zugangsnetz für die Anbindung eines Endknotens zum bestehenden Pfad zum nächsten (kostengünstigsten) Vermittlungsknoten noch ein zweiter, längerer Pfad bestehend aus zusätzlichen Leitungen und Leitungskanälen installiert werden muss. Der zusätzliche Pfad scheint eben entsprechend *teuer* zu sein.

Für den Entwurf eingebetteter Kommunikationsnetze kann aus dieser Untersuchung der Schluss gezogen werden, dass Ausfallsicherheit im Vermittlungsnetz nur geringe Mehrkosten verursacht, während für die ausfallsichere Anbindung der Endknoten erheblich höhere Kosten anfallen. Daher ist es sinnvoll, für jeden Endknoten eine Einzelfallentscheidung zu treffen, ob eine ausfallsichere Anbindung überhaupt zwingend erforderlich ist. Endknoten aufgrund ihres geforderten Konnektivitätsgrades zu klassifizieren, ist also aus Wirtschaftlichkeitsgründen durchaus sinnvoll.

7.3 Bestimmung einer Kompromisslösung

Im letzten Teil dieses Kapitels wird für eine Probleminstance mit 35 Endknoten und für eine mit 50 Endknoten mittels vollständiger Enumeration (siehe Abschnitt 5.4) der Zielkonflikt zwischen Minimierung der Anzahl der Vermittlungsknoten und Minimierung der Leitungskosten aufgelöst und eine Kompromisslösung bestimmt. Genau wie bei der Untersuchung zur Maximalausstattung wird angenommen, dass die maximale Übertragungsrate der gewählten Technologie ausreichend für die Verkehrsanforderungen ist und daher diese nicht berücksichtigt werden müssen. Die Vermittlungsknoten können beliebig positioniert und müssen nicht in den Endknoten integriert werden. Des Weiteren wird der Kostenfaktor $\lambda = 0,8$ gewählt und es werden keine positionsabhängigen Kosten berücksichtigt. Als Kosten für einen Vermittlungsknoten werden folgende Werte angenommen: $c_s = 25$ KE und $c_v = 100$ KE.

Zur kostengünstigen Positionierung der Vermittlungsknoten und Bündelungspunkte wird das SA-basierte Verfahren mit folgendem Cooling Schedule verwendet: $\tau = 1,0$, $p = 0,9$, $m_{\text{Iter}} = 100$, $m_{\text{Atmp}} = 100$, $m_{\text{Impr}} = 25$ und initiale Anzahl der Bündelungspunkte $|N| - |S| - 2$ (siehe Algorithmus 5.2). Aus Laufzeitgründen wird das gierige Verfahren (siehe Abschnitt 5.3.1.3) zur Erstellung einer kostengünstigen baumförmigen Vernetzungsstruktur gewählt und wie folgt parametrisiert: $d = 5$ und $\beta_{\text{iter}} = 5$.

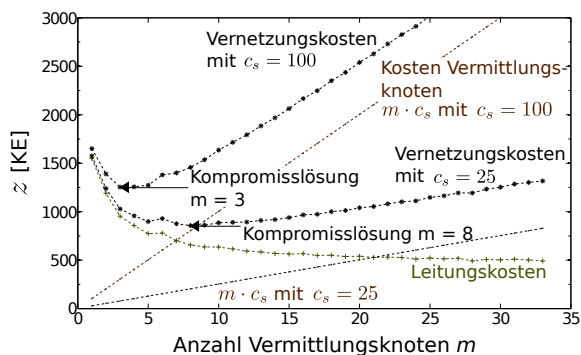


Abbildung 7.1: Leitungskosten, Kosten für die Vermittlungsknoten und Vernetzungskosten für eine Problem Instanz mit 35 Endknoten in Abhängigkeit von der Anzahl der Vermittlungsknoten m

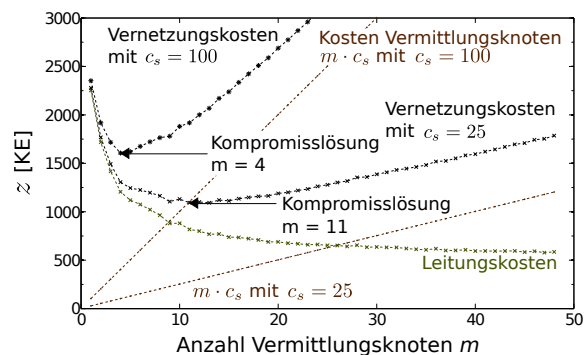


Abbildung 7.2: Leitungskosten, Kosten für die Vermittlungsknoten und Vernetzungskosten für eine Problem Instanz mit 50 Endknoten in Abhängigkeit von der Anzahl der Vermittlungsknoten m

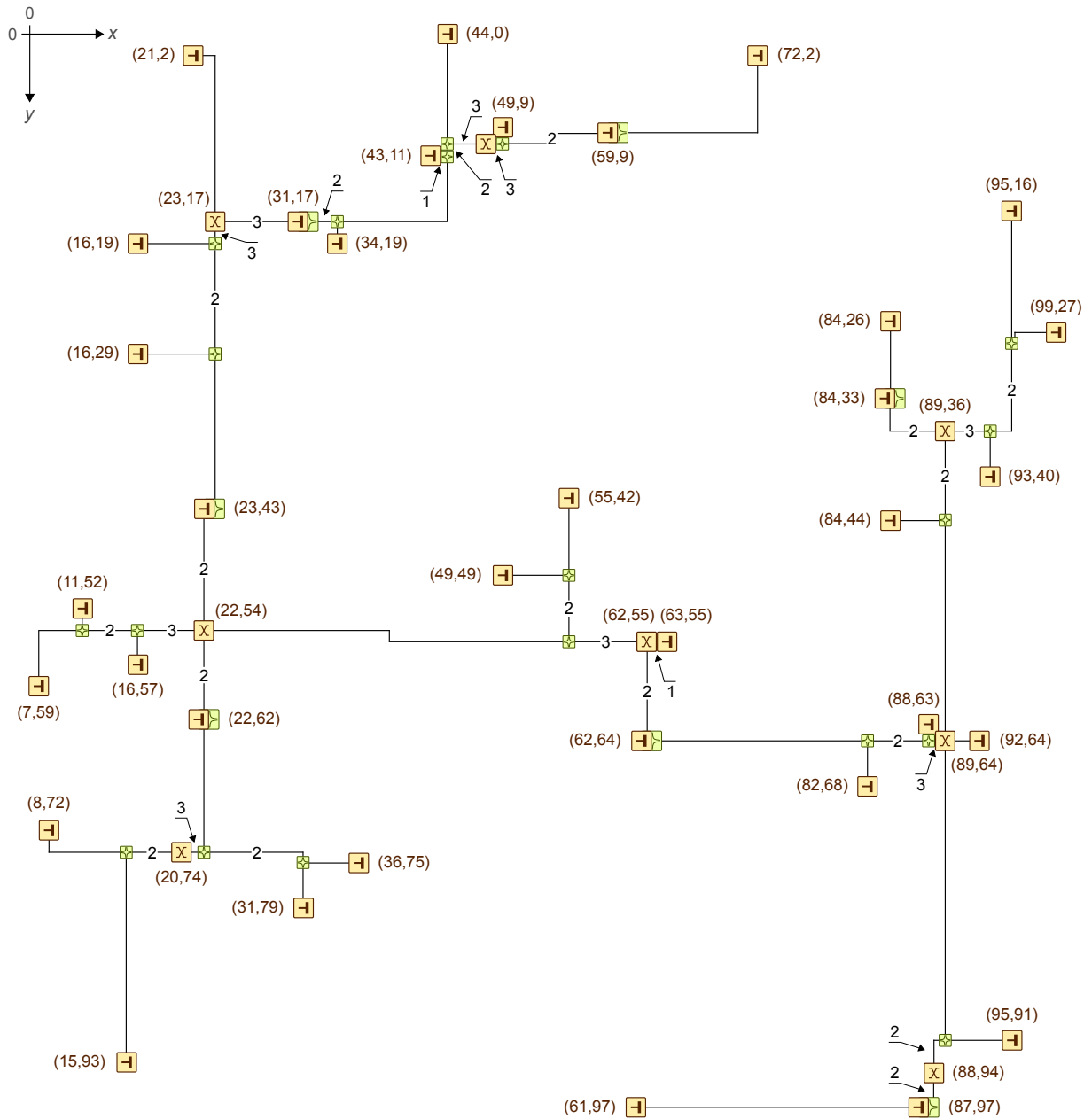
In den Abbildungen 7.1 und 7.2 sind die Leitungskosten, die Kosten für die Vermittlungsknoten und die Vernetzungskosten in Abhängigkeit von der Anzahl der Vermittlungsknoten m dargestellt. Diese Abbildungen verdeutlichen nochmals den Zusammenhang zwischen der Anzahl der Vermittlungsknoten und den Leitungskosten. Mit zunehmender Anzahl von Vermittlungsknoten sinken die Leitungskosten und umgekehrt.

Die Kompromisslösung ist, wie aus Gleichung (5.20) und den oben dargestellten Abbildungen hervorgeht, auch abhängig von den Kosten für einen Vermittlungsknoten. Je teurer ein Vermittlungsknoten ist, desto geringer ist die Anzahl der Vermittlungsknoten in der Kompromisslösung. In Abbildung 7.1 besteht bei $c_s = 100$ KE die Kompromisslösung aus 3 Vermittlungsknoten, wohingegen sie bei $c_s = 25$ KE aus 8 Vermittlungsknoten besteht. Bei der Problem Instanz mit 50 Endknoten besteht die Kompromisslösung aus 4 bzw. 11 Vermittlungsknoten.

In den Abbildungen 7.1 und 7.2 ist auch erkennbar, dass die Leitungskosten mit zunehmender Anzahl von Vermittlungsknoten nicht monoton fallend sind. Die Ursache hierfür liegt in den Heuristiken. Diese finden zwar Lösungen, die sehr nahe am Optimum sind, jedoch eben nicht optimal. Allerdings sind die *Ausreißer* sehr gering, so dass mit der vollständigen Enumeration durchaus eine *eindeutige* heuristische Kompromisslösung bestimmt werden kann.

In Abbildung 7.3 ist zur Veranschaulichung die Kompromisslösung für eine Problem Instanz mit 35 Endknoten und $c_s = 25$ KE dargestellt. In der abgebildeten Lösung, die letztendlich aus 26 Bündelungspunkten besteht, wurden irrelevante Bündelungspunkte bereits entfernt. In Abbildung 7.3 sind auch die Koordinaten der End- und Vermittlungsknoten enthalten. Bei den Endknoten mit Leitungsbündelung wurde ein Bündelungspunkt genau, d. h. identische Koordinaten, beim Endknoten positioniert. Die Leitungskosten betragen 653 KE und die Kosten für die Vermittlungsknoten $8 \cdot 25 = 200$ KE. Folglich betragen die Vernetzungskosten 853 KE.

Zusammenfassend wurde in dieser Untersuchung unter Anwendung eines Positionierungsverfahrens, eines Verfahrens zur Erstellung einer kostengünstigen Vernetzungsstruktur und mittels









- | | | | |
|---|-------------------------------------|---|---|
|  | Endknoten |  | Vermittlungsknoten |
|  | Bündelungspunkt |  | Endknoten mit Leitungsbündelung |
|  | Leitungskanal mit 1 Leitungssegment |  | Leitungskanal mit $\varphi = x$ Leitungssegmenten |

Abbildung 7.3: Kompromisslösung für eine Problem Instanz mit 35 Endknoten und $c_s = 25$ KE

vollständiger Enumeration eine Kompromisslösung bestimmt und somit das aufgezeigte multi-kriterielle Optimierungsproblem gelöst.

8 Zusammenfassung und Ausblick

Zusammenfassung Die Vernetzung eingebetteter Systeme gestaltet technische Systeme und Prozesse effizienter und sicherer und ermöglicht die Realisierung innovativer, verteilter Dienste. In dieser Arbeit wurde diskutiert, welche Vorteile Ethernet als Technologie zur Vernetzung eingebetteter Systeme hat. Diese sind im Wesentlichen Kosteneinsparungen, standardisierte Erweiterungen, Technologierobustheit, Flexibilität sowie die weite Verbreitung. Die in Kapitel 2.7 aufgezeigten Entwicklungen und Trends lassen den Schluss zu, dass Ethernet weitere Anwendungsbereiche durchdringen und derzeit eingesetzte (Feld-) Bussysteme mittel- und längerfristig ablösen wird.

Heutige Ethernet-Netze bestehen aus Vermittlungsknoten und aus Vollduplexleitungen. Dadurch werden zwar Kollisionen vermieden und die Leistungsfähigkeit gesteigert, jedoch führt dies auch zu zusätzlichen Kosten. Deshalb müssen für einen attraktiven Einsatz von Ethernet die Vernetzungskosten, d. h. die Leitungskosten und die Kosten für die Vermittlungsknoten, minimiert werden. Dazu muss die Topologie des Netzes kostenoptimal gestaltet werden. Hierzu zählen die optimale Positionierung der Vermittlungsknoten und eine optimale Vernetzungsstruktur. Zur Gestaltung einer optimalen Vernetzungsstruktur müssen zudem Bündelungspunkte, an denen Leitungen gebündelt werden, kostenoptimal positioniert werden.

In dieser Arbeit wurden mehrere Verfahren zur Kostenoptimierung von Topologien eingebetteter Kommunikationsnetze entwickelt. Zuvor wurden in Kapitel 2 die Grundlagen und Anwendungsbereiche eingebetteter Kommunikationsnetze behandelt und in Kapitel 3 notwendige Grundlagen der Topologieoptimierung eingeführt.

Nach den Grundlagen wurden in Kapitel 4 das Optimierungsproblem umfassend beschrieben, geeignet modelliert und eine Zielfunktion formuliert. Bei der Modellierung und Formulierung der Zielfunktion wurden auch die Eigenschaften eingebetteter Kommunikationsnetze entsprechend berücksichtigt. Zudem erfolgte eine Abschätzung der Komplexität des Optimierungsproblems und der Größe des Lösungsraumes.

Zur Minimierung der Vernetzungskosten müssen sowohl die Leitungskosten als auch die Kosten für die Vermittlungsknoten minimiert werden. Hierfür wurden in Kapitel 5 folgende Verfahren entwickelt:

- **Positionierungsverfahren** Ein exaktes und zwei heuristische Verfahren zur kostengünstigen Positionierung von Vermittlungsknoten und Bündelungspunkten:
 - (i) Exaktes Lösungsverfahren für eingeschränkte Probleminstanzen basierend auf einem Mixed Integer Linear Program (MILP)

- (ii) Heuristische Lösungsverfahren für uneingeschränkte Probleminstanzen basierend auf Simulated Annealing (SA) und Tabu Search (TS)
- **Vernetzungsverfahren** Verfahren zur Erstellung einer kostengünstigen baumförmigen sowie ausfallsicheren Vernetzungsstruktur. Die Strategie der entwickelten Verfahren ist, eine kostengünstige Vernetzungsstruktur stets in zwei Schritten zu erstellen. Zuerst wird eine zulässige Startlösung bestimmt, welche dann iterativ durch lokale Strukturveränderungen verfeinert wird:
 - (i) Baumförmige Vernetzungsstruktur
 - Eröffnungsverfahren: Minimal Spanning Tree (MST) im Vermittlungsnetz und möglichst kurzen Leitungskanälen zur Anbindung der Endknoten
 - Verfeinerungsverfahren: Lokales Suchverfahren und gieriges Verfahren
 - (ii) Ausfallsichere Vernetzungsstruktur
 - Eröffnungsverfahren: Two-Minimal Spanning Tree (2-MST) im Vermittlungsnetz und möglichst kurzen Leitungskanälen zur redundanten Anbindung der Endknoten
 - Verfeinerungsverfahren: Gierige und zufällige Auswahlstrategie zum Finden von zu verändernden Kreisen
- **Verfahren zur Bestimmung einer Kompromisslösung** Zwischen den Kosten für die Vermittlungsknoten und den Leitungskosten besteht ein Zusammenhang. Je größer die Anzahl installierter Vermittlungsknoten, desto geringer die Leitungskosten und umgekehrt. Die entwickelten Verfahren bestimmen eine Kompromisslösung entweder mittels vollständiger Enumeration oder mittels Intervallschachtelung.

In Kapitel 6 wurden die Positionierungs- und die Vernetzungsverfahren evaluiert. Zu diesem Zweck wurden die Optimierungsverfahren mit gleichem Anwendungszweck miteinander verglichen. Zur Evaluierung der heuristischen Positionierungsverfahren wurden deren gefundene Lösungen mit den optimalen Lösungen, die mit dem exakten Verfahren bestimmt wurden, eingeschränkter Probleminstanzen miteinander verglichen. Bei diesem Vergleich wurde die Leistungsfähigkeit der beiden heuristischen Lösungsverfahren experimentell nachgewiesen. In weiteren Vergleichen zwischen dem SA- und TS-basierten Verfahren mit uneingeschränkten Probleminstanzen wurde gezeigt, dass das SA-basierte Verfahren bessere Lösungen mit höherer Qualität bei vergleichbarem Berechnungsaufwand findet.

Bei den Vernetzungsverfahren wurden zuerst die Verfahren zur Verfeinerung einer baumförmigen und anschließend die zur Verfeinerung einer ausfallsicheren Vernetzungsstruktur evaluiert. Vor den Evaluierungen wurden, falls notwendig, geeignete Einstellungen für die jeweiligen Steuerparameter durchgeführt und diese dabei festgelegt. Die beiden Verfahren *lokales Suchverfahren* und *gieriges Verfahren* zur Verfeinerung einer baumförmigen Vernetzungsstruktur konnten die vom Eröffnungsverfahren erstellte Startlösung deutlich verbessern und fanden überwiegend ähnliche Ergebnisse. Allerdings ist der Berechnungsaufwand, insbesondere bei Probleminstanzen mit vielen Endknoten, Vermittlungsknoten und Bündelungspunkten, beim gierigen Verfahren wesentlich geringer als beim lokalen Suchverfahren. Zwischen den beiden Strategien *Gierig* und *Zufällig* zur Verfeinerung einer ausfallsicheren Vernetzungsstruktur war kein signifikanter Unterschied hinsichtlich der Lösungsqualität zu erkennen. Allerdings ist bei der zufälligen Auswahlstrategie der Berechnungsaufwand deutlich geringer.

In Kapitel 7 wurde die Anwendungsreife der entwickelten Verfahren anhand exemplarischer, praxisrelevanter Fragestellungen gezeigt. So wurden für ein Netz die Maximalausstattung, für Probleminstanzen unterschiedlicher Größen die Mehrkosten für eine ausfallsichere Vernetzungsstruktur und für zwei Probleminstanzen beispielhaft die Kompromisslösung bestimmt.

Mit den in dieser Arbeit entwickelten Verfahren können Topologien eingebetteter Kommunikationsnetze hinsichtlich ihrer Kosten optimiert werden. Die Verfahren können beliebig miteinander kombiniert werden. Falls die Leitungskanäle vorgegeben sind, so können die kostengünstigen Positionen der Vermittlungsknoten bestimmt werden. Umgekehrt kann bei Vorgabe der Vermittlungsknoten und/oder Bündelungspunkte eine kostengünstige Vernetzungsstruktur bestimmt werden. Existieren keine Vorgaben, so kann unter Ausnutzung der *völligen* Freiheit eine kostengünstige Gesamtlösung bestimmt werden.

Ausblick Bei der Bestimmung einer Kompromisslösung werden für alle Vermittlungsknoten identische Kosten angenommen. Es werden nicht die Positionen der Vermittlungsknoten und somit keine Umgebungskosten wie bei den Leitungskosten berücksichtigt. Zur Berücksichtigung der Umgebungskosten müsste lediglich die erstellte Softwarebibliothek zur Kostenoptimierung geringfügig erweitert werden. Wird angenommen, dass die Werte in der Kostenmatrix relative und keine absoluten Kosten repräsentieren, dann müsste lediglich von einem Ausrüster oder Hersteller festgelegt werden, ob sowohl für die Leitungskosten als auch für die Kosten der Vermittlungsknoten die gleiche Kostenmatrix verwendet werden könnte oder hierfür zwei unterschiedliche Matrizen festgelegt werden müssten. Das entwickelte Verfahren zur Bestimmung einer Kompromisslösung wäre nach wie vor uneingeschränkt anwendbar.

Da für alle Vermittlungsknoten identische Kosten angenommen wurden, sind diese unabhängig von der Anzahl der Anschlüsse und dem Funktionsumfang. Allerdings können die Kosten auch durchaus abhängig von der Anzahl der Anschlüsse sein. Ein Vermittlungsknoten mit vielen Anschlüssen wäre in diesem Fall teurer als einer mit wenigen. Durch eine Kostenfunktion der Form $c_S(x) = c_{\text{fix}} + x \cdot c_{\text{Port}}$ könnte diese Eigenschaft berücksichtigt werden, wobei c_{fix} die Fixkosten, beispielsweise für das interne Koppelnetz und die Verwaltungseinheit, x die Anzahl der Anschlüsse und c_{Port} die Kosten für einen Anschluss sind.

In einem baumförmig vernetzten, kollisionsfreien Ethernet-Netz beträgt die Gesamtanzahl der Anschlüsse $2(n + m - 1)$, wobei n die Anzahl der Endknoten und m die Anzahl der Vermittlungsknoten sind. Mit steigender Anzahl von Vermittlungsknoten steigt also die Gesamtanzahl der Anschlüsse. Durch die erweiterte Kostenfunktion $c_S(x)$ könnte dieser Zusammenhang bei der Bestimmung einer Kompromisslösung berücksichtigt werden.

Detailverbesserungen an der erstellten Softwarebibliothek zur Kostenoptimierung könnten die Performanz weiter erhöhen. Ein Ansatzpunkt wäre, die Datenstrukturen, die zur internen Abbildung einer Probleminstanz eingesetzt werden, zu optimieren. Wegen der anhaltenden Zunahme der Anzahl von Rechenkernen in einem Computer und des anhaltenden Preisverfalls von Hardware wäre es sinnvoll, ausgewählte Verfahren zu parallelisieren. Zum einen könnte das Verfahren zur Bestimmung einer Kompromisslösung parallelisiert werden. So könnten Probleminstanzen mit unterschiedlicher Anzahl von Vermittlungsknoten parallel gelöst und somit schneller eine Kompromisslösung gefunden werden. Zum anderen könnten die Nachbarschaftssuchverfahren parallelisiert werden, so dass mehrere Lösungen parallel erstellt und bewertet werden

könnten. Somit könnte in derselben Zeit ein größerer Teil des Lösungsraumes untersucht und eventuell eine bessere Lösung gefunden werden.

Bisher werden bei der Dimensionierung der zu installierenden Leitungen lediglich die mittleren Verkehrsanforderungen berücksichtigt. Wie bereits erwähnt werden jedoch keine weiteren Leistungskriterien berücksichtigt. Sollen weitere Leistungskriterien berücksichtigt werden, könnte die Dimensionierung der Leitungen nach bekannten, verkehrstheoretischen Modellen und Verfahren, wie in [244] behandelt, während der Optimierung erfolgen. Dies sind im Allgemeinen nichtlineare Zusammenhänge zwischen maximaler Übertragungsrate, Auslastungsgrad und Dienstgüte. Allerdings würde dadurch die Optimierung erheblich komplexer und vermutlich könnten Lösungen nicht in akzeptabler Zeit gefunden werden. Deshalb sollte eine Dimensionierung in einem nachgelagerten Schritt erfolgen.

Ein anderer Ansatz wäre eine lose Kopplung zwischen Optimierung und Dimensionierung. So könnte eine optimierte Lösung unter Performanzkriterien bewertet werden. Falls die Lösung gewisse Kriterien nicht erfüllt oder für deren Erfüllung erhebliche Mehrkosten entstehen, könnte diese Lösung bei einer erneuten Optimierung ausgeschlossen werden. Dies würde zu einem iterativen Vorgehen führen, bei dem abwechselnd eine Optimierung unter Berücksichtigung einer Menge ausgeschlossener Lösungen und einer Dimensionierung durchgeführt wird. Dies wäre ähnlich zu Tabu Search, bei dem auch Lösungen mittels einer Tabu-Liste ausgeschlossen werden. Die Erforschung eines methodischen Vorgehens für die Kombination aus Optimierung und Dimensionierung wäre durchaus auch für andere Entwurfsprobleme im Bereich der Kommunikationsnetze interessant.

A Softwarebibliothek zur Kostenoptimierung von Topologien

Im Rahmen dieser Arbeit wurde eine Softwarebibliothek zur Kostenoptimierung von Topologien stufenweise entworfen und implementiert. In diese Bibliothek sind sämtliche in Kapitel 5 entwickelten Verfahren integriert. Im Prinzip kann die Bibliothek für zwei Ziele verwendet werden:

1. **Entwurf und Analyse von Verfahren** zur Kostenoptimierung von Topologien eingebetteter Kommunikationsnetze. Neue Verfahren können unter Verwendung vorhandener Komponenten und Klassen mit geringem Aufwand einfach implementiert werden. Es können Studien durchgeführt und Aussagen über die Leistungsfähigkeit des betrachteten Verfahrens und der Qualität der gefundenen Lösungen getroffen werden. Zudem können auch neue Konfigurationen und Kombinationen von Verfahren analysiert werden.
2. **Optimierung und Analyse von Topologien** zukünftiger eingebetteter Kommunikationsnetze unter Berücksichtigung unterschiedlicher Anforderungen und Randbedingungen. Die Bibliothek ermöglicht den Entwurf kostengünstiger/-optimaler Lösungen. Optimierte Lösungen können in einem nachgelagerten Prozessschritt hinsichtlich der Wirtschaftlichkeit (Kosten) analysiert werden.

Die Bibliothek wurde in den letzten beiden Jahren stetig weiter entwickelt und bereits in mehreren Forschungsarbeiten [1, 2, 5, 12–15] erfolgreich eingesetzt. In diesem Kapitel wird zuerst die Architektur sowie der Funktionsumfang beschrieben. Anschließend wird der Ablauf einer Optimierungsstudie vorgestellt.

A.1 Architektur und Funktionsumfang

Die Architektur der Bibliothek ist in Abbildung A.1 dargestellt. Die Architektur ist vollständig modular gestaltet. Ein Modul umfasst eine abgrenzbare Einheit. Bei der Modularisierung wurde das Ziel einer möglichst geringen Kopplung zwischen den Modulen verfolgt. Die Schnittstellen eines Moduls definieren klare Grenzen innerhalb der Bibliothek. Ein derartig modularer Aufbau erfüllt die Anforderung der Flexibilität hinsichtlich der Erweiterbarkeit und Wiederverwendbarkeit. Neue Funktionen können durch Hinzufügen neuer Module oder nachträgliches Erweitern

bestehender Module implementiert werden. Die starke Modularisierung ermöglicht auch das modulweise Testen und erhöht somit die Korrektheit der Bibliothek.

Beim Entwurf wurde auf die Auswahl und Gestaltung der internen Datenstrukturen besonders Wert gelegt. Für die Attribute zur Repräsentation von Probleminstanzen wurden im Allgemeinen Datenstrukturen gewählt, die effizient genutzt werden können. In der Regel sind dies statische Datenstrukturen, da der Zugriff darauf schneller ist im Vergleich zu dynamischen Datenstrukturen wie verketteten Listen. Die Bibliothek wurde nach objektorientierten Regeln und Mustern entworfen. Sie ist vollständig in Java implementiert und somit weitestgehend plattformunabhängig. Neben der Java Plattform baut die Bibliothek noch auf weiteren frei verfügbaren Klassenbibliotheken auf:

- Die **IKR Simulation Library** (IKR SimLib) [245] ist eine institutseigene Bibliothek zur Simulation von Kommunikationsnetzen. In [7] geben der Autor und Scharf eine umfassende Beschreibung über die Architektur und den Funktionsumfang der IKR SimLib. Für die Bibliothek zur Kostenoptimierung werden insbesondere die Komponenten zur Steuerung von Simulations- und Optimierungsläufen, Ausgabefunktionen sowie die Statistiken der IKR SimLib genutzt.
- Die Bibliothek **Batik** [246] unterstützt das Erzeugen, Bearbeiten und Darstellen von Grafiken. Sie wird genutzt, um Lösungen, die im Visualisierer angezeigt werden, als Grafikdatei im Scalable Vector Graphics Format zu speichern.
- Mit der Bibliothek **JFreeChart** [247] können komplexe Diagramme unterschiedlicher Typen (zum Beispiel Säulen-, Balken-, und Gantt-Diagramme) erstellt werden. Für die Bibliothek werden Klassen zur Visualisierung der Lösungsqualität in Abhängigkeit der Anzahl betrachteter Nachbarschaftslösungen genutzt. Damit kann die Leistungsfähigkeit der Lösungsverfahren visualisiert und analysiert werden.
- Die Bibliothek **JTransforms** [248] bietet Methoden für schnelle Fourier-Transformationen. Diese werden für die Erzeugung einer Matrix mit zufälligen Werten und einer Autokorrelationsstruktur (siehe [5]) zur Repräsentation der positionsabhängigen Kosten benötigt.

Anfangs wurde bereits die modulare Gestaltung erwähnt. In Abbildung A.1 sind die Module dargestellt. Ein Modul erfüllt eine bestimmte Aufgabe und umfasst in der Regel mehrere abgrenzbare Klassen. Die Module sind verschiedenen Domänen zugeordnet. Im Folgenden werden diese Domänen genannt und der Funktionsumfang der darin enthaltenen Module kurz beschrieben.

Basis Darin enthalten sind Module zur zufälligen Erzeugung von Matrizen mit positionsabhängigen Kosten. Verkehrsanforderungen können entweder aus einer externen Datei eingelesen oder zufällig erzeugt werden. Bei letzterem kann die Anzahl der Quellen, die Verkehrscharakteristik und die mittlere Verkehrsanforderung parametrisiert werden. Für die Berechnung der Distanz bzw. der Kosten zwischen zwei Punkten in der Ebene stehen mehrere Metriken zur Verfügung. Häufiger werden bei der Optimierung Zufallszahlen benötigt. Hierfür werden mehrere Zufallszahlengeneratoren angeboten, die Zufallszahlen aus unabhängigen Strömen liefern.

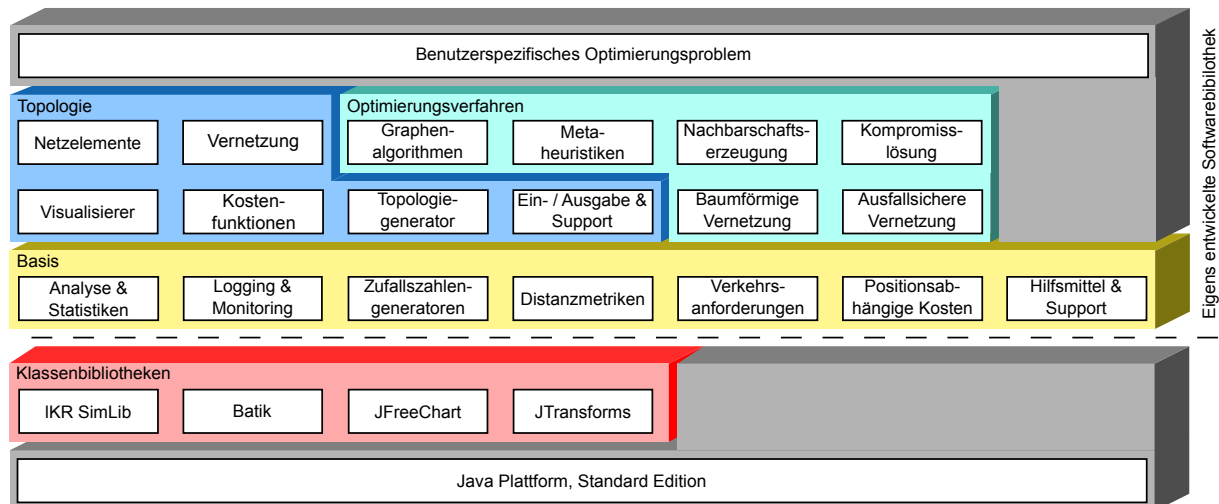


Abbildung A.1: Architektur der Softwarebibliothek

Zur Analyse stehen mehrere Methoden und Statistiken, die unter anderem für benutzerspezifische Optimierungsprobleme initialisiert werden können, bereit. Komponenten zum Logging und Monitoring sowie diverse Hilfsmittel vervollständigen den Umfang der Basismodule.

Topologie Darin enthalten sind die Module zur Repräsentation einer Probleminstance (Topologie), zur Kostenberechnung und ein Visualisierer. Letzterer ermöglicht neben dem Visualisieren (Anzeigen) einer Topologie auch deren interaktives Modifizieren. Dabei können Vermittlungsknoten und Bündelungspunkte entfernt oder an beliebigen Positionen in der Ebene hinzugefügt werden. Nach einer solchen Modifikation wird die Vernetzungsstruktur neu berechnet und aktualisiert. Zudem können mit dem Visualisierer alle verbesserten Lösungen, die während eines Optimierungslaufes gefunden wurden, angezeigt werden. Ein gutes Beispiel für den modularen Aufbau ist das Modul *Kostenfunktionen*. Durch dessen Erweiterung können bei Bedarf mit minimalem Aufwand neuartige oder erweiterte Kostenfunktionen integriert und berücksichtigt werden.

Die Netzelemente umfassen die Grundbausteine *Endknoten*, *Vermittlungsknoten* und *Bündelungspunkt*. Die Vernetzung umfasst die logische Zuordnung der Leitungskanäle zu den Netzelementen sowie die Anzahl physikalischer Leitungsegmente in den Leitungskanälen. Ein Topologiegenerator ermöglicht das zufällige Erzeugen von Topologien mit unterschiedlichen Eigenschaften. Dies ist insbesondere bei der Analyse neuer Verfahren sinnvoll, da die Leistungsfähigkeit eines Verfahrens nicht mit einer einzelnen Probleminstance bewertet werden kann. Eine Ein-/Ausgabekomponente ermöglicht das Schreiben und Lesen einer Topologie aus einer bzw. in eine Datei. Die Ausgabe in eine strukturierte Textform ermöglicht eine Weiterverarbeitung. So kann eine Topologie in unterschiedliche Dateiformate exportiert werden, die andere Bibliotheken unterstützen. Beispielsweise wurden für den Vergleich in Abschnitt 6.2.1 die Probleminstance mit dem Topologiegenerator erzeugt, in Dateien geschrieben und von der Bibliothek GeoSteiner eingelesen und gelöst.

Optimierungsverfahren Diese Domäne beinhaltet die eigentlichen Kernmodule der Bibliothek und enthält sämtliche Verfahren zur Kostenoptimierung. Die implementierten Graphenal-

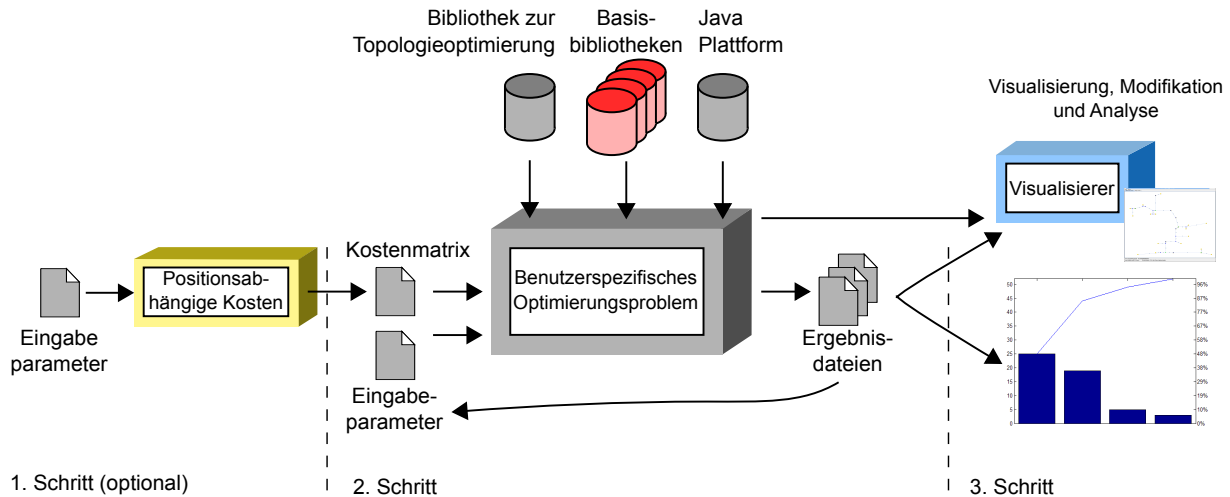


Abbildung A.2: Ablauf einer Optimierungsstudie

gorithmen wie der Algorithmus von Dijkstra, Floyd und Warshall, Suurballe und Prim sind für die in der Bibliothek verwendeten Datenstrukturen optimiert. Die Metaheuristiken sind weitestgehend allgemeingültig und könnten nahezu unverändert auch zur Lösung anderer Optimierungsprobleme verwendet werden. Die Metaheuristiken werden ergänzt durch Verfahren zur Erzeugung von Nachbarschaftslösungen. Hierfür stehen in der Bibliothek neben den in dieser Arbeit entwickelten Verfahren weitere zur Verfügung. An dieser Stelle zeigen sich die Vorteile des modularen Aufbaus. So können verschiedene Metaheuristiken mit beliebigen Verfahren zur Nachbarschaftserzeugung kombiniert und analysiert werden. Zur Bestimmung einer Kompromisslösung, wie in Abschnitt 5.4 beschrieben, wurde ebenfalls ein Modul implementiert. Wahlweise kann eine baumförmige oder ausfallsichere Vernetzungsstruktur erzeugt werden. Das Erzeugen einer Vernetzungsstruktur umfasst neben dem Bestimmen der Leitungskanäle auch die Berechnung benötigter physikalischer Leitungssegmente unter Berücksichtigung von Verkehrsanforderungen.

A.2 Ablauf einer Optimierungsstudie

Nachdem die Architektur und der Funktionsumfang der eigens entwickelten Bibliothek zur Kostenoptimierung beschrieben wurden, wird nun der Ablauf einer Optimierungsstudie vorgestellt. Dieser Ablauf ist auch in Abbildung A.2 dargestellt. Falls ein neues Optimierungsverfahren hinsichtlich seiner Leistungsfähigkeit analysiert werden soll und die positionsabhängigen Kosten der Zielumgebung nicht vorliegen, kann im ersten Schritt eine Matrix mit zufälligen Werten und einer Autokorrelationsstruktur (siehe [5]) zur Repräsentation der positionsabhängigen Kosten erzeugt werden. Das Modul *Positionsabhängige Kosten* bietet hierfür Funktionen an. Über die Eingabeparameter können die Eigenschaften (Größe, Wertebereiche, etc.) der Matrix definiert werden. Der erste Schritt ist optional.

Im zweiten Schritt erfolgt dann die eigentliche Optimierung. Falls keine positionsabhängigen Kosten definiert wurden, wird die Manhattan Distanz als Distanzmetrik verwendet. In Abhängigkeit vom benutzerspezifischen Optimierungsproblem variieren die Eingabeparameter. Wahl-

weise kann festgelegt werden, ob der Bündelungseffekt berücksichtigt werden soll und welchen Wert der Kostenfaktor λ besitzt. Zudem können optional die Verkehrsanforderungen angegeben werden oder ein Generator zur Erzeugung zufälliger Werte parametrisiert werden. Die Positionen der Endknoten können ebenfalls angegeben werden oder es wird lediglich deren Anzahl angegeben. Des Weiteren müssen die Parameter zur Steuerung der Heuristiken (sowohl zur Positionsoptimierung als auch zur Optimierung der Vernetzungsstruktur) und gegebenenfalls gewünschte Verfahren zur Erzeugung einer Nachbarschaftslösung definiert werden. Sämtliche Parameter werden in einer Datei beschrieben und durch einen Parser, der Bestandteil der IKR SimLib ist, eingelesen. Zum Vergleich unterschiedlicher Problemgrößen (beispielsweise Anzahl der Vermittlungsknoten, etc.) können mehrere Optimierungsläufe parallel gestartet und die Ergebnisse anschließend analysiert werden.

Als wichtigstes Ergebnis wird die optimierte Topologie in einer Datei ausgegeben. Zusätzlich können die Topologie auch im Visualisierer angezeigt und modifiziert sowie weitere Informationen (Laufzeit, Statistikergebnisse, etc.) ausgegeben werden. Im dritten Schritt kann mit Hilfe externer Analysemethoden und -werkzeuge die Lösungsqualität und die Leistungsfähigkeit des angewandten Verfahrens bewertet werden. Basierend auf den Erkenntnissen der Optimierung können die Eingabeparameter modifiziert und ein weiterer Optimierungslauf gestartet werden. In ähnlicher Weise arbeitet auch das Modul zum Bestimmen einer Kompromisslösung.

Abschließend ist der Aufbau einer Parameterdatei für eine Optimierungsstudie dargestellt. Es werden die aufgeführten Parameter kurz beschrieben und deren Wertebereich genannt. Es sei angemerkt, dass die dargestellte Parameterdatei aus Gründen der Übersichtlichkeit nicht vollständig ist.

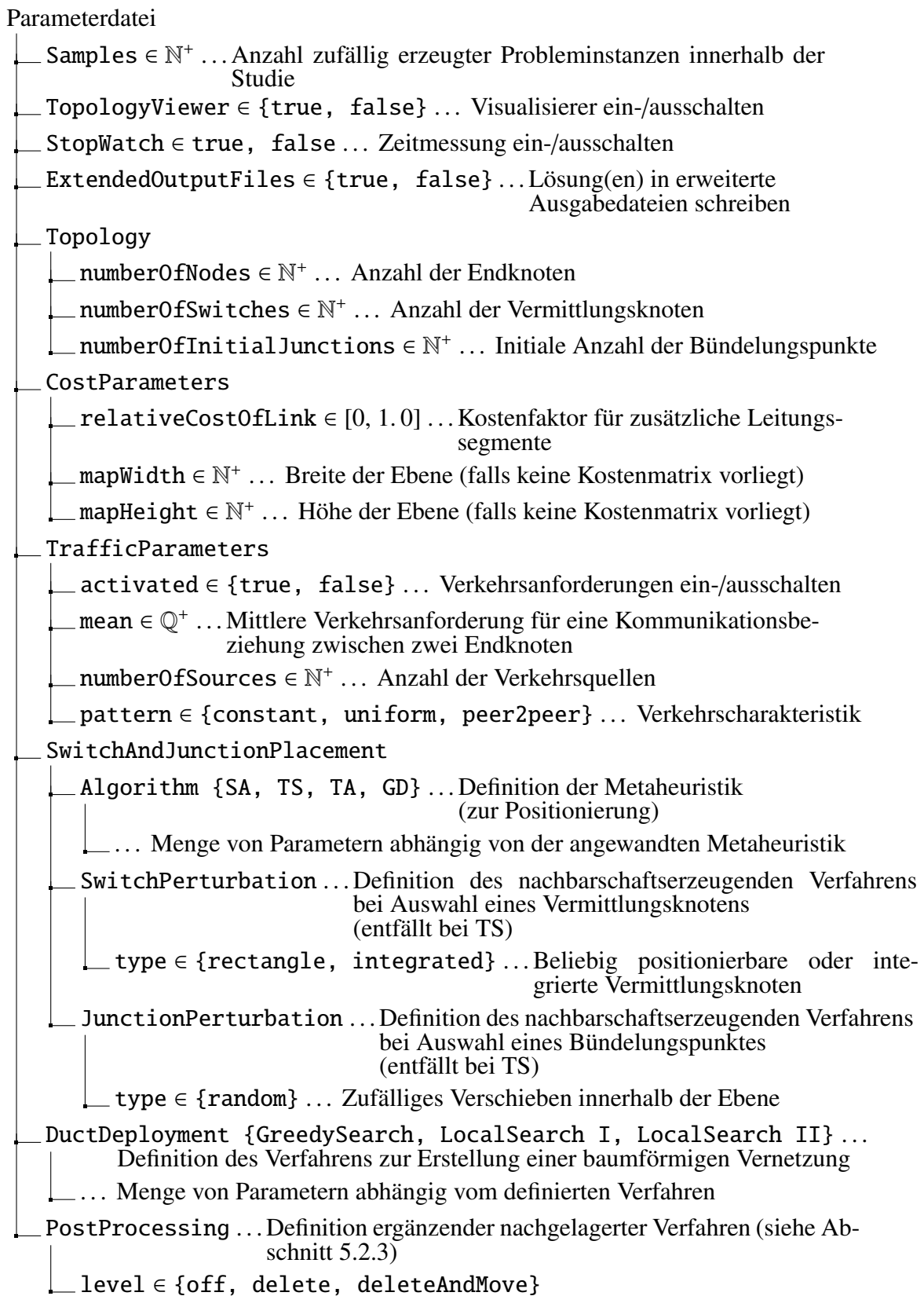


Abbildung A.3: Aufbau einer Parameterdatei

B Ergänzende Ergebnisse

Tabelle B.1: Vergleich der Verfahren zur Erstellung einer baumförmigen Vernetzungsstruktur unter Berücksichtigung einer zufällig erzeugten Kostenmatrix der Größe 100×100

Problemgröße			Parameter		Eröffnungs- verfahren	Lokales Suchver- fahren	Gieriges Verfahren	Δz		
$ N $	$ S $	$ B $	μ	λ	\bar{z} [KE]	\bar{z} [KE]	β_Σ	\bar{z} [KE]	β_Σ	[%]
50	10	10	0,01	0,4	1 352,3	1 268,8	5 271	1 268,8	1 066	0,00
				0,6	1 495,0	1 361,8	5 909	1 362,2	1 214	0,03
				0,8	1 637,7	1 450,6	6 404	1 451,4	1 322	0,06
			0,1	0,4	1 404,0	1 313,7	5 421	1 314,1	1 138	0,03
				0,6	1 572,7	1 423,7	6 295	1 425,2	1 338	0,11
				0,8	1 741,3	1 529,2	6 623	1 531,6	1 473	0,16
50	10	25	0,01	0,4	1 455,4	1 243,4	49 185	1 245,4	3 985	0,16
				0,6	1 691,1	1 364,2	53 485	1 367,4	4 471	0,23
				0,8	1 926,8	1 474,1	57 648	1 480,5	5 012	0,43
			0,1	0,4	1 501,5	1 282,7	48 433	1 285,8	4 104	0,24
				0,6	1 760,2	1 418,7	52 027	1 425,0	4 651	0,44
				0,8	2 019,0	1 541,4	56 983	1 553,6	5 202	0,79
75	10	25	0,01	0,4	1 928,7	1 689,9	49 864	1 691,9	3 988	0,12
				0,6	2 263,3	1 882,2	56 668	1 885,2	4 661	0,16
				0,8	2 597,9	2 064,3	61 967	2 070,3	5 294	0,29
			0,1	0,4	2 020,8	1 769,9	48 469	1 774,2	4 056	0,24
				0,6	2 401,5	1 994,0	54 326	2 004,7	4 791	0,54
				0,8	2 782,2	2 207,7	58 592	2 223,5	5 454	0,72
75	25	35	0,01	0,4	1 578,8	1 362,7	302 972	1 363,8	9 800	0,08
				0,6	1 770,1	1 449,4	323 784	1 452,5	10 809	0,21
				0,8	1 961,4	1 528,5	347 432	1 534,0	11 847	0,36
			0,1	0,4	1 693,5	1 446,8	286 860	1 452,5	10 390	0,39
				0,6	1 942,2	1 568,9	303 333	1 577,1	11 605	0,52
				0,8	2 190,9	1 677,0	328 117	1 695,3	12 703	1,09

Fortsetzung auf der nächsten Seite ...

... Fortsetzung der vorherigen Seite (Tabelle B.1)

Problemgröße			Parameter		Eröffnungs- verfahren	Lokales Suchver- fahren	Gieriges Verfahren		Δz	
$ N $	$ S $	$ B $	μ	λ	\bar{z} [KE]	\bar{z} [KE]	β_{Σ}	\bar{z} [KE]	β_{Σ}	[%]
100	25	25	0,01	0,4	1 843,9	1 680,0	149 177	1 680,5	6 355	0,03
				0,6	2 045,8	1 790,6	168 678	1 791,9	7 311	0,07
				0,8	2 247,8	1 894,6	186 060	1 897,1	8 276	0,13
			0,1	0,4	2 018,3	1 816,5	144 488	1 819,1	7 085	0,14
				0,6	2 307,4	1 982,9	162 113	1 992,0	8 208	0,46
				0,8	2 596,6	2 138,2	171 963	2 155,5	9 197	0,81
100	25	50	0,01	0,4	2 013,7	1 666,0	742 939	1 669,9	16 554	0,23
				0,6	2 330,1	1 800,8	817 147	1 807,8	18 766	0,39
				0,8	2 646,4	1 924,4	870 603	1 937,3	20 557	0,67
			0,1	0,4	2 189,4	1 798,9	668 651	1 812,1	17 693	0,73
				0,6	2 593,6	1 988,1	725 881	2 014,3	19 751	1,32
				0,8	2 997,8	2 164,1	760 980	2 204,6	21 527	1,87

Tabelle B.2: Vergleich der Verfahren zur Erstellung einer ausfallsicheren Vernetzungsstruktur unter Berücksichtigung einer zufällig erzeugten Kostenmatrix der Größe 100×100

Problemgröße			Parameter		Eröffnungs- verfahren	Auswahlstrategie		Verfeinerungsverfahren		Δz
$ N $	$ S $	$ B $	μ	λ	\bar{z} [KE]	Zufällig	Gierig	\bar{z} [KE]	Σ iter	[%]
50	10	10	0,01	0,4	3 045,2	3 014,8	72,1	3 017,7	111,6	0,10
				0,6	3 299,0	3 260,0	73,7	3 269,3	108,7	0,29
				0,8	3 552,8	3 502,7	77,3	3 511,6	112,6	0,25
			0,1	0,4	3 170,8	3 137,4	65,5	3 146,4	105,9	0,29
				0,6	3 487,4	3 434,9	71,4	3 447,4	107,5	0,36
				0,8	3 804,0	3 713,1	76,6	3 740,2	108,8	0,73
50	10	25	0,01	0,4	3 054,5	3 024,0	78,9	3 029,8	106,8	0,19
				0,6	3 471,3	3 430,0	78,8	3 437,7	111,0	0,22
				0,8	3 888,1	3 825,0	84,6	3 842,6	111,3	0,46
			0,1	0,4	3 182,5	3 145,0	78,0	3 156,8	107,5	0,38
				0,6	3 663,4	3 596,6	77,8	3 619,5	106,7	0,64
				0,8	4 144,3	4 032,8	85,9	4 070,2	109,0	0,93
75	10	25	0,01	0,4	4 270,8	4 245,0	76,8	4 248,8	109,1	0,09
				0,6	4 897,1	4 860,1	81,4	4 866,8	110,4	0,14
				0,8	5 523,3	5 466,3	84,3	5 480,1	113,4	0,25
			0,1	0,4	4 486,4	4 451,9	77,8	4 461,6	108,5	0,22
				0,6	5 220,5	5 146,2	83,5	5 174,7	107,5	0,55
				0,8	5 954,7	5 840,6	85,1	5 880,0	111,2	0,67

Fortsetzung auf der nächsten Seite ...

... Fortsetzung der vorherigen Seite (Tabelle B.2)

Problemgröße			Parameter		Eröffnungs-	Auswahlstrategie		Verfeinerungsverfahren		Δz
$ N $	$ S $	$ B $	μ	λ	verfahren	Zufällig	Gierig			[%]
					\bar{z} [KE]	\bar{z} [KE]	Σ iter	\bar{z} [KE]	Σ iter	
75	25	35	0,01	0,4	3 155,0	3 125,1	100,2	3 113,8	147,3	-0,36
				0,6	3 452,9	3 404,1	123,9	3 406,1	152,0	0,06
				0,8	3 750,9	3 686,3	126,0	3 688,9	158,2	0,07
			0,1	0,4	3 438,4	3 393,7	98,8	3 394,6	133,8	0,03
				0,6	3 878,1	3 800,2	112,6	3 806,2	133,6	0,16
				0,8	4 317,8	4 180,4	131,6	4 214,9	140,4	0,83
100	25	25	0,01	0,4	3 916,3	3 885,0	105,9	3 876,4	149,7	-0,22
				0,6	4 244,1	4 205,7	109,0	4 201,8	144,2	-0,09
				0,8	4 571,9	4 515,6	113,7	4 516,2	144,8	0,01
			0,1	0,4	4 329,9	4 281,2	93,0	4 284,6	130,9	0,08
				0,6	4 864,5	4 778,3	103,1	4 784,0	137,9	0,12
				0,8	5 399,2	5 257,2	114,9	5 280,1	132,5	0,44
100	25	50	0,01	0,4	3 992,1	3 960,0	107,4	3 953,6	146,7	-0,16
				0,6	4 486,1	4 440,0	113,8	4 434,1	155,3	-0,13
				0,8	4 980,0	4 907,1	132,3	4 913,0	156,2	0,12
			0,1	0,4	4 409,5	4 358,9	99,7	4 361,8	127,8	0,07
				0,6	5 112,2	5 017,0	112,9	5 030,5	137,6	0,27
				0,8	5 815,0	5 657,6	143,8	5 684,7	139,2	0,48

C Konfidenzintervalle

Tabelle C.1: Konfidenzintervalle zu den Ergebnissen in Tabelle 6.2

Problemgröße		optimale Lösung		SA-basiert		TS-basiert	
N	S	\bar{z} [KE]	cint	\bar{z} [KE]	cint	\bar{z} [KE]	cint
10	1	4 507,2	140,8	4 507,2	140,8	4 507,2	140,8
	2	3 347,8	94,8	3 349,3	94,9	3 348,3	94,8
	3	2 886,6	79,6	2 890,1	79,8	2 896,6	80,1
	[1,10]	2 294,0	62,6	2 313,1	63,5	2 356,8	65,8
15	1	6 956,8	156,1	6 957,6	156,1	6 956,8	156,1
	2	5 121,4	111,1	5 122,9	111,1	5 121,9	111,2
	3	4 325,9	90,8	4 330,5	91,0	4 328,1	90,9
	[1,15]	2 890,1	57,0	2 919,0	58,3	2 992,6	60,0
20	1	9 481,0	180,7	9 481,0	180,7	9 481,0	180,7
	2	6 945,7	115,5	6 950,2	115,9	6 946,9	115,4
	3	5 721,9	90,7	5 784,9	90,8	5 780,4	91,2
	[1,20]	3 376,2	51,1	3 417,5	52,0	3 492,2	55,0

Tabelle C.2: Konfidenzintervalle zu den Ergebnissen in Tabelle 6.3

Problemgröße		optimale Lösung		SA-basiert		TS-basiert	
N	S	\bar{z} [KE]	cint	\bar{z} [KE]	cint	\bar{z} [KE]	cint
10	1	4 616,0	147,0	4 616,0	147,0	4 616,0	147,0
	2	3 462,1	96,1	3 462,1	96,1	3 607,4	105,4
	3	2 995,6	82,8	2 995,6	82,8	3 229,4	92,8
	[1, 10]	2 553,7	73,9	2 553,7	73,9	2 553,7	73,9
15	1	7 114,5	161,2	7 114,5	161,2	7 114,5	161,2
	2	5 235,1	113,9	5 235,1	113,9	5 510,3	130,0
	3	4 368,7	96,5	4 455,5	96,5	4 832,7	111,8
	[1, 15]	3 236,8	70,2	3 236,8	70,2	3 236,8	70,2
20	1	9 653,8	192,9	9 653,8	192,9	9 653,8	192,9
	2	7 081,1	122,5	7 081,1	122,4	7 366,6	135,3
	3	5 923,6	95,9	5 923,6	95,9	6 359,1	106,0
	[1, 20]	3 793,6	64,4	3 793,6	64,4	3 793,6	64,4

Tabelle C.3: Konfidenzintervalle zu den Ergebnissen in Tabelle 6.4

Problemgröße		Parameter		SA-basiert		TS-basiert	
$ N $	$ S $	λ	Γ	\bar{z} [KE]	cint	\bar{z} [KE]	cint
50	5	0,6	=	478,6	4,9	498,7	5,0
			≠	427,3	3,9	449,8	4,4
		0,8	=	526,3	5,3	567,5	6,3
			≠	465,4	4,5	505,9	5,5
	10	0,6	=	380,2	3,2	394,1	3,8
			≠	341,3	2,8	358,5	3,3
		0,8	=	402,3	3,2	429,7	4,3
			≠	358,8	2,8	386,8	3,7
	25	0,6	=	307,7	2,6	313,2	3,0
			≠	277,9	2,4	287,5	2,8
		0,8	=	314,3	2,6	324,8	3,1
			≠	281,9	2,5	296,6	3,0
75	10	0,6	=	540,7	4,3	562,4	4,6
			≠	486,7	4,6	512,1	4,6
		0,8	=	589,7	6,1	631,2	5,6
			≠	514,4	4,7	570,5	5,4
	20	0,6	=	438,1	3,3	450,3	3,4
			≠	394,5	2,7	414,3	3,2
		0,8	=	457,7	3,7	484,0	3,9
			≠	406,0	2,8	443,3	3,7
	30	0,6	=	398,0	2,8	403,6	3,0
			≠	359,3	2,4	372,7	2,7
		0,8	=	409,1	2,9	424,4	3,2
			≠	366,3	2,4	389,3	3,0
100	10	0,6	=	701,3	5,2	716,2	4,9
			≠	629,1	4,7	658,1	4,9
		0,8	=	791,2	6,7	819,8	6,3
			≠	689,0	6,8	743,1	5,6
	25	0,6	=	520,3	3,8	525,3	3,5
			≠	467,6	3,5	486,1	3,0
		0,8	=	546,7	4,4	567,2	4,1
			≠	481,3	3,4	521,5	3,6
	50	0,6	=	436,3	2,8	446,8	2,9
			≠	398,5	2,5	414,2	2,7
		0,8	=	447,0	2,9	465,2	3,1
			≠	402,9	2,4	429,2	2,8

Tabelle C.4: Konfidenzintervalle zu den Ergebnissen in Tabelle 6.5

Problemgröße		Parameter		SA-basiert		TS-basiert			
$ N $	$ S $	λ	Γ	\bar{z} [KE]	cint	\bar{z} [KE]	cint		
50	5	0,6	=	470,4	4,2	536,8	5,5		
			≠	428,3	4,4	483,5	4,8		
		0,8	=	527,1	5,1	617,4	7,2		
			≠	472,6	4,8	551,2	6,2		
		10	0,6	=	374,0	3,2	431,8	4,4	
				≠	342,0	2,9	393,9	4,1	
	0,8		=	400,7	3,5	479,0	5,3		
			≠	364,4	3,3	430,9	4,9		
	25		0,6	=	306,0	2,9	337,0	3,1	
				≠	280,6	2,7	306,8	2,9	
			0,8	=	312,1	2,8	352,4	3,4	
				≠	283,8	2,6	317,7	3,0	
75	10	0,6	=	524,8	4,5	609,4	5,9		
			≠	481,1	3,8	556,4	5,4		
		0,8	=	576,2	5,2	692,7	7,6		
			≠	519,5	4,5	623,9	6,8		
		20	0,6	=	423,4	3,1	492,2	4,1	
				≠	390,7	2,9	451,5	3,7	
	0,8		=	446,1	3,4	537,1	4,9		
			≠	408,8	3,0	487,5	4,2		
	30	0,6	=	387,3	2,8	438,9	3,4		
			≠	356,8	2,5	402,6	3,0		
		0,8	=	399,9	2,9	466,5	3,8		
			≠	367,5	2,7	424,0	3,4		
	100	10	0,6	=	678,5	4,5	786,0	5,8	
				≠	618,5	3,9	719,4	5,6	
			0,8	=	758,6	6,2	907,2	7,5	
				≠	682,3	5,5	822,5	6,6	
			25	0,6	=	496,9	3,3	580,7	3,8
					≠	457,8	3,1	535,4	3,4
0,8		=		523,6	3,4	635,3	4,5		
		≠		479,2	3,3	580,0	3,8		
50		0,6	=	425,7	2,6	475,9	3,1		
			≠	395,8	2,6	439,0	2,7		
		0,8	=	435,9	2,7	497,9	3,2		
			≠	401,8	2,5	455,4	3,0		

Tabelle C.5: Konfidenzintervalle zu den Ergebnissen in Tabelle 6.6

Problemgröße			Parameter		Eröffnungsverfahren		Lokales Suchverfahren		Gieriges Verfahren	
$ N $	$ S $	$ B $	μ	λ	\bar{z} [KE]	cint	\bar{z} [KE]	cint	\bar{z} [KE]	cint
50	10	10	0,01	0,4	1 394,3	20,8	1 314,0	18,0	1 314,0	18,0
				0,6	1 536,7	28,3	1 408,5	22,9	1 408,9	22,9
				0,8	1 679,1	36,6	1 499,3	28,4	1 500,3	28,5
			0,1	0,4	1 448,6	21,3	1 362,0	18,3	1 362,6	18,3
				0,6	1 618,1	29,4	1 475,8	23,7	1 477,0	23,9
				0,8	1 787,6	38,2	1 583,2	29,7	1 586,0	30,1
50	10	25	0,01	0,4	1 503,6	32,5	1 290,9	15,4	1 293,1	15,6
				0,6	1 746,4	47,7	1 414,4	19,8	1 417,9	20,0
				0,8	1 989,3	63,2	1 528,3	24,7	1 534,6	25,1
			0,1	0,4	1 553,5	31,7	1 332,5	15,1	1 336,9	15,3
				0,6	1 821,3	46,7	1 473,2	19,6	1 479,3	19,9
				0,8	2 089,1	61,9	1 601,4	24,6	1 612,9	24,9
75	10	25	0,01	0,4	2 004,4	41,6	1 764,5	25,5	1 766,9	26,1
				0,6	2 349,3	60,2	1 964,3	33,8	1 968,1	34,7
				0,8	2 694,1	79,1	2 154,3	42,1	2 159,8	42,7
			0,1	0,4	2 095,6	40,3	1 848,9	26,1	1 853,3	26,2
				0,6	2 486,1	58,0	2 083,8	34,4	2 092,8	34,2
				0,8	2 876,5	76,1	2 308,7	42,6	2 323,8	42,5
75	25	35	0,01	0,4	1 627,5	29,8	1 408,3	14,1	1 412,3	14,8
				0,6	1 820,9	42,9	1 496,7	17,5	1 500,9	17,7
				0,8	2 014,4	56,3	1 576,8	21,0	1 583,9	21,2
			0,1	0,4	1 745,3	28,3	1 499,8	14,7	1 507,7	15,3
				0,6	1 997,6	40,7	1 622,9	18,5	1 637,5	18,9
				0,8	2 249,9	53,4	1 738,0	22,4	1 761,4	23,6
100	25	25	0,01	0,4	1 899,8	26,0	1 741,2	16,1	1 741,4	16,1
				0,6	2 099,5	36,7	1 851,9	20,3	1 853,2	20,3
				0,8	2 299,2	47,8	1 956,0	24,7	1 958,6	24,8
			0,1	0,4	2 079,9	26,7	1 884,6	17,3	1 888,4	17,1
				0,6	2 369,6	37,9	2 052,2	21,7	2 064,1	22,2
				0,8	2 659,3	49,4	2 210,4	26,5	2 229,7	27,5
100	25	50	0,01	0,4	2 056,9	29,7	1 723,4	13,9	1 730,0	14,4
				0,6	2 370,2	43,5	1 860,1	17,6	1 870,4	18,6
				0,8	2 683,5	57,5	1 986,0	21,6	2 002,3	23,4
			0,1	0,4	2 230,9	29,9	1 862,4	15,4	1 883,3	16,4
				0,6	2 631,2	43,5	2 056,6	20,0	2 091,8	22,2
				0,8	3 031,5	57,3	2 237,2	25,0	2 295,5	28,3

Tabelle C.6: Konfidenzintervalle zu den Ergebnissen in Tabelle 6.7

Problemgröße			Parameter		Eröffnungsverfahren		Auswahlstrategie		Verfeinerungsverfahren	
N	S	B	μ	λ	\bar{z} [KE]	cint	Zufällig		Gierig	
							\bar{z} [KE]	cint	\bar{z} [KE]	cint
50	10	10	0,01	0,4	2 609,0	38,3	2 576,6	38,4	2 580,7	38,0
				0,6	2 824,2	48,0	2 782,3	47,8	2 787,1	47,5
				0,8	3 039,3	59,3	2 985,9	59,0	2 996,1	58,8
			0,1	0,4	2 716,0	37,4	2 682,9	36,7	2 688,9	37,9
				0,6	2 984,6	47,6	2 936,8	47,7	2 942,3	46,9
				0,8	3 253,2	59,5	3 177,8	59,5	3 194,4	59,2
50	10	25	0,01	0,4	2 613,2	37,4	2 583,5	37,7	2 588,2	37,6
				0,6	2 964,2	50,0	2 920,1	49,8	2 932,1	49,8
				0,8	3 315,2	63,6	3 260,1	63,3	3 273,0	63,4
			0,1	0,4	2 725,3	36,7	2 689,9	36,4	2 698,8	36,1
				0,6	3 132,4	49,5	3 068,3	48,5	3 087,7	47,6
				0,8	3 539,3	63,4	3 443,4	61,4	3 468,2	61,6
75	10	25	0,01	0,4	3 690,6	58,3	3 662,7	58,0	3 665,8	58,6
				0,6	4 226,3	77,5	4 186,5	77,0	4 195,4	77,4
				0,8	4 762,0	98,0	4 701,2	98,3	4 719,5	98,3
			0,1	0,4	3 881,9	55,9	3 840,0	55,9	3 851,3	56,1
				0,6	4 513,3	74,1	4 438,0	74,0	4 461,1	73,5
				0,8	5 144,6	93,8	5 020,9	94,3	5 061,7	93,1
75	25	35	0,01	0,4	2 694,7	33,6	2 658,1	32,8	2 649,8	33,3
				0,6	2 942,6	42,7	2 894,5	42,0	2 890,4	41,8
				0,8	3 190,4	52,4	3 124,3	50,8	3 126,6	51,7
			0,1	0,4	2 937,6	33,7	2 893,9	32,6	2 893,1	32,7
				0,6	3 306,9	44,2	3 231,4	41,6	3 234,7	42,2
				0,8	3 676,2	55,6	3 564,4	50,8	3 573,0	52,3
100	25	25	0,01	0,4	3 358,1	34,7	3 329,2	34,7	3 317,4	35,5
				0,6	3 633,5	45,7	3 590,6	45,5	3 587,5	46,2
				0,8	3 908,9	57,8	3 850,7	57,6	3 851,0	57,9
			0,1	0,4	3 714,0	34,9	3 667,5	34,6	3 669,5	34,3
				0,6	4 167,3	47,0	4 087,5	45,2	4 095,6	45,6
				0,8	4 620,7	60,2	4 491,2	58,2	4 507,8	57,4
100	25	50	0,01	0,4	3 419,4	35,9	3 388,0	35,0	3 382,0	35,5
				0,6	3 838,3	50,2	3 793,1	48,9	3 790,9	49,5
				0,8	4 257,2	65,3	4 191,2	63,3	4 192,2	64,4
			0,1	0,4	3 786,6	36,8	3 736,8	35,9	3 740,0	35,6
				0,6	4 389,1	52,2	4 296,0	51,2	4 309,4	51,2
				0,8	4 991,6	68,3	4 832,0	64,9	4 866,0	64,4

Tabelle C.7: Konfidenzintervalle zu den Ergebnissen in Tabelle B.1

Problemgröße			Parameter		Eröffnungsverfahren		Lokales Suchverfahren		Gieriges Verfahren	
$ N $	$ S $	$ B $	μ	λ	\bar{z} [KE]	cint	\bar{z} [KE]	cint	\bar{z} [KE]	cint
50	10	10	0,01	0,4	1 352,3	22,4	1 268,8	19,4	1 268,8	19,4
				0,6	1 495,0	30,6	1 361,8	24,6	1 362,2	24,6
				0,8	1 637,7	39,5	1 450,6	30,2	1 451,4	30,3
			0,1	0,4	1 404,0	23,0	1 313,7	19,8	1 314,1	19,8
				0,6	1 572,7	31,7	1 423,7	25,4	1 425,2	25,5
				0,8	1 741,3	41,1	1 529,2	31,3	1 531,6	31,8
50	10	25	0,01	0,4	1 455,4	30,7	1 243,4	15,7	1 245,4	15,8
				0,6	1 691,1	44,8	1 364,2	20,4	1 367,4	20,7
				0,8	1 926,8	59,0	1 474,1	25,3	1 480,5	25,8
			0,1	0,4	1 501,5	30,0	1 282,7	16,2	1 285,8	16,3
				0,6	1 760,2	43,8	1 418,7	20,8	1 425,0	21,1
				0,8	2 019,0	57,7	1 541,4	25,7	1 553,6	26,5
75	10	25	0,01	0,4	1 928,7	39,1	1 689,9	23,5	1 691,9	23,7
				0,6	2 263,3	56,1	1 882,2	31,0	1 885,2	31,1
				0,8	2 597,9	73,3	2 064,3	38,8	2 070,3	38,9
			0,1	0,4	2 020,8	37,1	1 769,9	23,8	1 774,2	24,0
				0,6	2 401,5	53,0	1 994,0	31,2	2 004,7	31,5
				0,8	2 782,2	69,1	2 207,7	38,7	2 223,5	39,3
75	25	35	0,01	0,4	1 578,8	25,7	1 362,7	12,8	1 363,8	12,8
				0,6	1 770,1	37,2	1 449,4	15,9	1 452,5	16,0
				0,8	1 961,4	48,9	1 528,5	18,9	1 534,0	19,0
			0,1	0,4	1 693,5	24,8	1 446,8	13,0	1 452,5	12,8
				0,6	1 942,2	35,8	1 568,9	16,1	1 577,1	15,8
				0,8	2 190,9	47,1	1 677,0	19,8	1 695,3	19,0
100	25	25	0,01	0,4	1 843,9	27,0	1 680,0	15,1	1 680,5	15,1
				0,6	2 045,8	38,8	1 790,6	19,2	1 791,9	19,2
				0,8	2 247,8	50,9	1 894,6	23,6	1 897,1	23,6
			0,1	0,4	2 018,3	27,1	1 816,5	16,3	1 819,1	16,3
				0,6	2 307,4	38,9	1 982,9	20,9	1 992,0	21,3
				0,8	2 596,6	51,0	2 138,2	25,8	2 155,5	26,3
100	25	50	0,01	0,4	2 013,7	35,1	1 666,0	13,6	1 669,9	13,7
				0,6	2 330,1	51,8	1 800,8	17,1	1 807,8	17,4
				0,8	2 646,4	68,8	1 924,4	20,7	1 937,3	21,1
			0,1	0,4	2 189,4	34,6	1 798,9	14,9	1 812,1	15,1
				0,6	2 593,6	50,8	1 988,1	19,2	2 014,3	19,8
				0,8	2 997,8	67,2	2 164,1	23,5	2 204,6	24,5

Tabelle C.8: Konfidenzintervalle zu den Ergebnissen in Tabelle B.2

Problemgröße			Parameter		Eröffnungsverfahren		Auswahlstrategie		Verfeinerungsverfahren	
N	S	B	μ	λ	\bar{z} [KE]	cint	Zufällig		Gierig	
							\bar{z} [KE]	cint	\bar{z} [KE]	cint
50	10	10	0,01	0,4	3 045,2	44,6	3 014,8	44,8	3 017,7	45,0
				0,6	3 299,0	56,9	3 260,0	57,9	3 269,3	57,3
				0,8	3 552,8	70,8	3 502,7	71,2	3 511,6	70,6
			0,1	0,4	3 170,8	44,1	3 137,4	43,9	3 146,4	44,4
				0,6	3 487,4	57,2	3 434,9	57,5	3 447,4	56,7
				0,8	3 804,0	72,1	3 713,1	70,5	3 740,2	72,6
50	10	25	0,01	0,4	3 054,5	42,3	3 024,0	42,8	3 029,8	42,2
				0,6	3 471,3	57,2	3 430,0	57,4	3 437,7	57,4
				0,8	3 888,1	73,3	3 825,0	73,1	3 842,6	73,8
			0,1	0,4	3 182,5	42,5	3 145,0	41,5	3 156,8	42,3
				0,6	3 663,4	58,3	3 596,6	56,6	3 619,5	57,7
				0,8	4 144,3	75,3	4 032,8	73,8	4 070,2	75,1
75	10	25	0,01	0,4	4 270,8	64,0	4 245,0	64,3	4 248,8	64,4
				0,6	4 897,1	86,4	4 860,1	86,7	4 866,8	86,2
				0,8	5 523,3	110,2	5 466,3	110,5	5 480,1	110,2
			0,1	0,4	4 486,4	61,4	4 451,9	61,6	4 461,6	61,6
				0,6	5 220,5	83,2	5 146,2	83,9	5 174,7	83,8
				0,8	5 954,7	106,6	5 840,6	107,2	5 880,0	107,1
75	25	35	0,01	0,4	3 155,0	35,3	3 125,1	34,8	3 113,8	34,9
				0,6	3 452,9	46,2	3 404,1	44,8	3 406,1	45,6
				0,8	3 750,9	57,8	3 686,3	55,7	3 688,9	57,4
			0,1	0,4	3 438,4	33,4	3 393,7	33,3	3 394,6	32,5
				0,6	3 878,1	44,4	3 800,2	43,4	3 806,2	44,2
				0,8	4 317,8	56,1	4 180,4	54,6	4 214,9	54,4
100	25	25	0,01	0,4	3 916,3	38,3	3 885,0	39,1	3 876,4	38,3
				0,6	4 244,1	52,0	4 205,7	51,9	4 201,8	51,7
				0,8	4 571,9	66,9	4 515,6	66,4	4 516,2	66,2
			0,1	0,4	4 329,9	39,7	4 281,2	38,5	4 284,6	38,8
				0,6	4 864,5	54,7	4 778,3	54,2	4 784,0	53,0
				0,8	5 399,2	70,9	5 257,2	69,3	5 280,1	69,2
100	25	50	0,01	0,4	3 992,1	39,8	3 960,0	39,1	3 953,6	38,6
				0,6	4 486,1	55,2	4 440,0	54,4	4 434,1	54,7
				0,8	4 980,0	71,4	4 907,1	70,7	4 913,0	71,0
			0,1	0,4	4 409,5	40,2	4 358,9	39,8	4 361,8	40,1
				0,6	5 112,2	56,0	5 017,0	56,0	5 030,5	56,2
				0,8	5 815,0	72,6	5 657,6	69,9	5 684,7	72,5

Literaturverzeichnis

- [1] J. SOMMER. “On Optimal Communication Spanning Trees in Embedded Ethernet Networks”. In: *Proceedings of the 8th IEEE International Workshop on Factory Communication Systems (WFCS)*. Mai 2010, S. 141–150. doi: 10.1109/WFCS.2010.5548629.
- [2] J. SOMMER, E. A. DOUMITH und Q. DUVAL. “On Link Harness Optimization of Embedded Ethernet Networks”. In: *Proceedings of the IEEE International Symposium on Industrial Embedded Systems (SIES)*. Juli 2009, S. 191–200. doi: 10.1109/SIES.2009.5196215.
- [3] J. SOMMER und E. A. DOUMITH. “Topology Optimization of In-vehicle Multimedia Communication Systems”. In: *Proceedings of the First Annual International Symposium on Vehicular Computing Systems (ISVCS)*. Dublin, Juli 2008.
- [4] J. SOMMER, S. GUNREBEN, A. MIFDAOUI, F. FELLER, M. KÖHN, D. SASS und J. SCHARF. “Ethernet – A Survey on its Fields of Application”. In: *IEEE Communications Surveys & Tutorials* 12.2 (April 2010), S. 263–284. doi: 10.1109/SURV.2010.021110.00086.
- [5] J. SOMMER, E. A. DOUMITH und A. REIFERT. “Cost-based Topology Optimization of Embedded Ethernet Network”. In: *Journal of Embedded and Real-Time Communication Systems (IJERTCS)* 2.1 (2011), S. 1–22.
- [6] J. SOMMER. “Topologieoptimierung von eingebetteten Kommunikationsnetzen – Herausforderungen und Lösungsansätze”. In: *Workshop: Optimierung und Optimierungsverfahren, -algorithmen*. Hrsg. von ITG FACHGRUPPE 5.2.1. Stuttgart, November 2009.
- [7] J. SOMMER und J. SCHARF. “Modeling and Tools for Network Simulation”. In: Hrsg. von K. WEHRLE, M. GÜNES und J. GROSS. 1. Aufl. s. Springer, 2010. Kap. IKR Simulation Library, S. 61–68.
- [8] J. SOMMER, V. FEIL und E. A. SANZ. “Scheduling Algorithms for Simultaneous Software Updates of Electronic Devices in Vehicles”. In: *IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE. Beijing, China, Mai 2008, S. 393–399. doi: 10.1109/ICCW.2008.80.
- [9] M. C. NECKER, M. KÖHN, A. REIFERT, J. SCHARF und J. SOMMER. “Optimized Frame Packing for OFDMA Systems”. In: *IEEE Vehicular Technology Conference (VTC Spring)*. IEEE. Singapore, Mai 2008, S. 1483–1488. doi: 10.1109/VETECS.2008.348.

- [10] J. SOMMER und R. BLIND. “Optimized Resource Dimensioning in an embedded CAN-CAN Gateway”. In: *Proceedings of the IEEE Second International Symposium on Industrial Embedded Systems (SIES)*. Juli 2007, S. 55–62. doi: 10.1109/SIES.2007.4297317.
- [11] J. SOMMER, L. BURGSTÄHLER und V. FEIL. “An Analysis of Automotive Multi-Domain CAN Systems”. In: *Proceedings of the 12th EUNICE Open European Summer School (EUNICE)*. 2006.
- [12] R. I. YOUSSEF. “Evaluation of Heuristics for Topology Optimization of Embedded Ethernet Networks”. Bachelorthesis. Universität Stuttgart, Institut für Kommunikationsnetze und Rechnersysteme (IKR), Juli 2010.
- [13] D. TEUCHERT. “Topologieoptimierung von ausfallsicheren, eingebetteten Kommunikationsnetzen”. Diplomarbeit. Universität Stuttgart, Institut für Kommunikationsnetze und Rechnersysteme (IKR), Mai 2010.
- [14] I. KAUFFMANN. “Implementierung und Bewertung einer Heuristik zur Topologieoptimierung von eingebetteten Ethernet-Netzen”. Studienarbeit. Universität Stuttgart, Institut für Kommunikationsnetze und Rechnersysteme (IKR), 2011.
- [15] Q. DUVAL. “Topology Optimization of an Embedded Ethernet Network”. Masterthesis. Universität Stuttgart, Institut für Kommunikationsnetze und Rechnersysteme (IKR), Mai 2009.
- [16] A. DAHER. “Topology Optimization of an Ethernet-based In-vehicle Communication System”. Masterthesis. Universität Stuttgart, Institut für Kommunikationsnetze und Rechnersysteme (IKR), Februar 2008.
- [17] W. ROSENSTIEL. “Entwurfsmethoden für eingebettete Systeme”. In: *it+ti-Informationstechnik und Technische Informatik*. Bd. 41. München: Oldenburg Verlag, 1999.
- [18] P. MARWEDEL und L. WEHMEYER. *Eingebettete Systeme*. 1. Aufl. Springer, 2007, S. 265.
- [19] B. WALKE. *Mobilfunknetze und ihre Protokolle – Grundlagen, GSM, UMTS und andere zellulare Mobilfunknetze*. Hrsg. von M. BOSSERT und N. FLIEGE. 3. Aufl. Vieweg+Teubner, 2001, S. 553.
- [20] T. PFEIFER und K.-U. HEILER. “Ziele und Anwendungen von Feldbussystemen”. In: *Automatisierungstechnische Praxis atp* 29.12 (1987), S. 549–557.
- [21] N. NAVET, Y. SONG, F. SIMONOT-LION und C. WILWERT. “Trends in Automotive Communication Systems”. In: *Proceedings of the IEEE*. Bd. 93. 6. IEEE. Juni 2005, S. 1204–1223.
- [22] R. SCHREINER. *Computernetzwerke – Von den Grundlagen zur Funktion und Anwendung*. 3. Aufl. Hanser, 2009, S. 312.
- [23] H. WEBER und M. BROJ. “Systemorientiertes Automotive Engineering”. In: *Informatik-Spektrum* 32.3 (Juni 2009), S. 206–213. doi: 10.1007/s00287-009-0337-6.
- [24] H. SALZWEDEL. *Control Design of Aerospace and Automotive Systems – How to reduce risk in complex system design*. Symposium on Automotive/Avionics Systems Engineering (SAASE). Oktober 2009.

- [25] G. SCHNELL und B. WIEDEMANN. *Bussysteme in der Automatisierungs- und Prozess-technik*. 6. Aufl. Vieweg, Oktober 2006, S. 420.
- [26] J.-D. WANG. “Intelligenter Hausservice”. In: *ABB Technik 2* (2009), S. 55–57.
- [27] D. SNOONIAN. “Smart buildings”. In: *IEEE Spectrum*. Bd. 40. 8. August 2003, S. 18–23. doi: 10.1109/MSPEC.2003.1222043.
- [28] W. KASTNER, G. NEUGSCHWANDTNER, S. SOUCEK und H. M. NEWMAN. “Communication Systems for Building Automation and Control”. In: *Proceedings of the IEEE 93.6* (Juni 2005), S. 1178–1203. doi: 10.1109/JPROC.2005.849726.
- [29] A. S. TANENBAUM. *Computernetzwerke*. 4. Aufl. Pearson Studium, 2003.
- [30] T. SAUTER und M. WOLLSCHLAEGER. “Feldbussysteme – Historie, Eigenschaften und Entwicklungstrends”. In: *it+ti-Informationstechnik und Technische Informatik 42.4* (2000), S. 7–16.
- [31] J. F. WOLLERT. “Handbuch der Mess- und Automatisierungstechnik in der Produktion”. In: Springer, 2006. Kap. Teil E - Bussysteme. doi: 10.1007/3-540-34823-9.
- [32] J.-P. THOMESSE. “Fieldbus technology in industrial automation”. In: *Proceedings of the IEEE*. Bd. 93. 6. IEEE. Juni 2005, S. 1073 –1101.
- [33] ROBERT BOSCH GMBH. *Bosch Automotive Handbook*. Hrsg. von ROBERT BOSCH GMBH. 7. Aufl. Germany: Robert Bosch GmbH, Juli 2007.
- [34] P.-M. ZIEGLER und B. BENZ. “Fliegendes Rechnernetz – IT-Technik an Bord des Airbus A380”. In: *c’t – Magazin für Computertechnik 15* (2005), S. 84–91.
- [35] J. RENNER. “Mobile Agenten für den Fernzugriff auf eingebettete Systeme”. Diss. Technische Universität Chemnitz, 2006.
- [36] H. BALZERT. *Lehrbuch der Software-Technik – Software-Entwicklung*. 2. Aufl. Spektrum-Akademischer Verlag, Dezember 2000, S. 1136.
- [37] T. EMMANUEL. “Planguage – Spezifikation nichtfunktionaler Anforderungen”. In: *Informatik-Spektrum 33.3* (Juni 2010), S. 292–295. doi: 10.1007/s00287-010-0435-5.
- [38] P. J. KÜHN. *Communication Networks I*. Skript zur Vorlesung an der Universität Stuttgart. 2006.
- [39] P. RECHENBERG und G. POMBERGER. *Informatik-Handbuch*. Hanser, 2002.
- [40] J. JASPERNEITE. “Leistungsbewertung eines lokalen Netzwerkes mit Class-of-Service Unterstützung für die prozessnahe Echtzeitkommunikation”. Diss. Otto-von-Guericke Universität Magdeburg, 2002.
- [41] A. BORMANN und I. HILGENKAMP. *Industrielle Netze. Ethernet-Kommunikation für Automatisierungsanwendungen*. 1. Aufl. Hüthig, 2006.
- [42] D. MILLS. *Network Time Protocol (NTP)*. RFC 958. Obsoleted by RFCs 1059, 1119, 1305. September 1985.
- [43] D. MILLS. *Simple Network Time Protocol (SNTP)*. RFC 1769 (Informational). Obsoleted by RFC 2030. März 1995.
- [44] D. MILLS. *Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI*. RFC 2030 (Informational). Oktober 1996.

- [45] IEEE COMPUTER SOCIETY. *1588: IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. IEEE Computer Society, November 2002.
- [46] F. FREILING. *Angewandte IT-Sicherheit*. Skript zur Vorlesung an der Universität Mannheim. 2006.
- [47] P. J. KÜHN. *Technische Informatik II – Rechnerarchitekturen und Rechnerkommunikation*. Skript zur Vorlesung an der Universität Stuttgart. 2008.
- [48] M. STÜMPFLE. “Planung und Optimierung von prioritätsbasierten Steuergerätenetzen für Fahrzeuge”. Diss. Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1999.
- [49] N. KAMMENHUBER, A. FESSI und G. CARLE. “Resilience: Widerstandsfähigkeit des Internets gegen Störungen – Stand der Forschung und Entwicklung”. In: *Informatik-Spektrum* 33.2 (April 2010), S. 131–142.
- [50] A. HAIDER und R. HARRIS. “Recovery techniques in next generation networks”. In: *IEEE Communications Surveys and Tutorials* 9.1-4 (2007), S. 2–17.
- [51] M. MENTH, R. MARTIN, A. M. KOSTER und S. ORLOWSKI. “Overview of Resilience Mechanisms Based on Multipath Structures”. In: *The 6th International Workshop on Design of Reliable Communication Networks (DRCN)*. La Rochelle, France, Oktober 2007.
- [52] ISO. *International Standard 7498–1: Information technology – Open Systems Interconnection – Basic Reference Model: The basic model*. ISO/IEC, 1994.
- [53] G. K. UND DIETRICH RESCHKE. *Lehr- und Übungsbuch Telematik: Netze, Dienste, Protokolle*. 3. Aufl. Hanser, 2004, S. 439.
- [54] E. W. BIERSACK. “Annotated bibliography on network interconnection”. In: *IEEE Journal on Selected Areas in Communications* 8.1 (Januar 1990), S. 22–41.
- [55] D. J. STERLING und S. P. WISSLER. *The Industrial Ethernet Networking Guide*. 1. Aufl. Delmar Cengage Learning, Oktober 2002, S. 352.
- [56] *IEC 61158 – Industrial communication networks – Fieldbus specifications*. English. Geneva, Switzerland: International Electrotechnical Commission, 2007.
- [57] U. KIENCKE. “Controller Area Network – From Concept to Reality”. In: *Proceedings of the 1st. International CAN Conference*. CAN in Automation (CIA). Erlangen, 1994, S. 2–16.
- [58] *ISO 11898-1:2003 – Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signalling*. International Organization for Standardization, November 2003.
- [59] W. ZIMMERMANN und R. SCHMIDGALL. *Bussysteme in der Fahrzeugtechnik*. 1. Aufl. Vieweg, 2006.
- [60] *Digital Time Division Command/Response Multiplex Data Bus (Mil-Std-1553B)*. English. USA: Department of Defense, United States of America, September 1978.
- [61] *Mil-Std-1553 Tutorial*. Condor Engineering, Inc. Santa Barbara, CA 93101, Juni 2000.

- [62] *ISO/IEC 14543 – Information technology - Home electronic system (HES) architecture*. English. Geneva, Switzerland: International Electrotechnical Commission.
- [63] T. WEINZIERL. “Gebäudesystemtechnik mit KNX – ein Überblick”. In: *WissenHeute* 61.7 (Juli 2008), S. 4–8.
- [64] *EN 50090 – Home and Building Electronic Systems (HBES)*. English. Brussels, Belgium: Europäisches Komitee für elektrotechnische Normung.
- [65] H. MERZ, T. HANSEMANN und C. HÜBNER. *Gebäudeautomation – Kommunikationssysteme mit EIB/KNX, LON und BACnet*. 2. Aufl. Hanser, November 2009, S. 300.
- [66] *MOST Specification – MOST Media Oriented Systems Transport – Multimedia and Control Networking Technology*. English. 76185 Karlsruhe, Germany: MOST Cooperation, Oktober 2009.
- [67] A. GRZEMBA. *MOST: The Automotive Multimedia Network*. 1. Aufl. Franzis Verlag, 2008, S. 388.
- [68] G. GRUHLER. *Feldbusse und Gerätekommunikationssysteme*. Franzis Verlag, 2001, S. 244.
- [69] R. ZURAWSKI. *Embedded Systems Handbook*. CRC Press Inc., August 2005, S. 1160.
- [70] R. ZURAWSKI. *The Industrial Communication Technology Handbook*. Hrsg. von R. ZURAWSKI. CRC Press, Februar 2005.
- [71] *LIN – Specification Package*. LIN Consortium, November 2006.
- [72] *IEC 62026-2 – Low-voltage switchgear and controlgear - Controller-device interfaces (CDIs) - Part 2: Actuator sensor interface (AS-i)*. English. Geneva, Switzerland: International Electrotechnical Commission, Januar 2008.
- [73] J. HEIDEMANN und R. GOVINDAN. *An Overview of Embedded Sensor Networks*. Techn. Ber. ISI-TR-2004-594. University of Southern California, Information Sciences Institute, November 2004.
- [74] C. E. SPURGEON. *Ethernet: The Definitive Guide*. 1. Aufl. O’Reilly Media, Februar 2000.
- [75] N. ABRAMSON. “Development of the ALOHANET”. In: *Information Theory, IEEE Transactions on* 31.2 (März 1985), S. 119–123.
- [76] R. ROM und M. SIDI. *Multiple Access Protocols. Performance and Analysis*. Springer, 1990, S. 177.
- [77] R. M. METCALFE und D. R. BOGGS. “Ethernet: Distributed Packet Switching for Local Computer Networks”. In: *Commun. ACM* 19.7 (1976), S. 395–404. doi: <http://doi.acm.org/10.1145/360248.360253>.
- [78] R. M. METCALFE, D. R. BOGGS, C. P. THACKER und B. W. LAMPSON. “Multipoint data communication system with collision detection”. English. Pat. 4063220. Dezember 1977.
- [79] IEEE COMPUTER SOCIETY. *802.3: IEEE Standard for Local and Metropolitan Area Networks—Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*. 2005.

- [80] IEEE COMPUTER SOCIETY. *802.2: IEEE Standard for Local and Metropolitan Area Networks—Logical Link Control*. 1998.
- [81] IEEE COMPUTER SOCIETY. *P802.3ba: IEEE Standard for Local and Metropolitan Area Networks—Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Amendment: Media Access Control Parameters, Physical Layers and Management Parameters for 40 Gb/s and 100 Gb/s Operation*. 2008.
- [82] IEEE COMPUTER SOCIETY. *IEEE 802.1 AV Bridging Task Group*. 2010.
- [83] IEEE COMPUTER SOCIETY. *P802.1Qat/D1.3 Draft Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks—Amendment: Stream Reservation Protocol (SRP)*. This is an unapproved IEEE Standards Draft. Mai 2008.
- [84] IEEE COMPUTER SOCIETY. *P802.1Qav/D3.0 Draft Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks—Amendment: Forwarding and Queuing Enhancements for Time-Sensitive Streams*. This is an unapproved IEEE Standards Draft. Juli 2008.
- [85] FRAUNHOFER ESK. *Ethernet macht Infotainment im Auto flexibel*. <http://www.esk.fraunhofer.de/de/Presse/pressemitteilungen/pm1003.html>. Pressemitteilung. April 2010.
- [86] J. HILLEBRAND, M. RAHMANI, R. BOGENBERGER und E. STEINBACH. “Coexistence of Time-Triggered and Event-Triggered Traffic in Switched Full-Duplex Ethernet Networks”. In: *Proceedings of the IEEE Second International Symposium on Industrial Embedded Systems (SIES 2007)*. 2007, S. 217–224.
- [87] R. STEFFEN, R. BOGENBERGER, J. HILLEBRAND, W. HINTERMAIER, A. WINCKLER und M. RAHMANI. “Design and Realization of an IP-based In-car Network Architecture”. In: *Proceedings of the First Annual International Symposium on Vehicular Computing Systems (ISVCS)*. Dublin, Juli 2008.
- [88] M. RAHMANI, R. STEFFEN, K. TAPPAYUTHPIJARN, E. STEINBACH und G. GIORDANO. “Performance analysis of different network topologies for in-vehicle audio and video communication”. In: *4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks (IT-NEWS)* (Februar 2008), S. 179–184. doi: 10.1109/ITNEWS.2008.4488150.
- [89] M. RAHMANI. “A Resource-Efficient IP-based Network Architecture for In-Vehicle Communication”. Diss. Technische Universität München – Lehrstuhl für Medientechnik, 2009.
- [90] K. FOULI und M. MAIER. “The road to carrier-grade ethernet”. In: *Communications Magazine, IEEE* 47.3 (März 2009), S30–S38. doi: 10.1109/MCOM.2009.4804387.
- [91] INTERNET ASSIGNED NUMBERS AUTHORITY (IANA). *IANA Ethernet numbers assignment*. Februar 2010.
- [92] IEEE COMPUTER SOCIETY. *802.1D: IEEE Standard for Local and Metropolitan Area Networks—Media Access Control (MAC) Bridges*. 2004.
- [93] IEEE COMPUTER SOCIETY. *802.1Q: IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks*. 2005.

- [94] IEEE COMPUTER SOCIETY. *802.1X: IEEE Standard for Local and Metropolitan Area Networks—Port-Based Network Access Control*. Oktober 2001.
- [95] B. ABOBA, L. BLUNK, J. VOLLBRECHT, J. CARLSON und H. LEVKOWETZ. *Extensible Authentication Protocol (EAP)*. RFC 3748 (Proposed Standard). Juni 2004.
- [96] E. L. BROWN. *802.1X Port-Based Authentication*. Boston, MA, USA: Auerbach Publications, 2006.
- [97] J. GEIER. *Implementing 802.1X Security Solutions for Wired and Wireless Networks*. Wiley, 2008.
- [98] IEEE COMPUTER SOCIETY. *802.3ad: Amendment to Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications—Aggregation of Multiple Link Segments*. 2000.
- [99] IEEE COMPUTER SOCIETY. *802.3at: IEEE Standard for Local and Metropolitan Area Networks—Access Method and Physical Layer Specifications Amendment 3: Data Terminal Equipment (DTE) Power via the Media Dependent Interface (MDI) Enhancements*. 2009.
- [100] IEEE COMPUTER SOCIETY. *802.3af: IEEE Standard for Local and Metropolitan Area Networks—Local and Metropolitan Area Networks - Specific Requirements*. 2003.
- [101] D. BREUER und C. KARTES. “Strom aus dem Netzkabel”. In: *Funkschau – Kommunikationstechnik für Profis* 21 (Oktober 2009), S. 14–17.
- [102] J. RECH. *Ethernet. Technologien und Protokolle für die Computervernetzung*. 2. Aufl. Verlag Heinz Heise, 2007.
- [103] *IEC 61784-2 – Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3*. English. Geneva, Switzerland: International Electrotechnical Commission, Dezember 2007.
- [104] G. SANTNER. “Industrial Ethernet: 23 inkompatible Standards in einer IEC-Norm”. In: *Bulletin SEV/AES* (2009), S. 11–14.
- [105] J.-D. DECOTIGNIE. “Ethernet-based real-time and industrial communications”. In: *Proceedings of the IEEE*. Bd. 93. 6. Swiss Center for Electron. & Microtechnology, Neuchatel, Switzerland, Juni 2005, S. 1102–1117.
- [106] M. FELSER. “Real-Time Ethernet – Industry Prospective”. In: *Proceedings of the IEEE* 93.6 (Juni 2005), S. 1118–1129. DOI: 10.1109/JPROC.2005.849720.
- [107] OPEN DEVICE NET VENDORS ASSOCIATION (ODVA). *EtherNet/IP Library*. 2010.
- [108] S.-K. KWEON, K. G. SHIN und Q. ZHENG. “Statistical Real-time Communication over Ethernet for Manufacturing Automation Systems”. In: *Proceedings 5th IEEE Real-Time Technology and Applications Symposium*. 1999, S. 192–202.
- [109] M. ALVES, E. TOVAR und F. VASQUES. *Ethernet Goes Real-time: a Survey on Research and Technological Developments*. Techn. Ber. Polytechnic Institute of Porto, School of Engineering, Januar 2000.
- [110] E. VONNAHME, S. RUPING und U. RUCKERT. “Measurements in switched Ethernet networks used for automation systems”. In: *Proceedings of the IEEE International Workshop on Factory Communication Systems (WFCS)* (2000), S. 231–238. DOI: 10.1109/WFCS.2000.882554.

- [111] Y. SONG. “Time constrained communication over switched Ethernet”. In: *IFAC international conference on fieldbus systems and their applications*. November 2001, S. 152–159.
- [112] Y. SONG, A. KOUBAA und L. I. LORRAINE. “Switched Ethernet for real-time industrial communication: Modelling and message Buffering delay evaluation”. In: *4th IEEE International Workshop on Factory Communication Systems (WFCS)*. Vasteras, Sweden: Springer-Verlag, 2002, S. 27–30.
- [113] K. C. LEE und S. LEE. “Performance evaluation of switched Ethernet for real-time industrial communications”. In: *Comput. Stand. Interfaces* 24.5 (2002), S. 411–423. doi: [http://dx.doi.org/10.1016/S0920-5489\(02\)00070-3](http://dx.doi.org/10.1016/S0920-5489(02)00070-3).
- [114] J. CHEN, Z. WANG und Y. SUN. “Real-time capability analysis for switch industrial Ethernet traffic priority-based”. In: *Proceedings of the 2002 International Conference on Control Applications*. Bd. 1. 2002, S. 525–529. doi: 10.1109/CCA.2002.1040240.
- [115] T. SKEIE, S. JOHANNESSEN und O. HOLMEIDE. “The road to an end-to-end deterministic Ethernet”. In: *4th IEEE International Workshop on Factory Communication Systems (WFCS)*. 2002, S. 3–9. doi: 10.1109/WFCS.2002.1159694.
- [116] X. FAN, Z. WANG und Y. SUN. “How to guarantee factory communication with switched Ethernet: Survey of its emerging technology”. In: *IEEE 2002 28th Annual Conference of the Industrial Electronics Society (IECON)*. Bd. 3. November 2002, S. 2525–2530. doi: 10.1109/IECON.2002.1185371.
- [117] Z. WANG, Y.-Q. SONG, J.-M. CHEN und Y.-X. SUN. “Real Time Characteristics of Ethernet and Its Improvement”. In: *Proceedings of the 4th World Congress on Intelligent Control and Automation*. Bd. 2. Juni 2002, S. 1311–1318.
- [118] J. JASPERNEITE, P. NEUMANN, M. THEIS und K. WATSON. “Deterministic real-time communication with switched Ethernet”. In: *Proceedings of 4th IEEE International Workshop on Factory Communication Systems (WFCS)*. 2002, S. 28–30.
- [119] J. JASPERNEITE und P. NEUMANN. “How to gurantee realtime behavior using Ethernet”. In: *11th IFAC Symposium on Information Control Problems in Manufacturing (INCOM)*). April 2004.
- [120] J. LÖSER und H. HÄRTIG. “Low-Latency Hard Real-Time Communication over Switched Ethernet”. In: *Proceedings of th 6th Euromicro Conference on Real-Time Systems (ECRTS)*. Juni 2004, S. 13–22.
- [121] Q. ZHANG und W. ZHANG. “Priority scheduling in switched industrial Ethernet”. In: Juni 2005, S. 3366–3370. doi: 10.1109/ACC.2005.1470490.
- [122] M. AZAROV. *Approach to a Latency-bound Ethernet*. Techn. Ber. IEEE 802.1 AVB group, April 2006.
- [123] S. VARADARAJAN und T. CHIUH. “EtheReal: A host-transparent real-time fast Ethernet switch”. In: *Proceedings of the 6th International Conference on Network Protocols, 1998*. Oktober 1998, S. 12–21. doi: 10.1109/ICNP.1998.723721.
- [124] J. SOLOMON. *Applicability Statement for IP Mobility Support*. RFC 2005 (Proposed Standard). Oktober 1996.
- [125] ETHERCAT TECHNOLOGY GROUP. *Website of the EtherCAT Technology Group*. 2010.

- [126] J. POSTEL. *Internet Protocol*. RFC 791 (Standard). Updated by RFC 1349. September 1981.
- [127] J. POSTEL. *Transmission Control Protocol*. RFC 793 (Standard). Updated by RFC 3168. September 1981.
- [128] J. POSTEL. *User Datagram Protocol*. RFC 768 (Standard). August 1980.
- [129] R. FIELDING, J. GETTYS, J. MOGUL, H. FRYSTYK, L. MASINTER, P. LEACH und T. BERNERS-LEE. *Hypertext Transfer Protocol – HTTP/1.1*. RFC 2616 (Draft Standard). Updated by RFC 2817. Juni 1999.
- [130] J. POSTEL. *File Transfer Protocol specification*. RFC 765. Obsoleted by RFC 959. Juni 1980.
- [131] J. POSTEL und J. REYNOLDS. *File Transfer Protocol*. RFC 959 (Standard). Updated by RFCs 2228, 2640, 2773. Oktober 1985.
- [132] J. CASE, M. FEDOR, M. SCHOFFSTALL und J. DAVIN. *Simple Network Management Protocol (SNMP)*. RFC 1157 (Historic). Mai 1990.
- [133] D. PLUMMER. *Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware*. RFC 826 (Standard). November 1982.
- [134] R. DROMS. *Dynamic Host Configuration Protocol*. RFC 2131 (Draft Standard). Updated by RFC 3396. März 1997.
- [135] M. BRINKMANN. “Neue Wege in der industriellen Kommunikation”. In: *Elektronik* 14 (2008), S. 64–67.
- [136] S. RÜPING, E. VONNAHME und J. JASPERNEITE. “Analysis of Switched Ethernet Networks with different Topologies used in Automation Systems”. In: *In: Field Bus Technology*. Springer-Verlag, 1999, S. 351–358.
- [137] T. SAUTER. “The Industrial Communication Technology Handbook”. In: Hrsg. von R. ZURAWSKI. CRC Press, Februar 2005. Kap. Fieldbus Systems: History and Evolution, S. 1–39.
- [138] IEEE COMPUTER SOCIETY. *802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. 2003.
- [139] G. REICHART und E. SCHMIDT. “Anforderungen an die Elektronik Hardware im Kraftfahrzeug”. In: *VDI-Berichte Nr. 1688*. 2004, S. 235–242.
- [140] *ARINC Report 664 Aircraft Data Network – Part 7, Avionics Full-Duplex Switched Ethernet Network*. September 2009.
- [141] M. SCHUMACHER, J. JASPERNEITE und K. WEBER. “Optimierte Datenübertragung bei Echtzeit-Ethernet”. In: *Automation 2009 (VDI Mess- und Automatisierungstechnik)*. Juni 2009.
- [142] J. JASPERNEITE und M. SCHUMACHER. *Forschungsprojekt ESANA: Echtzeit-Ethernet für die Sensor/Aktorvernetzung (Schlussbericht)*. Techn. Ber. Institut Industrial IT der Hochschule Ostwestfalen-Lippe, Februar 2009.
- [143] F. KADERALI und W. POGUNTKE. *Graphen. Algorithmen. Netze. Grundlagen und Anwendungen in der Nachrichtentechnik*. Vieweg, 1995, S. 528.

- [144] T. OTTMANN und P. WIDMAYER. *Algorithmen und Datenstrukturen*. 3. Aufl. Spektrum Akademischer Verlag, 1996.
- [145] R. H. GÜTING und S. DIEKER. *Datenstrukturen und Algorithmen*. Bd. 2. Teuber, 2003.
- [146] R. DIESTEL. *Graphentheorie*. 3. Aufl. Springer, März 2006, S. 345.
- [147] S. O. KRUMKE und H. NOLTEMEIER. *Graphentheoretische Konzepte und Algorithmen*. 1. Aufl. Vieweg+Teubner, Mai 2005, S. 407.
- [148] M. GRÖTSCHEL. *Graphen- und Netzwerkalgorithmen*. Skript zur Vorlesung an der Technischen Universität Berlin. Juli 2006.
- [149] A. STEGER. *Diskrete Strukturen Band 1: Kombinatorik, Graphentheorie, Algebra*. 2. Aufl. Springer-Lehrbuch. Springer Berlin Heidelberg, 2007. DOI: 10.1007/978-3-540-46664-2.
- [150] U. GREMMELMAIER. “Methoden zur Planung von hierarchischen Hochgeschwindigkeitsnetzen (MAN)”. Diss. Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1996.
- [151] V. TURAU. *Algorithmische Graphentheorie*. 3. Aufl. Oldenbourg, Oktober 2009, S. 445.
- [152] H.-J. KREOWSKI. *Algorithmen auf Graphen*. Skript zur Vorlesung an der Universität Bremen. 2003.
- [153] A. V. AHO, J. E. HOPCROFT und J. D. ULLMAN. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Januar 1974, S. 470.
- [154] J. B. KRUSKAL. “On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem”. In: *Proceedings of the American Mathematical Society* 7.1 (Februar 1956), S. 48–50.
- [155] R. PRIM. “Shortest connection networks and some generalizations”. In: *Bell System Technical Journal* 36 (1957), S. 1389–1401.
- [156] P. W. FRANK K. HWANG DANA S. RICHARDS. *The Steiner Tree Problem*. Annals of Discrete Mathematics. Elsevier Science Publishing, September 1992, S. 240.
- [157] M. ZACHARIASEN. “Rectilinear full Steiner tree generation”. In: *Networks* 33 (1999), S. 125–143.
- [158] H.-J. PRÖMEL und A. STEGER. *The Steiner Tree Problem: A Tour through Graphs, Algorithms, and Complexity*. 1. Aufl. Vieweg Advanced Lectures in Mathematics 3528067624. Vieweg+Teubner, März 2002, S. 241.
- [159] D. DU und X. HU. *Steiner Tree Problems in Computer Communication Networks*. World Scientific Pub Co, Februar 2008, S. 376.
- [160] F. S. HILLIER und G. J. LIEBERMANN. *Operations Research: Einführung*. 5. Aufl. Oldenbourg, November 1996, S. 854.
- [161] T. H. CORMEN, C. E. LEISERSON, R. L. RIVEST und C. STEIN. *Introduction to Algorithms*. The MIT Press, 2001.
- [162] E. W. DIJKSTRA. “A note on two problems in connexion with graphs”. In: *Numerische Mathematik* 1.1 (Dezember 1959), S. 269–271.

- [163] R. BELLMAN. “On a routing problem”. In: *Quarterly of Applied Mathematics* 16.1 (1958), S. 87–90.
- [164] L. R. FORD. *Network flow theory*. Techn. Ber. Santa Monica, California: The RAND Corporation, 1956.
- [165] J. W. SUURBALLE. “Disjoint paths in a network”. In: *Networks* 4.2 (1974), S. 125–145.
- [166] J. W. SUURBALLE und R. E. TARJAN. “A quick method for finding shortest pairs of disjoint paths”. In: *Networks* 14.2 (1984), S. 325–336. doi: 10.1002/net.3230140209.
- [167] E. WOLLNER. “Analyse und Synthese zuverlässiger Netze”. In: *Mathematical Methods of Operations Research* 23.7 (November 1979), S. 225–250.
- [168] R. BHANDARI. *Survivable Networks: Algorithms for Diverse Routing*. 1. Aufl. Kluwer International Series in Engineering & Computer Science, 1999, S. 200.
- [169] R. W. FLOYD. “Algorithm 97: Shortest path”. In: *Communications fo the ACM* 5.6 (1962), S. 345. doi: <http://doi.acm.org/10.1145/367766.368168>.
- [170] S. WARSHALL. “A Theorem on Boolean Matrices”. In: *Journal of the ACM* 9.1 (1962), S. 11–12. doi: <http://doi.acm.org/10.1145/321105.321107>.
- [171] W. DOMSCHKE und A. DREXL. *Einführung in Operations Research*. 7. Aufl. Springer, 2007, S. 268.
- [172] L. SUHL und T. MELLOULI. *Optimierungssysteme: Modelle, Verfahren, Software, Anwendungen*. 2. Aufl. Springer, Juni 2009, S. 318.
- [173] W. KOHN und R. ÖZTÜRK. *Mathematik für Ökonomen – Ökonomische Anwendungen der linearen Algebra und Analysis mit Scilab*. Springer, September 2009, S. 352.
- [174] B. KORTE und J. VYGEN. *Kombinatorische Optimierung: Theorie und Algorithmen*. Springer, 2008, S. 675.
- [175] E. L. LAWLER. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart und Winston, 1976, S. 374.
- [176] A. HEINRICI. *Leistungsvergleich von Nachbarschaftssuchverfahren*. 1. Aufl. VWF Verlag für Wissenschaft und Forschung, 1996, S. 210.
- [177] D. WAGNER. “Kombinatorische Optimierung”. In: *Schriften zur Informatik und angewandten Mathematik*. 130. Rheinisch-Westfälische Technische Hochschule Aachen, 1988, S. 79.
- [178] C. H. PAPADIMITRIOU und K. STEIGLITZ. *Combinatorial Optimization: Algorithms and Complexity*. 2. Aufl. Dover Publications, Inc., 1998, S. 496.
- [179] K. RICHTER, P. BACHMANN und S. DEMPE. *Diskrete Optimierungsmodelle. Effektive Algorithmen und näherungsweise Lösung*. Bd. 1. Verlag Technik Berlin, 1988, S. 188.
- [180] H. STREIM. “Heuristische Lösungsverfahren Versuch einer Begriffsklärung”. In: *Mathematical Methods of Operations Research* 19.5 (Oktober 1975), S. 143–162. doi: 10.1007/BF01999748.
- [181] H. MÜLLER-MERBACH. “Heuristics and their design: a survey”. In: *European Journal of Operational Research* 8.1 (September 1981), S. 1–23.
- [182] E. A. FEIGENBAUM und J. FELDMAN, Hrsg. *Computers and Thought*. Bd. 1. AAAI Press, 1963, S. 550.

- [183] N. METROPOLIS, A. W. ROSENBLUTH, M. N. ROSENBLUTH, A. H. TELLER und E. TELLER. “Equation of State Calculations by Fast Computing Machines”. In: *The Journal of Chemical Physics* 21.6 (1953), S. 1087–1092. doi: 10.1063/1.1699114.
- [184] E. AARTS, J. KORST und W. MICHIELS. “Handbook of Approximation Algorithms and Metaheuristics”. In: Hrsg. von T. F. GONZALEZ. Chapman & Hall, 2007. Kap. Simulated Annealing, S. 25–1–25–11.
- [185] S. KIRKPATRICK, C. D. GELATT, JR. und M. P. VECCHI. “Optimization by Simulated Annealing”. In: *Science* 220 (1983), S. 671–680.
- [186] V. ČERNÝ. “Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm”. In: *Journal of Optimization Theory and Applications* 45.1 (Januar 1985), S. 41–51. doi: 10.1007/BF00940812.
- [187] D. S. JOHNSON, C. R. ARAGON, L. A. MCGEOCH und C. SCHEVON. “Optimization by Simulated Annealing: An Experimental Evaluation; Part I, Graph Partitioning”. In: *Operations Research* 36.6 (1989), S. 865–892. doi: 10.1287/opre.37.6.865.
- [188] R. V. V. VIDAL. *Applied Simulated Annealing*. Hrsg. von R. V. V. VIDAL. 1. Aufl. Springer, 1993, S. 358.
- [189] W. MICHIELS, E. H. L. AARTS und J. KORST. *Theoretical Aspects of Local Search*. Monographs in Theoretical Computer Science. Springer Berlin Heidelberg, 2007, S. 238. doi: 10.1007/978-3-540-35854-1.
- [190] U. FAIGLE und R. SCHRADER. “Simulted Annealing – Eine Fallstudie”. In: *Angewandte Informatik* 30.6 (1988), S. 259–263.
- [191] N. COLLINS, R. EGGLESE und B. GOLDEN. “Simulated Snnealing – An annotated bibliography”. In: *American Journal of Mathematical and Management Sciences* 8.3-4 (1988), S. 209–307.
- [192] P. C. FETTEROLF und G. ANANDALINGAM. “Optimizing Interconnections of Local Area Networks: An Approach Using Simulated Annealing”. In: *ORSA Journal of Computing* 3.4 (1991), S. 275–287.
- [193] P. CHARDAIRE und J. L. LUTTON. “Applied Simulated Annealing”. In: Hrsg. von R. V. V. VIDAL. Springer, 1993. Kap. Using Simulated annealing to solve concentrator location problems in telecommunication networks, S. 173–199.
- [194] K. ANDERSEN, R. V. V. VIDAL und V. B. IVERSEN. “Applied Simulated Annealing”. In: Hrsg. von R. V. V. VIDAL. Springer, 1993. Kap. Design of a Teleprocessing Communication Network using Simulated Annealing, S. 201–215.
- [195] M. RANDALL, G. McMAHON und S. SUGDEN. “A Simulated Annealing Approach to Communication Network Design”. In: *Journal of Combinatorial Optimization* 6.1 (März 2002), S. 55–65.
- [196] G. DUECK und T. SCHEUER. “Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing”. In: *Journal of Computational Physics* 90.1 (1990), S. 161–175. doi: 10.1016/0021-9991(90)90201-B.
- [197] G. DUECK. “New Optimization Heuristics: The Great Deluge Algorithm and the Record-to-Record Travel”. In: *Journal of Computational Physics* 104.1 (1993), S. 86–92. doi: DOI:10.1006/jcph.1993.1010.

- [198] G. DUECK. “Abenteuer Optimierung – trocken in der Sintflut”. In: *Bild der Wissenschaft* 5 (1994), S. 74–77.
- [199] G. DUECK, H.-M. WALLMEIER und T. SCHEUER. “Toleranzschwelle und Sintflut: neue Ideen zur Optimierung”. In: *Spektrum der Wissenschaft (German Ed. of Scientific American)* 3 (März 1993), S. 42–51.
- [200] F. GLOVER. “Future paths for integer programming and links to artificial intelligence”. In: *Computers & OR* 13.5 (1986), S. 533–549.
- [201] F. GLOVER. “Tabu Search – Part I”. In: *ORSA Journal on Computing* 1.3 (1989), S. 190–206. doi: 10.1287/ijoc.1.3.190.
- [202] F. GLOVER. “Tabu Search – Part II”. In: *ORSA Journal on Computing* 2.1 (1990), S. 4–32. doi: 10.1287/ijoc.2.1.4.
- [203] P. HANSEN. *The Steepest Ascent Mildest Descent Heuristic for Combinatorial Programming*. presented at Congress on Numerical Methods in Combinatorial Optimization, Capri, Italy. 1986.
- [204] F. GLOVER und M. LAGUNA. *Tabu Search*. Kluwer Academic Publisher, 1997, S. 382.
- [205] F. GLOVER. “Tabu Search: A Tutorial”. In: *Interfaces* 20.4 (Juli 1990), S. 74–94.
- [206] F. GLOVER, M. LAGUNA und R. MARTÍ. “Handbook of Approximation Algorithms and Metaheuristics”. In: Hrsg. von T. F. GONZALEZ. Chapman & Hall, 2007. Kap. Principles of Tabu Search, S. 23–1–23–12.
- [207] F. GLOVER, M. LEE und J. RYAN. “Least-cost network topology design for a new service”. In: *Annals of Operations Research* 33.5 (Mai 1991), S. 351–362.
- [208] E. COSTAMAGNA, A. FANNIB und G. GIACINTO. “A Tabu Search algorithm for the optimisation of telecommunication networks”. In: *European Journal of Operational Research* 106.2-3 (April 1998), S. 357–372. doi: 10.1016/S0377-2217(97)00279-8.
- [209] J. PAULLI. “Applied Simulated Annealing”. In: Hrsg. von R. V. V. VIDAL. Springer, 1993. Kap. A Computational Comparison of Simulated Annealing and Tabu Search Applied to the Quadratic Assignment Problem, S. 85–102.
- [210] H. BENKER. *Mathematische Optimierung mit Computeralgebrasystemen: Einführung für Ingenieure, Naturwissenschaftler und Wirtschaftswissenschaftler*. 1. Aufl. Springer, 2003, S. 499.
- [211] W. DÜCK. *Optimierung unter mehreren Zielen*. Braunschweig: Vieweg, 1979, S. 104.
- [212] M. EHRGOTT. *Multicriteria Optimization*. 2. Aufl. Springer, 2005, S. 323.
- [213] A. TRESCH. *Maschinelles Lernen*. Skript zur Vorlesung an der Universität Mainz. 2007.
- [214] L. FAHRMEIR, A. HAMERLE und G. TUTZ. *Multivariate statistische Verfahren*. 2. Aufl. Gruyter, April 1996, S. 902.
- [215] F. ROTHLAUF. *Design of Modern Heuristics – A Guide to Efficient and Effective Problem Solving*. 1. Aufl. draft (not published). Springer, April 2008.
- [216] I. FORKEL, M. SCHINNENBURG und M. ANG. “Generation of Two-Dimensional Correlated Shadowing for Mobile Radio Network Simulation”. In: *Proceedings of The 7th International Symposium on Wireless Personal Multimedia Communications (WPMC)*. Abano Terme (Padova), Italy, September 2004, S. 5.

- [217] J.-P. VASSEUR, M. PICKAVET und P. DEMEESTER. *Network Protection and Restoration Techniques – Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. Morgan Kaufmann, 2004, S. 544.
- [218] M. R. GAREY, R. L. GRAHAM und D. JOHNSON. “The complexity of computing Steiner minimal trees”. In: *SIAM Journal on Applied Mathematics* 31.4 (Juni 1977), S. 835–859.
- [219] K.-M. CHAO und B. Y. WU. *Spanning Trees and Optimization Problems*. CRC Press, 2004, S. 2000.
- [220] F. ROTHLAUF. “On Optimal Solutions for the Optimal Communication Spanning Tree Problem”. In: *Operations Research* 57.2 (März 2009), S. 413–425. doi: 10.1287/opre.1080.0592.
- [221] M. R. GAREY und D. S. JOHNSON. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, April 1979, S. 340.
- [222] A. CAYLEY. *The Collected Mathematical Papers of Arthur Cayley*. BiblioBazaar, Mai 2009, S. 148.
- [223] W. DOMSCHKE und A. DREXL. *Logistik Band 3: Standorte*. 4. Aufl. Oldenbourg, 1995, S. 276.
- [224] D. SCHÖNLEIN. *Planung und Optimierung eines Versorgungsnetzes: Die Suche nach der optimalen Kombination aus Standortanzahl, Tourenlänge und Personalbedarf*. Vdm Verlag Dr. Müller, Januar 2009, S. 88.
- [225] M. PIÓRO und D. MEDHI. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers, 2004, S. 800.
- [226] R. R. BOORSTYN und H. FRANK. “Large-Scale Network Topological Optimization”. In: *IEEE Transactions on Communications* 25.1 (Januar 1977), S. 29–47.
- [227] A. NOLTEMEIER. “Modelle und Lösungsverfahren zur Generierung von Flugplänen”. Diss. Universität zu Köln, Mai 2000.
- [228] M. GERLA und L. KLEINROCK. “On the Topological Design of Distributed Computer Networks”. In: *IEEE Transactions on Communications* 25.1 (Januar 1977), S. 48–60.
- [229] B. Y. WU, K.-M. CHAO und C. Y. TANG. “Approximation Algorithms for Some Optimum Communication Spanning Tree Problems”. In: *Algorithms and Computation* 1533/1998 (1998), S. 408–416. doi: 10.1007/3-540-49381-6.
- [230] P. SHARMA. “Algorithms for the optimum communication spanning tree problem”. In: *Annals of Operations Research* 143.1 (März 2006), S. 203–209. doi: 10.1007/s10479-006-7382-1.
- [231] A. BALAKRISHNAN. “LP extreme points and cuts for the fixed-charge network design problem”. In: *Mathematical Programming* 39.3 (Oktober 1987), S. 263–284. doi: 10.1007/BF02592077.
- [232] K. HOLMBERG und D. YUAN. “Optimization of Internet Protocol network design and routing”. In: *Networks* 43.1 (Januar 2004), S. 39–53.
- [233] T. L. MAGNANTI, P. MIRCHANDANI und R. VACHANI. “Modeling and Solving the Two-Facility Capacitated Network Loading Problem”. In: *Operations Research* 43.1 (1995), S. 142–157.

- [234] T. G. ROBERTAZZI. *Planning Telecommunications Networks*. IEEE Press, Dezember 1998, S. 204.
- [235] S. GUPTA und K. W. ROSS. “Performance modeling and optimization of interconnected Ethernets”. In: *Proceedings of INFOCOM '91*. Bd. 3. April 1991, S. 1353–1359. doi: 10.1109/INFCOM.1991.147662.
- [236] D. ARNOLD, H. ISERMANN, A. KUHN, H. TEMPELMEIER und K. FURMANS, Hrsg. *Handbuch Logistik*. 3. Aufl. Springer, April 2008, S. 1137.
- [237] J. F. CAMPBELL, A. T. ERNST und M. KRISHNAMOORTHY. “Hub Location Problems”. In: *Facility Location: Application and Theory*. Hrsg. von Z. DREZNER und H. W. HAMACHER. Springer-Verlag, 2002. Kap. 12, S. 373–407.
- [238] A. MERCHANT und B. SENGUPTA. “Assignment of cells to switches in PCS networks”. In: *IEEE/ACM Transactions on Networking* 3.5 (Oktober 1995), S. 521–526. doi: <http://dx.doi.org/10.1109/90.469954>.
- [239] M. J. ATALLAH und M. BLANTON. *Algorithms and Theory of Computation Handbook*. 2. Aufl. CRC Press, November 2009, S. 1312.
- [240] C. L. MONMA und D. F. SHALLCROSS. “Methods for Designing Communications Networks with Certain Two-Connected Survivability Constraints”. In: *Operations Research* 37.4 (Juli 1989), S. 531–541. doi: 10.1287/opre.37.4.531.
- [241] KONRAD-ZUSE-ZENTRUM FÜR INFORMATIONSTECHNIK BERLIN. *Solving Constraint Integer Programs (SCIP) Website*. <http://scip.zib.de>. Mai 2010.
- [242] T. ACHTERBERG. “SCIP: solving constraint integer programs”. In: *Journal Mathematical Programming Computation* 1.1 (2009), S. 1–41. doi: 10.1007/s12532-008-0001-1.
- [243] S. SKIENA. *GeoSteiner: Software for Computing Steiner Trees*. <http://www.diku.dk/hjemmesider/ansatte/martinz/geosteiner>. 2010.
- [244] P. J. KÜHN. *Performance Modelling and Simulation*. Skript zur Vorlesung an der Universität Stuttgart. 2010.
- [245] INSTITUTE OF COMMUNICATION NETWORKS AND COMPUTER ENGINEERING. *IKR Simulation and Emulation Library*. <http://www.ikr.uni-stuttgart.de/IKRSimLib>. 2010.
- [246] THE APACHE SOFTWARE FOUNDATION. *Batik SVG Toolkit*. <http://xmlgraphics.apache.org/batik>. 2010.
- [247] D. GILBERT. *JFreeChart*. <http://www.jfree.org/jfreechart>. 2010.
- [248] P. WENDYKIER. *JTransforms*. <http://sites.google.com/site/piotrwendykier/software/jtransforms>. 2010.

Danksagung

Die vorliegende Dissertation entstand während meiner Tätigkeit als Wissenschaftlicher Angestellter am Institut für Kommunikationsnetze und Rechnersysteme an der Universität Stuttgart. Während meiner sechsjährigen Tätigkeit am Institut konnte ich mich persönlich, fachlich und methodisch weiterentwickeln und wertvolle Erfahrungen sammeln. Hierfür möchte ich mich bei allen Institutsangehörigen recht herzlich bedanken.

Mein außerordentlicher Dank gilt vor allem Herrn Prof. em. Dr.-Ing. Dr. h.c. mult. Paul J. Kühn für die wissenschaftliche Betreuung und Förderung, für seine Unterstützung und für die Übernahme des Hauptberichtes.

Bei Herrn Prof. Dr.-Ing. Rudi Knorr und Herrn Prof. Dr.-Ing. Andreas Kirstädter möchte ich mich für die Übernahme der Mitberichte bedanken.

Allen wissenschaftlichen Kolleginnen und Kollegen des Instituts möchte ich für die gute und professionelle Zusammenarbeit danken. Insbesondere möchte ich mich bei Elias A. Doumith und Andreas Reifert für die ergebnisreichen Diskussionen und die gemeinsamen Veröffentlichungen im Bereich der Optimierung bedanken. Auch möchte ich mich bei Martin Köhn und Detlef Saß für zahlreiche Diskussionen und Anregungen bedanken. Marc Barisch, Joachim Scharf, Frank Feller und Ulrich Gemkow möchte ich für das Lesen einzelner Kapitel des Manuskriptes und die wertvollen Kommentare danken. Ulrich Gemkow möchte ich darüber hinaus für die Betreuung, die anvertrauten Dienstaufgaben und die eingeräumten Freiheiten zur Forschung und in der Lehre recht herzlich danken.

Die in dieser Arbeit vorgestellten Verfahren wurden teilweise in mehreren studentischen Arbeiten und von studentischen Hilfskräften implementiert und getestet. In diesem Zusammenhang möchte ich mich besonders bei Domenic Teuchert, Benjamin Fischbach und Quentin Duval für ihr überdurchschnittliches Engagement und ihre Beiträge bedanken.

Viele Denkanstöße habe ich in bilateralen Industrieprojekten mit Automobilherstellern erhalten. Besonders danken möchte ich Herrn Dr.-Ing. Volker Feil für die vertrauensvolle Zusammenarbeit, die anregenden Diskussionen und die gemeinsamen Veröffentlichungen.

Frau Hildegard Walter möchte ich für die sprachliche Durchsicht des Manuskriptes und für die nützlichen Hinweise danken.

Ganz besonders möchte ich mich bei meiner Familie, vor allem bei meiner Frau Laura, für die Geduld, Unterstützung und Motivation in den letzten Jahren und insbesondere während dem Anfertigen dieser Dissertation bedanken.

