



FAKULTÄT II – INFORMATIK, WIRTSCHAFTS- UND RECHTSWISSENSCHAFTEN
DEPARTMENT FÜR INFORMATIK

Selbstorganisierte Einsatzplanung dezentraler Akteure im Smart Grid

Dissertation zur Erlangung des Grades eines
Doktors der Naturwissenschaften

vorgelegt von
Christian Hinrichs

Gutachter:
Prof. Dr. M. Sonnenschein
Prof. Dr. S. Lehnhoff
Prof. Dr. W. Reif

Tag der Disputation: 15. Mai 2014

Dieses Dokument wurde unter Verwendung von KOMA-Script und $\LaTeX$ gesetzt in Bitstream Charter (11 pt, Fließtext und mathematischer Formelsatz) sowie Kp-Fonts (für serifenlose Elemente wie Überschriften und Akronyme). Die visuelle Gestaltung basiert auf *ClassicThesis* und *ArsClassica*, wurde jedoch an den eigenen Geschmack angepasst. Das Literaturverzeichnis wurde von *biblatex* in Verbindung mit *Biber* generiert, während für das Akronym- und das Symbolverzeichnis *nomencl* eingesetzt wurde. Für Zeichnungen kam *PGF/TikZ* zum Einsatz, Diagramme wurden mit *matplotlib* unter *Python* erzeugt. Die verwendeten typografischen Richtlinien zum mathematischen Formelsatz entstammen [FD04].

In der elektronischen Version dieses Dokumentes sind alle Akronyme sowie alle wichtigen Symbole (d. h. solche, die an mehreren Textstellen auftreten) zur leichteren Nachvollziehbarkeit mit einem *Tooltip* ausgestattet, welcher die jeweilige Definition des Akronyms/Symbols zeigt.

Christian Hinrichs
Oldenburg, 6. Januar 2014

Zusammenfassung

Verteilte kombinatorische Optimierungsprobleme zeichnen sich dadurch aus, dass die einem Problem zugrundeliegenden Entscheidungsvariablen von verteilten, autonomen Prozessen kontrolliert werden. Aufgrund von Abhängigkeiten (Constraints) zwischen den Variablenbelegungen müssen die Entscheidungen der Prozesse im Sinne einer zielführenden Optimierung mittels Informationsaustausch untereinander koordiniert werden. Diese Situation ist beispielsweise in der prädiktiven, fahrplanbasierten Einsatzplanung autonomer, dezentraler Energiewandlungsanlagen zu finden. Hier muss für jede beteiligte Anlage ein Fahrplan bestimmt werden, sodass der resultierende Einsatzplan der Anlagen einen zuvor vereinbarten Wirkleistungsverlauf produziert. In einem vollständig verteilten Koordinationsparadigma sucht somit jede Anlage in dem ihr eigenen, durch lokale Nebenbedingungen eingeschränkten Suchraum nach derjenigen Teillösung, welche unter Berücksichtigung aller anderen beteiligten Anlagen das globale Optimierungsziel (den Wirkleistungsverlauf) erfüllt. Der resultierende Einsatzplan aller Anlagenfahrpläne stellt damit einen Konsens unter allen Anlagen dar.

Der in dieser Arbeit entwickelte Lösungsansatz basiert auf der Anwendung von Selbstorganisationsmechanismen und ermöglicht eine verteilte kombinatorische Optimierung in einer asynchronen Kommunikationsumgebung. Die resultierende Heuristik COHDA (*Combinatorial Optimization Heuristic for Distributed Agents*) ist in der Lage, eine vollständig verteilte Konsensbildung dezentraler Akteure durchzuführen, ohne zentrale Komponenten oder Hierarchien zu benötigen. Die garantierte Konvergenz und Terminierung des Verfahrens wird formal hergeleitet. Simulationsstudien belegen die Robustheit sowie Skalierbarkeit des Ansatzes und zeigen die Einsatzfähigkeit in verschiedenen Smart Grid Planungsszenarien.

Abstract

In decentralized systems, where the search space of a given optimization problem is distributed into disjoint subspaces, centralized optimization approaches often cannot be applied. For example, the global collection of data might violate privacy considerations or bandwidth restrictions. The gathering of such data might even be impossible, as it is the case if local search spaces are partially unknown or cannot be enumerated (i. e. due to infiniteness). Another limitation is that distributed search spaces are often not independent. Such interdependencies require to evaluate search spaces with relation to each other. For instance, this type of problem is present in the transition of today's electricity grid to a decentralized smart grid. Here, we have to cope with an increasing number of distributed energy resources (DER). The thesis focuses on day-ahead planning of the provision of active power for these DER, which can be expressed as a distributed combinatorial problem: Given a set of DER and a global target power profile, each unit has to select its own schedule for the planning horizon in such a way, that the schedules of all units jointly match the global target profile as close as possible.

For this purpose, the heuristic COHDA (*Combinatorial Optimization Heuristic for Distributed Agents*) is developed. The approach is based on self-organization strategies and yields a distributed combinatorial optimization process in an asynchronous communication environment. Central components for data storage or coordination purposes are not necessary. The correctness of COHDA with respect to convergence and termination is proven formally. Simulation experiments show the robustness and scalability of the approach. Finally, the effectiveness of COHDA in a number of smart grid planning scenarios is shown.

INHALTSVERZEICHNIS

Akronyme	XIII
Symbole	XV
1 Einleitung	1
1.1 Dezentrale Kontrollstrukturen	3
1.2 Dezentral koordinierte Wirkleistungsbereitstellung	6
1.3 Verwandte Arbeiten	10
1.3.1 COBB	11
1.3.2 Stigspace	12
1.3.3 Restriktionslose Gerätepools	13
1.3.4 Holonische virtuelle Kraftwerke	14
1.3.5 AVPP	15
1.3.6 EPOS	16
1.3.7 Stand der Technik (Zusammenfassung)	17
1.4 Zielsetzung der Arbeit	18
1.5 Gang der Untersuchung	19
1.6 Eigene Vorarbeiten	20
2 Grundlagen	23
2.1 Verteilte kombinatorische Optimierung	23
2.1.1 Restriktionen	25
2.1.2 Low-Arity Constraints	27
2.1.3 High-Arity Constraints	27
2.1.4 Koordinationsparadigmen	27
2.2 Multiagentensysteme	29
2.3 Selbstorganisation	31
2.3.1 Nichtlinearität	32

2.3.2	Variation	33
3	Selbstorganisierte Einsatzplanung	35
3.1	Das prädiktive Einsatzplanungsproblem	38
3.1.1	Ganzzahliges lineares Optimierungsproblem	45
3.2	Agenten	47
3.2.1	Kollaboration	48
3.2.2	Wettbewerb	48
3.2.3	Fähigkeiten	50
3.3	Kommunikation	51
3.4	Planung des Systemdesigns	52
3.5	Globale Systemfunktion	53
3.6	Paradigma #1: Lokale Verhaltensregeln	54
3.6.1	Designschritt 1: Iterative Annäherung	54
3.6.2	Designschritt 2: Overlay-Netz	56
3.6.3	Designschritt 3: Konsens	59
3.6.4	Designschritt 4: Freie Gesinnungswahl	67
3.7	Paradigma #2: Implizite Koordination	71
3.8	Paradigma #3: Adaptivität	72
3.8.1	Neuaufwurf der Planung	74
3.8.2	Umplanung	75
3.9	Paradigma #4: Reduzierte Zustandsinformationen	76
4	Heuristik: COHDA	79
4.1	Start der Heuristik	79
4.2	Verhalten der Heuristik	81
4.3	Terminierung der Heuristik	83
4.4	Verhalten der Agenten	84
4.4.1	Pflichten	84
4.4.2	Freiheiten	87
4.5	Fairness	90

5	Konvergenz des Verfahrens	93
5.1	Shared Memory Model	94
5.1.1	Erste Konvergenzstufe	95
5.1.2	Zweite Konvergenzstufe	95
5.1.3	Dritte Konvergenzstufe	101
5.2	Message Passing Model	103
5.2.1	Asynchrones Modell mit fester Nachrichtenverzögerung	103
5.2.2	Asynchrones Modell mit variabler Verzögerung	106
5.3	Konvergenz bei freier Gesinnungswahl	107
5.4	Konvergenzgeschwindigkeit	109
5.5	Finale Lösungsgüte	111
6	Umsetzung	113
6.1	Simulationsumgebung	113
6.2	Kommunikationsmodell	114
6.3	Prozessmodell	114
6.4	Suchraum und Suchraummodell	117
6.4.1	Enumeration mit sequenzieller Suche	117
6.4.2	Maschinelles Lernverfahren	119
6.5	Umsetzung für reale Anlagen	122
7	Evaluation	125
7.1	Synthetische Probleminstanzen	127
7.1.1	Konstruktion	129
7.1.2	Lösungsraum	132
7.2	Laufzeitverhalten	140
7.3	Robustheit	143
7.3.1	Topologie	143
7.3.2	Nachrichtenverzögerung	147
7.3.3	Reaktionszeit	149
7.3.4	Freie Gesinnungswahl	151
7.4	Skalierbarkeit	155
7.4.1	Verbundgröße	158
7.4.2	Suchraumgröße	161

7.4.3	Planungshorizont	161
7.5	Regressionsanalyse	164
7.5.1	Topologie	167
7.5.2	Nachrichtenverzögerung	168
7.5.3	Reaktionszeit	171
7.5.4	Verbundgröße	171
7.5.5	Suchraumgröße	174
7.5.6	Planungshorizont	174
7.6	Domänenspezifische Szenarien	176
7.6.1	Mikro-Blockheizkraftwerke	177
7.6.2	Einfluss der Verbundgröße	180
7.6.3	Suchraummodelle	182
7.6.4	Auktionshandel EPEX SPOT	186
7.6.5	Spreizungsreduktion Tageslastprofil	197
7.7	Vergleich mit existierenden Verfahren	204
8	Schluss	209
8.1	Ergebnisse	209
8.2	Anknüpfender Forschungsbedarf	215
A	Synthetische Problem instanzen	221
	Abbildungsverzeichnis	229
	Literatur	231

AKRONYME

ALMA	Adaptive Load Management by Aggregation
AVPP	Autonomous Virtual Power Plant
BDEW	Bundesverband der Energie- und Wasserwirtschaft
BHKW	Blockheizkraftwerk
CIM	Common Information Model
COBB	Constraint Optimization by Broadcasting
COHDA	Combinatorial Optimization Heuristic for Distributed Agents
DCOP	Distributed Constraint Optimization Problem
dena	Deutsche Energie-Agentur GmbH
DER	Distributed Energy Resource
DIN	Deutsches Institut für Normung e. V.
DO-MC-COP	Distributed-Objective Multiple-Choice Combinatorial Optimization Problem
EEG	Gesetz für den Vorrang Erneuerbarer Energien
EPOS	Energy Plan Overlay Self-stabilization
FIPA	Foundation for Intelligent Physical Agents
GI	Gesellschaft für Informatik
ID	Identifizier (alphanumerischer Bezeichner eines Verbundteilnehmers)

IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IQR	Interquartilsabstand
KWK	Kraft-Wärme-Kopplung
KWKG	Gesetz für die Erhaltung, die Modernisierung und den Ausbau der Kraft-Wärme-Kopplung
MAS	Multiagentensystem
MC-COP	Multiple-Choice Combinatorial Optimization Problem
MC-KP	Multiple-Choice Knapsack Problem
MC-SSP	Multiple-Choice Subset-Sum Problem
MOCO	Multiple Objective Combinatorial Optimization
MWK	Niedersächsisches Ministerium für Wissenschaft und Kultur
NP	Nichtdeterministisch-Polynomiell
OFFIS	Oldenburger Forschungs- und Entwicklungsinstitut für Informatik
OPC	OPC Foundation
OPC UA	OPC Unified Architecture
P2P	Peer-to-Peer Netz
PEP	Prädiktives Einsatzplanungsproblem
SCADA	Supervisory Control and Data Acquisition
SO	Selbstorganisation
VK	Virtuelles Kraftwerk
WP	Wärmepumpe

SYMBOLE

Allgemein

P_{el} Elektrische Wirkleistung

P_{th} Thermische Leistung

Prädiktives Einsatzplanungsproblem ([Kapitel 3](#))

a Verbundteilnehmer

$\hat{a}$ Verbundsprecher

A Verbund

C Kommunikationsnetz

E Kantenmenge

f Globale Gütefunktion

g Erfüllungsfunktion bzgl. lokaler Präferenzen

N Nachbarschaft

O Overlay-Netz

t Planungsintervall

T Planungshorizont

v Güte

Δ	Störung
γ	Lösungskandidat
κ	Arbeitsgedächtnis
K	Schlüsselmenge einer Konfiguration
λ	Fahrplan-Zähler
μ	Nachricht
ω	Fahrplanauswahl
Ω	Konfiguration / wahrgenommener Systemzustand
π	Wirkleistungsverlauf
σ	Suchraummodell
S	Suchraum
θ	Fahrplan
Θ	Verbundfahrplan
ζ	Wirkleistungsprodukt

Konvergenz ([Kapitel 5](#))

$\mathcal{A}$	Prädikat einer Konvergenzstufe
c	Ausführungsfolge
d	Verzögerung einer Nachricht
e	Schleifenausführung
E	Parallele Schleifenausführungen
γ^*	Finaler Lösungskandidat

h	Gütesumme für Ausführungsfolge
L	Lösungsraum
r	Reaktionszeit
ρ	Periode
s	Zeitschritt
ts	Folge von Zeitschritten
VF	Potenzialfunktion für Terminierung

Evaluation ([Kapitel 7](#))

e	Normalisierter Fehler
$\tilde{e}$	Median der normalisierten Fehlerwerte
h	Lage eines Zielvektors
m	Größe eines Verbundes
n	Größe eines Suchraumes
p	Profitwert
q	Länge eines Planungshorizontes
R^2	Bestimmtheitsmaß
$\mathbf{s}$	Lösungsvektor
$\mathbf{w}$	Gewichtsvektor
$\mathbf{z}$	Zielvektor
β	Regressionskoeffizient
ψ_{dyn}	Dynamische Gesinnungswahl

ψ_{stat} Statische Gesinnungswahl

ϕ Small-World Dichte

1

EINLEITUNG

Das Energieversorgungssystem stellt als kritische Infrastruktur einen Eckpfeiler der modernen Gesellschaft dar. Daher ist es nicht verwunderlich, dass die derzeit stattfindende Energiewende auch als „Übergang in ein neues Elektrizitätszeitalter“ sowie als „dritte industrielle Revolution in der Energietechnik“ bezeichnet wird [ETG12]. Eine wesentliche Rolle spielen dabei dezentrale Energiewandlungsanlagen, welche oftmals aus erneuerbaren Energiequellen in Form von bspw. Solar- und Windenergie gespeist werden und zukünftig die konventionellen Großkraftwerke ersetzen sollen. Um diesen Trend zu fördern, werden aktuell von politischer Seite aus große Anstrengungen unternommen. So schreibt das Europäische Parlament/der Rat der Europäischen Union mit der Richtlinie 2009/28/EG die sog. „20-20-20-Ziele“ vor, in welchen mitunter gefordert wird, dass bis 2020 mindestens 20 % des Bruttoendenergieverbrauchs der Europäischen Gemeinschaft durch Energie aus erneuerbaren Quellen gedeckt wird [EU09]. In Deutschland wird dies im Nationalen Aktionsplan für erneuerbare Energie des Bundesumweltministeriums entsprechend gewürdigt [BMU10]. Mit dem Energiekonzept der Bundesregierung vom 28. September 2010 wurden diese Ziele auf nationaler Ebene nochmals erhöht, so soll der Anteil der Stromerzeugung aus erneuerbaren Energien am Bruttostromverbrauch bis 2020 bei 35 % liegen [BD10]. Langfristig ist diese Abwendung von der zentralisierten Großenergieerzeugung sowohl aus ökologischer als auch aus ökonomischer Sicht ein sinnvoller Schritt, da hiermit einerseits die Abhängigkeit von fossilen Quellen reduziert, sowie andererseits die Sicherheits- und Abfallproblematik der Kernspaltung vermieden wird. Aufgrund ihrer volatilen, zumeist nicht steuerbaren Verfügbarkeit stellen die regenerativen Energien jedoch immense Herausforderungen an die Betriebsführung der Netze, da elektrische Energie nicht effizient und ökonomisch sinnvoll in großen Mengen speicherbar ist [Leh10] und somit auf andere Weise die für das Netz zu jedem Zeitpunkt erforderliche Balance zwischen Einspeisung und Entnahme von

elektrischer Energie gewährleistet werden muss. Die Fachwelt ist sich einig, dass die Modernisierung des Elektrizitätsnetzes zu einem *Smart Grid* mit einer deutlichen Flexibilisierung einhergehen muss [KAS10; ETG12]. Hierzu ist es nicht ausreichend, eine rein technische Integration der dezentralen Energiewandlungsanlagen umzusetzen, es ist auch die organisatorische Integration in energiewirtschaftliche Beschaffungs- und Handelsprozesse erforderlich [Trö10]. Der Bundesverband der Energie- und Wasserwirtschaft (BDEW) sieht als Dachverband der Energieversorgungsunternehmen hierin sogar den Kerngedanken eines Smart Grid und fordert zur Nutzbarmachung der Flexibilität die Umsetzung eines umfassenden kommunikationstechnischen Energieinformationssystems im Verteilnetz bis 2018 [AA13]. Eine besondere Bedeutung kommt in diesem Kontext denjenigen Akteuren zu, welche ihren Betrieb hinsichtlich systemstabilisierender Maßnahmen adaptiv steuern können. Die Deutsche Energie-Agentur GmbH (dena) schätzt, dass durch den Einsatz des Lastmanagements (der gezielten Einflussnahme auf den Verbrauch von elektrischer Energie) hinsichtlich der Reduktion von Spitzenlasten eine Verminderung an benötigter Regelleistung um 60 % und damit eine Senkung der volkswirtschaftlichen Kosten der Stromerzeugung bis 2020 um 481 Mio.€ erreicht werden kann [KAS10]. Noch stärkere Aufmerksamkeit erfahren momentan steuerbare Kleinerzeuger wie Anlagen zur Kraft-Wärme-Kopplung (KWK-Anlagen), deren beschleunigter Ausbau von politischer Seite etwa durch das Gesetz für den Vorrang Erneuerbarer Energien (EEG) und das Gesetz für die Erhaltung, die Modernisierung und den Ausbau der Kraft-Wärme-Kopplung (KWKG) vorangetrieben wird [BD12; BD13]. Diese Anlagen zeichnen sich durch eine hohe Flexibilität aus, da sie üblicherweise thermische Pufferspeicher besitzen, mit denen die Gesteuerung vom Bedarf der Nutzenergie zeitlich entkoppelt wird, was prinzipiell einen gesteuerten (Teillast-) Betrieb möglich macht [Peh+06]. Zur Nutzung dieser Potenziale sind jedoch im Vergleich zu herkömmlichen Netzführungsstrategien völlig neuartige Koordinationsmechanismen erforderlich. Zentrale Aspekte sind hierbei die durch die Diversität in der Verortung der Anlagen entstehende *Dezentralität* des Anlagenparks, ihre vergleichsweise sehr hohe *Anzahl* sowie die Vielzahl an *Interessensgruppen* in Form von individuellen Anlagenbetreibern, denen jeweils in einem gemeinsamen Koordinationsansatz Rechnung getragen werden muss.

1.1 DEZENTRALE KONTROLLSTRUKTUREN IN ENERGIEVERSORGUNGSYSTEMEN

Die heutige Betriebsführung von Elektrizitätsnetzen ist eine komplexe Aufgabe, die eine Vielzahl sowohl organisatorischer als auch technischer Aspekte mitunter in Echtzeit berücksichtigen muss (für eine Einführung siehe bspw. [HDS10, Kap. 8]). Dies wird durch die eingangs genannten Entwicklungen stark verschärft, da zur Nutzbarmachung der Flexibilität von Kleinanlagen diese aktiv in den Betriebsführungsprozess integriert werden müssen. Die oben identifizierten Aspekte der Dezentralität, der immensen Anzahl der Anlagen sowie der heterogenen Interessen der Anlagenbetreiber erfordern dabei eine fundamentale Umgestaltung der klassischen Führungsstrategien. Hier ist auf den wegweisenden Beitrag von Wu, Moslehi und Bose aus dem Jahr 2005 [WMB05] zu verweisen, in welchem die Evolution der Betriebsführung von Elektrizitätsnetzen in detaillierter Weise aus verschiedenen Perspektiven betrachtet wird. Die Autoren kommen unter anderem zu folgendem Schluss:

Control centers today are in the transitional stage from the centralized architecture of yesterday to the distributed architecture of tomorrow. [...] To summarize, in a competitive environment, economic decisions are made by market participants individually and system-wide reliability is achieved through coordination among parties belonging to different companies, thus the paradigm has shifted from centralized to decentralized decision making. This requires data and application software in control centers to be *decentralized* and *distributed*.

Dies wird zwei Jahre später in einem ebenso grundlegenden Beitrag von Ilić aufgegriffen [Ili07] und vom Standpunkt komplexer dynamischer Systeme aus erweitert. Sie schreibt:

So, instead of having fully distributed, one would see gradually evolving, reconfiguring architectures, based on coalitions of actors with common goals. [...] Each component is characterized by its own internal states and interaction variables. While each component has its own primary subobjective, it is frequently grouped with several other components working toward a common subobjective within a larger system.

Dies ist in einer Linie mit der Feststellung des *Intelligent Systems* Unterkomitees der *IEEE Power Engineering Society* aus dem gleichen Jahr (siehe [McA+07]):

The current approach of using a central SCADA system and several smaller distributed SCADA systems is no longer sufficient for certain control operations. An approach that provides intelligent, fast, and adaptable local control and decision making is required.

Insgesamt zeichnen die Autoren somit ein Bild, in welchem die klassischen zentralisierten Kontrollstrukturen von einem dynamischen System bestehend aus einer Vielzahl unabhängiger Akteure mit *lokalen Entscheidungsprozessen* abgelöst werden, um der inhärent gesteigerten Komplexität des zukünftigen Energieversorgungssystems zu begegnen. Diese Vision ist seitdem an vielen Stellen in die Forschungslandschaft eingeflossen. So wird der Transitionsprozess zu einem von dezentralen Energiewandlungsanlagen geprägten System etwa von der Internationalen Energieagentur in den folgenden drei Schritten beschrieben (entnommen aus [NLH10]):

1. Technische Integration dezentraler Energiewandlungsanlagen in das bestehende Energieversorgungssystem ohne gesonderte organisatorische Anpassungen.
2. Organisatorische Integration der Anlagen in aggregierter Form zur zentral gesteuerten Bereitstellung von systemstabilisierenden Maßnahmen.
3. Individualisierte, autonome Teilnahme der Anlagen an dezentralen Handelsprozessen. Zentrale Kontrollinstanz hat nur noch vermittelnden Charakter.

Diese Sicht wird von der Deutschen Akademie der Technikwissenschaften geteilt und in ihrer Studie *Future Energy Grid* zum Ausdruck gebracht [AKM12, Abs. 3.4.12], in welcher verschiedene Migrationspfade des Energieversorgungssystems hin zu einem „Internet der Energie“ untersucht werden. Zahlreiche Forschungsarbeiten greifen diesen Grundgedanken auf und benennen weitere Argumente für die Dezentralisierung der Kontrollstrukturen. So vergleicht Nestle die Dezentralisierung mit der Transition von der Plan- zur Marktwirtschaft, wo sich gezeigt hätte, dass die theoretischen Vorteile der Planwirtschaft praktisch nicht nutzbar und erheblich weniger produktiv seien, wenn die Anzahl der beteiligten Akteure eine bestimmte Zahl überschreite [Nes07]. Das hierin angesprochene Problem der Skalierbarkeit kommt hinsichtlich der Berechenbarkeit auch in [Van06; BK10; Trö10; NLH10; Ram+12;

McN12; Pou13] zur Sprache. Ein weiteres technisches Hindernis wird in der Menge und Häufigkeit der an eine zentrale Instanz zu übermittelnden Informationen gesehen, welche für eine zentrale Kontrolle notwendig wären [NLH10; McN12; Pou13]. Die Robustheit spielt ebenfalls eine Rolle, da zentrale Systeme zwangsläufig betriebskritische Komponenten besitzen (sog. *Single Points of Failure*), deren Fehlverhalten die Funktionsweise des gesamten Systems beeinträchtigen würden. Diese Gefahr kann zwar durch redundante Auslegung der entsprechenden Komponenten verringert werden, was aber ökonomisch sowie organisatorisch aufwändig ist und daher kritisch gesehen wird [Trö10; Pou13]. Die mögliche Verletzung der Privatsphäre der nicht-autonomen Akteure (wie etwa Kunden eines Energieversorgers) in zentral kontrollierten Systemen wird ausführlich von [LMW10] und [AL10] diskutiert und fließt in [Peñ06; Nes07; NLH10; Pou13] als wichtiges Argument für dezentrale Kontrollstrukturen ein. Eng damit verwandt ist die Forderung nach der Aufrechterhaltung der Autonomie hinsichtlich Entscheidungsgewalt und Durchsetzung eigener Interessen von beteiligten Akteuren [NLH10; McN12]. Und schließlich wird die angestrebte Flexibilität selbst als Kriterium für dezentrale Koordinationsmechanismen identifiziert [Kad11; Ram+12].

Als Alternative zur zentralen Koordination unterscheidet [Trö10] prinzipiell zwei unterschiedliche Koordinationskonzepte: Auf der einen Seite steht hier die kollektivistische Koordination, wie sie etwa in sog. virtuellen Kraftwerken (VK) eingesetzt wird. Hierbei wird eine Menge von Anlagen hinsichtlich eines übergeordneten Ziels *aktiv gesteuert*. Dies erfolgt innerhalb eines VK zumeist wiederum auf zentrale Weise. Eine Übersicht entsprechender Forschungsprojekte und prototypischer Realisierungen ist aktuell etwa in [Ste13] zu finden. Aber auch Ansätze zur dezentralen Betriebsführung von VK werden diskutiert [BK10], wobei hier ebenfalls die aktive Steuerung der Anlagen im Vordergrund steht. Dem gegenüber steht die individualistische Koordination, in welcher die Akteure durch *Anreize*, etwa monetärer Art, indirekt zu einer dem globalen Ziel entgegenkommenden Handlung bewegt werden sollen. Dieses Konzept zeichnet sich also durch eine hochgradige Autonomie der partizipierenden Akteure aus und wird zumeist in Form von Marktmechanismen umgesetzt (eine Übersicht hierzu liefert [Ram+11]). Gleichwohl sieht [Trö10] diese Konzepte nicht als überschneidungsfrei an, und schlägt in seiner Arbeit ein dezentrales Führungskonzept für VK mit Fokus auf die reaktive Einsatzplanung vor, in welchem die Autonomie der Akteure zu einem gewissen Grad erhalten bleibt.

Zusammenfassend kann also festgehalten werden, dass das Konzept der dezentralen Kontrollstrukturen für Energieversorgungssysteme derzeit aktiv diskutiert und an vielen Stellen als erforderliche Maßnahme für eine erfolgreiche Realisierung des Smart Grid gesehen wird. Dabei stellt sich jedoch nicht die Frage, welcher konkrete Koordinationsansatz für das Gesamtsystem umzusetzen ist. Vielmehr wird es zu einem Nebeneinander verschiedener Kontrollstrukturen in Ergänzung der bestehenden Betriebsführungsprozesse kommen [Bec+13], und die Aufgabe wird es sein, diese zu einem stimmigen Gesamtbild zu integrieren.

1.2 DEZENTRAL KOORDINIERTES WIRKLEISTUNGSBEREITSTELLUNG

Einen ganzheitlichen Integrationsansatz für dezentrale Kontrollstrukturen im Kontext der einleitend geschilderten Problematik verfolgt aktuell der interdisziplinäre, vom Niedersächsischen Ministerium für Wissenschaft und Kultur (MWK) geförderte Forschungsverbund „Smart Nord – Intelligente Netze Norddeutschland“.¹ Ausgehend von neu entwickelten Konzepten zur dezentralen Bereitstellung von Wirkleistung, Regelleistung und Blindleistung in den Verteilnetzen auf Basis einer gemeinsamen informations- und kommunikationstechnischen Infrastruktur werden die resultierenden Effekte auf die physikalische Versorgungsstruktur, die ökonomische Ebene sowie ökologische, soziale und sicherheitstechnische Aspekte untersucht. Einen Überblick zu diesen verschiedenen Betrachtungswinkeln, welche eng verzahnt in unterschiedlichen Teilprojekten bearbeitet werden, zeigt [Abbildung 1.1](#). Einen zentralen Platz nimmt hierbei die dynamische Aggregation von Akteuren des Verteilnetzes zur fahrplanbasierten Bereitstellung von Wirkleistung ein, was Gegenstand von Teilprojekt 1 des Forschungsverbundes ist und im Folgenden näher betrachtet werden soll. Hierzu sei bemerkt, dass für die Bereitstellung von Systemdienstleistungen ein ähnlicher Ansatz verfolgt wird; weitere Informationen dazu liefert etwa [Nie+12].

Bei der fahrplanbasierten Wirkleistungsbereitstellung handelt es sich um die prädiktive Zuweisung von Fahrplänen an Energieanlagen für einen in der Zukunft gelegenen Erbringungszeitraum. Das Ziel ist es hierbei, den Betrieb einer oder mehrerer

¹<http://www.smartnord.de>, letzter Zugriff 6. Januar 2014

²<http://www.offis.de>, letzter Zugriff 6. Januar 2014

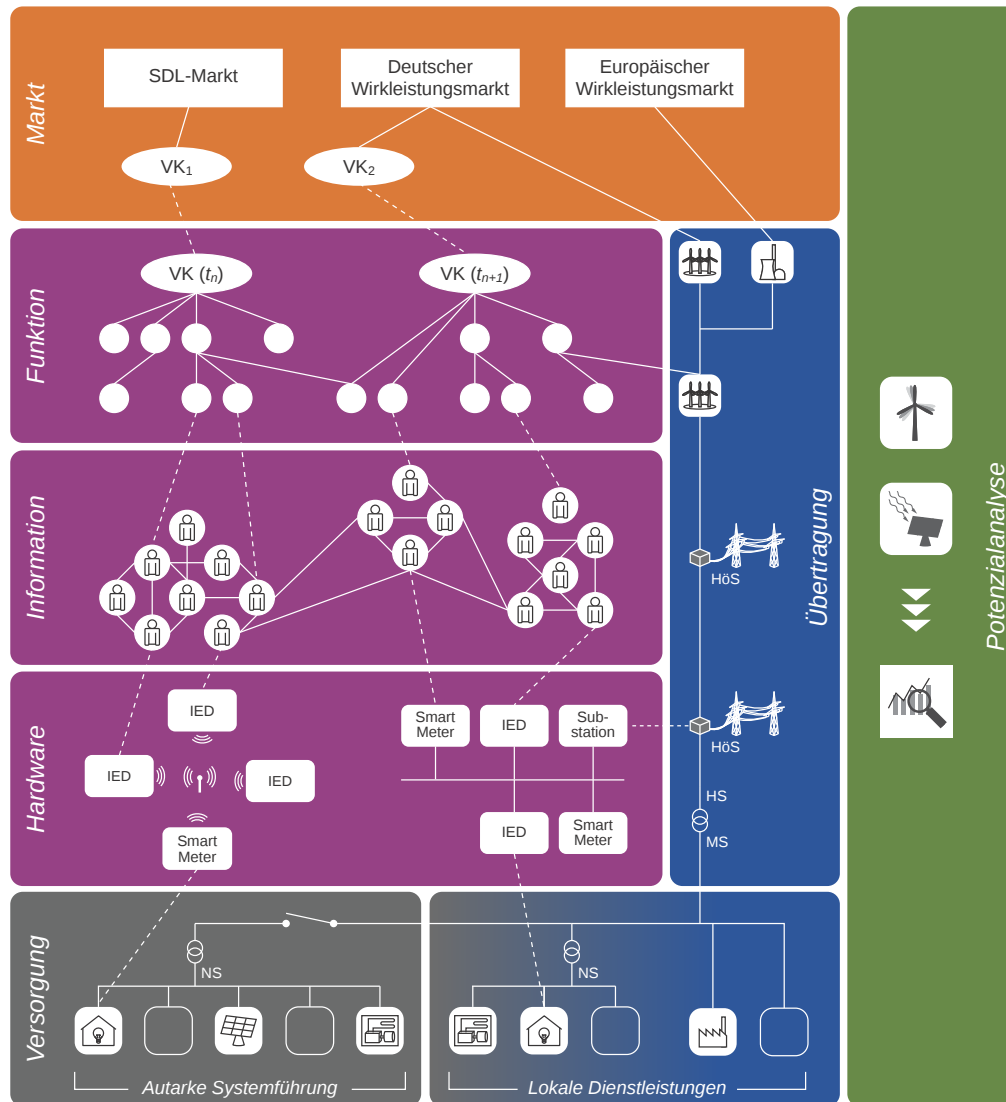


Abbildung 1.1: Gesamtübersicht der betrachteten Topologieebenen im Forschungsverbund Smart Nord. Autor der Darstellung: OFFIS – Institut für Informatik², mit Genehmigung verwendet.

Anlagen hinsichtlich eines spezifischen Erbringungszeitraumes so einzuplanen, dass die von ihnen dann erbrachte Wirkleistung (es kann sich dabei sowohl um positive als auch um negative Wirkleistung handeln) einem gewünschten zeitlichen Verlauf entspricht. Dieser angestrebte Verlauf wird auch als *Wirkleistungsprodukt* bezeichnet. Ein typischer Anwendungsfall für diese Art der *prädiktiven Einsatzplanung* ist die Vortagsplanung, bei welcher im Rahmen des Handels von Wirkleistung die Handelsmenge am Vortag bzgl. des vereinbarten Erbringungszeitpunktes vertraglich festgelegt wird [Ste13] (dieses Vorgehen kommt bspw. am sog. *Day-Ahead*-Markt von Stromhandelsbörsen zum Einsatz und wird später in [Abschnitt 7.6.4](#) genauer betrachtet). Mit der eingangs beschriebenen Integrationsproblematik einer großen Menge von dezentralen Energiewandlungsanlagen stellt sich hier die Frage nach einer effizienten Koordinationsstrategie, welche die Flexibilität der Anlagen in Form einer fahrplanbasierten Wirkleistungsbereitstellung systemstabilisierend nutzen kann und dabei aber die im vorigen Abschnitt identifizierten Herausforderungen bzgl. Berechenbarkeit, Kommunikationsaufwand, Datenschutz und Autonomieerhalt berücksichtigt. Zudem muss beachtet werden, dass für dezentrale Energiesysteme mit hohem Anteil erneuerbarer Energien aufgrund von Prognoseunsicherheiten eine Planung nur auf Zeitskalen von einigen Stunden bis hin zu wenigen Tagen möglich ist. Langfristige Planungsverfahren, welche alle relevanten Daten an zentraler Stelle sammeln und eine globale Optimierung unter Berücksichtigung aller Parameter durchführen, wie es etwa in der klassischen Kraftwerkseinsatzplanung der Fall ist (engl. *Unit Commitment*, siehe z. B. [Cra04; Pad04]), können hier nicht sinnvoll angewendet werden. Ebenso ist eine direkte Teilnahme am Wirkleistungshandel durch die einzelnen Anlagenbetreiber nicht sinnvoll durchführbar, da der Betrieb der Anlagen zu großen Unsicherheiten unterliegt, was eine stabile Einsatzplanung verhindert.

Im Forschungsverbund Smart Nord wird daher das Konzept der dynamischen Verbundbildung von dezentralen Akteuren verfolgt. Dieser Ansatz kann als Anpassung der Koordinationsstrategie der virtuellen Kraftwerke an die Anforderungen einer dezentralen Energieversorgung verstanden werden: Während virtuelle Kraftwerke eine mittel- bis langfristig (d. h. mehrere Wochen bis Jahre) stabile Zusammensetzung aufweisen und einen eindeutig identifizierbaren Betreiber haben, der die Flexibilität des VK in seinem persönlichen Interesse einsetzt, werden Verbünde im Forschungsverbund Smart Nord hier nur für den Zeitraum eines spezifischen kurzfristigen Ziels (bspw. einer tagesaktuellen Erbringung von Wirkleistung an einer Strombörse) ge-

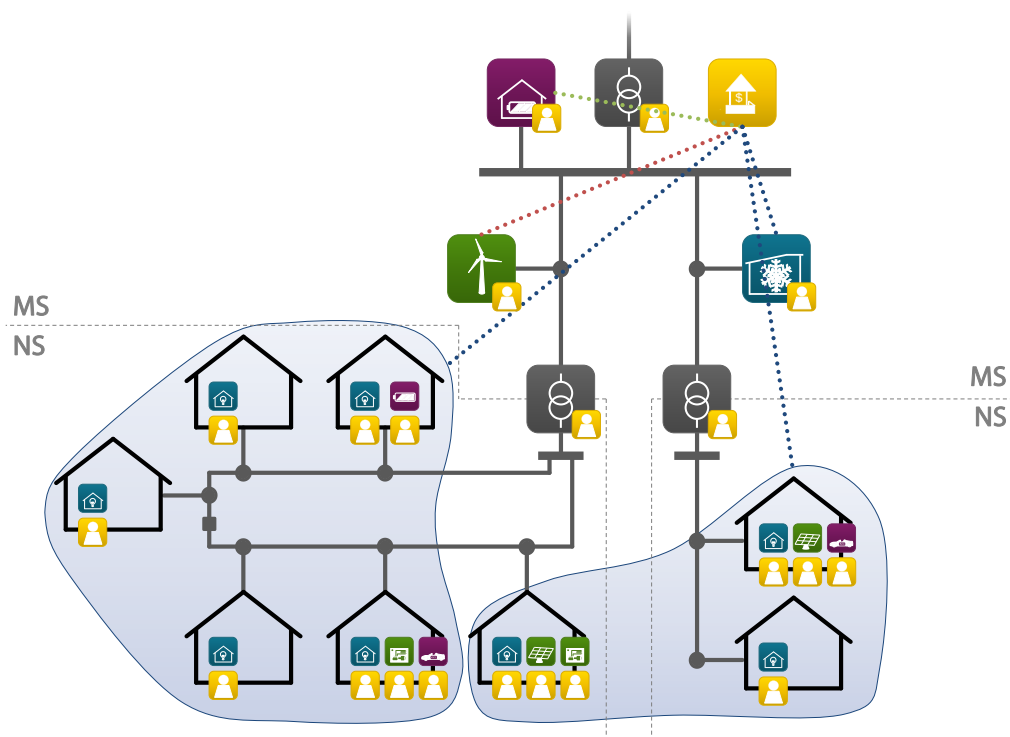


Abbildung 1.2: Dezentral koordinierte Wirkleistungsbereitstellung, Teilprojekt 1 im Forschungsverbund Smart Nord. Autor der Darstellung: OFFIS – Institut für Informatik², mit Genehmigung verwendet.

schlossen und bestehen somit aus einer Gemeinschaft von individuellen Akteuren, die für diesen Zeitraum ein gemeinsames Ziel verfolgen (siehe [Abbildung 1.2](#)). Der Vorteil besteht in diesem Ansatz darin, dass die langfristige Autonomie der partizipierenden Anlagenbetreiber gewährleistet ist und dennoch Planungssicherheit für die beteiligten Akteure besteht, da durch Aggregation mehrerer Anlagen Potenzial zur reaktiven Kompensation von Prognoseabweichungen innerhalb des Verbundes nutzbar gemacht wird, sodass vertragliche Bindungen an einem übergeordneten Handelsplatz eingehalten werden können. Es handelt sich somit um eine Hybridisierung der kollektivistischen und der individualistischen Koordination, mit welcher die Vorteile beider Konzepte verbunden, und ihre Nachteile vermieden werden sollen.

Um hiermit erfolgreich den oben identifizierten Herausforderungen bzgl. der In-

tegration einer großen Menge von dezentralen Energiewandlungsanlagen begegnen zu können, werden im beschriebenen Teilprojekt 1 des Forschungsverbundes Smart Nord (unter anderem) effiziente Verfahren zur Bildung von Verbünden, zur verbundinternen Optimierung von Anlagenfahrplänen sowie zur Reaktion auf unvorhergesehene Ereignisse hinsichtlich der Kompensation von Prognoseabweichungen während der Erbringung eines vereinbarten Wirkleistungsproduktes erforscht. So wurde bspw. zum prädiktiven Bilden eines Einsatzplans für die Anlagen eines bestehenden Verbundes zur Erbringung eines gemeinsamen Wirkleistungsproduktes eine Suchraumstruktur entwickelt, die es Optimierungsverfahren erlaubt, die Flexibilität der beteiligten Anlagen unabhängig von anlagenspezifischen Nebenbedingungen zu durchsuchen und im Rahmen einer kombinatorischen Optimierung daraus einen Einsatzplan zu bilden. Damit wird es möglich, verschiedene Optimierungsverfahren auf dieses kombinatorische Problem anzuwenden (eine genauere Darstellung und Definition des Optimierungsproblems wird später in dieser Arbeit gegeben), und somit losgelöst von diesem Anwendungskontext zu entwickeln. In [BS12] wird hierzu eine prototypische Heuristik vorgestellt, welche auf sequenzieller iterativer Anpassung der Fahrpläne der einzelnen Anlagen basiert und somit ein semi-dezentrales Verfahren unter Verwendung globalen Wissens darstellt. Hier besteht somit noch Forschungsbedarf hinsichtlich vollständig dezentraler Verfahren, um die oben genannten Anforderungen zu erfüllen. Dies wird im folgenden Abschnitt durch die Auswertung existierender Arbeiten zu diesem Thema untermauert.

1.3 VERWANDTE ARBEITEN

In bisherigen Arbeiten wurden für vergleichbare Fragestellungen vorwiegend markt-basierte Ansätze vorgeschlagen. Diese basieren jedoch entweder auf kombinatorischen Auktionen, für welche ein zentraler Preisbildungsprozess mit erheblichem Rechenaufwand (das Problem ist NP-vollständig, siehe [Peñ06]) unter Verwendung globaler Informationen durchgeführt werden muss, oder sie verwenden Bietkurven (= im Wertebereich beschränkte Funktionen, die Handelsmengen auf Preise abbilden) zur globalen Preisbildung, womit jedoch Abhängigkeiten zwischen aufeinanderfolgenden Handelsperioden kaum umsetzbar sind und welche daher primär im Rahmen von *Supply-Demand-Matching* Ansätzen eingesetzt werden [KWK05; LPJ10].

Die wenigen Arbeiten, die sich mit Verfahren zur dezentralen Einsatzplanung von Energieanlagen beschäftigen, werden im Folgenden in chronologischer Reihenfolge kurz diskutiert (eine Übersicht von verwandten Ansätzen aus anwendungsferner, algorithmischer Sicht wird in den Abschnitten 2.1.1 – 2.1.4 gegeben).

1.3.1 COBB

In [Peñ06] entwickelt Peña den Algorithmus *Constraint Optimization by Broadcasting* (COBB) zur optimalen prädiktiven Einsatzplanung flexibler Verbraucher. Hierbei handelt es sich um ein synchrones Verfahren, welches eine rekursive Enumeration des gesamten Lösungsraumes durchführt, und in jedem Rekursionsschritt eine iterative, semi-dezentrale Heuristik zur Generierung von vielversprechenden Lösungen anwendet: Ein ausgewählter Akteur dient als Koordinator des Verfahrens und fordert bei allen anderen beteiligten Akteuren eine Verbesserung des aktuellen (zu Beginn des initialen) Einsatzplanes an. Jeder dieser angefragten Akteure modifiziert den erhaltenen Einsatzplan durch Auswahl des dazu best passenden eigenen Fahrplans und sendet diesen zurück. Der Koordinator wählt nun aus allen Antworten die beste aus und schreibt diese wiederum zur Modifikation aus. Alle anderen Antworten werden zur späteren Bearbeitung gespeichert. Dies wird nun solange wiederholt, bis keine Verbesserung der aktuellen Lösung möglich ist (es wurde somit ein lokales Optimum im Lösungsraum gefunden). Diese Lösung wird als Zwischenergebnis separat abgelegt. Aus den zuvor gespeicherten Antworten wird nun derjenige Einsatzplan mit höchster Güte ausgewählt und das Verfahren rekursiv mit diesem neu gestartet.

Das Verfahren weist einige Schwachstellen auf. Zunächst handelt es sich um ein synchrones Verfahren – in jedem Iterationsschritt der inneren Heuristik finden parallele Modifikationen einer Lösung statt, von denen dann die beste zur unmittelbaren Weiterbearbeitung ausgewählt wird. Es muss somit erst auf die Antworten aller dezentralen Akteure gewartet werden, bis das Verfahren fortgeführt werden kann. Mit dem Koordinator existiert ein zentraler Punkt im System, bei dem alle Informationen zusammenlaufen, das Verfahren ist somit nur teilweise dezentral. Der Kommunikationsaufwand ist hier äußerst kritisch zu sehen. Zudem werden alle Zwischenlösungen zur späteren rekursiven Wiederaufnahme gespeichert, was unter dem Aspekt der kombinatorischen Problemkomplexität hinsichtlich der Größe des Lösungsraumes mindestens als schwierig anzusehen ist.

1.3.2 Stigspace

Eine Einsatzplanung auf Basis iterativer Anpassungen ähnlich zu dem COBB Algorithmus (siehe oben) wird von Li et al. vorgeschlagen [LPJ08; LPJ10]. Statt eines zentralen Koordinators wird hier eine zentrale Wissensbasis verwendet (von den Autoren als *Stigspace* bezeichnet), auf welche alle Akteure kommunikationstechnischen Zugriff besitzen. Die Akteure sind dazu verpflichtet, ihre jeweiligen Fahrpläne für den betrachteten Planungshorizont in diesem *Stigspace* zu veröffentlichen, sodass sie von allen beteiligten Akteuren einsehbar sind. Das globale Optimierungsziel wird von übergeordneter Stelle ebenfalls im *Stigspace* platziert und darüber von den dezentralen Akteuren wahrgenommen. Jeder Akteur passt nun unter Berücksichtigung der derzeitigen Situation (d. h. der veröffentlichten Fahrpläne der anderen Akteure) seinen eigenen Fahrplan hinsichtlich der Erfüllung des globalen Ziels und unter Beachtung lokaler privater Nebenbedingungen möglichst zielführend an, und platziert den modifizierten Fahrplan wiederum im *Stigspace*. In welcher Form der jeweils eigene Fahrplan modifiziert wird, ermittelt ein auf zufällig ausgewählten Lastverschiebungen basierender Algorithmus für die einzelnen Akteure individuell. Dieses Vorgehen wird nun iterativ solange wiederholt, bis keine Änderungen mehr auftreten und das System sich somit stabilisiert, oder bis ein übergeordnetes Abbruchkriterium in Kraft tritt.

Das beschriebene Verfahren ist speziell auf die punktuelle Spitzenlastreduktion unter Verwendung von zweipunktgeregelten thermischen Geräten ausgerichtet, kann aber mit geringen Anpassungen auch zur Planung eines Wirkleistungsproduktes verwendet werden. Problematisch ist hier die Zentralität des *Stigspace*, wodurch der Ansatz nicht ohne Einschränkungen als dezentral gesehen werden kann, und sich ähnliche Nachteile wie im COBB Ansatz oben ergeben. Durch die unabhängig voneinander stattfindenden individuellen Fahrplanmodifikationen besteht zudem prinzipiell die Gefahr, dass das System zwischen mehreren Lösungen oszilliert. Selbst unter Vernachlässigung dieses Effektes ist unklar, ob das Verfahren sinnvoll konvergieren kann und bei ausreichend guten Lösungen terminiert. Trotz dieser Unklarheiten scheint der grundlegende Koordinationsansatz der individuellen, iterativen Anpassung vielversprechend zu sein und wird daher im Laufe der Arbeit wiederholt als Vergleichsmaßstab aufgegriffen werden.

1.3.3 Restriktionslose Gerätepools

Eine Mischform aus autonomen Akteuren und dem Konzept des Virtuellen Kraftwerks wird in der Dissertation von Kamper betrachtet [Kam10]. Der vorgestellte Ansatz versucht, Gruppierungen von flexiblen Verbrauchern sowie dezentralen Erzeugern zu bilden, welche durch dynamische Einsatzplanungen einen Lastausgleich innerhalb der Gruppe vornehmen können und gleichzeitig Freiheitsgrade zur zentral gesteuerten Einflussnahme auf das Lastverhalten der Gruppe nach außen hin anbieten. Die Gruppeneinteilung erfolgt statisch anhand der organisatorischen Bilanzkreise des Netzbetreibers. Die Kommunikation der Akteure innerhalb jeder Gruppe wird durch ein *Peer-to-Peer* Netz (P2P) mit zufälliger Vernetzung realisiert, sodass jeder Akteur etwa 10 bis 100 Nachbarn besitzt, mit denen er kommunizieren kann. Der angestrebte Lastausgleich (*Supply-Demand-Matching*) innerhalb der Gruppe wird nun von den Verbrauchern ausgehend durchgeführt: Erkennt ein Akteur, dass „in der nächsten Zeit Strom verbraucht werden muss“ [Kam10, S. 104], wird eine Anfrage an eine Teilmenge seiner Nachbarn (die Auswahl ist nicht näher spezifiziert) gesendet. Diese antworten mit den Freiheitsgraden, die sie im betreffenden Zeitpunkt besitzen, aus denen der anfragende Akteur nun kombinatorisch mittels einer Verknüpfung einer auf den Anwendungsfall spezialisierten lokalen Suche mit einem evolutionären Algorithmus die optimale Deckung seines prognostizierten Verbrauches ermittelt, und die dafür erforderlichen Aktionen der Nachbarn diesen zurückmeldet.

Als Ergänzung hierzu wird die Option einer prädiktiven Einsatzplanung beschrieben, welche sich durch Modellierung eines virtuellen Akteurs realisieren lässt. Dieser tritt einer einzuplanenden Gruppe bei und schreibt den gewünschten Wirkleistungsverlauf wie oben beschrieben bei seinen Nachbarn aus. Mit den zurückgemeldeten Freiheitsgraden kann dann eine zentrale Einsatzplanung durchgeführt werden. Hierbei verbleibende Diskrepanzen zwischen gewünschtem Wirkleistungsverlauf und resultierendem Einsatzplan werden durch einen weiteren zentralen Planungsschritt im Nachgang behoben. Hierzu wird eine sog. elitäre Gruppe aus einigen wenigen Akteuren gebildet, welche jederzeit zentrales Wissen über verbleibende Leistungsdivergenzen in der Gruppe besitzen und versuchen, diese durch iterative Anpassungen der eigene Leistung zu kompensieren.

Auch dieser Ansatz weist einige Schwächen auf, die der Autor in seiner Arbeit bereits selbst erkannt hat. So können etwa Konflikte in der Verwertung von Freiheits-

graden einzelner Akteure auftreten, wenn diese gleichzeitig von verschiedenen anfragenden Akteuren benötigt werden. In diesem Fall muss der Vorgang abgebrochen werden [Kam10, S. 113]. Weiterhin stellt die individuelle Berechnung von optimalen Abdeckungen eines Akteurs kein effizientes Mittel dar, um der Komplexität des Gesamtproblems zu begegnen, da hierdurch eine Vielzahl kombinatorischer Optimierungsprobleme jeweils lokal bearbeitet werden muss, was rechnerisch sehr aufwändig ist [Kam10, S. 119] und insbesondere in Anbetracht der Anzahl von „ungefähr 90 000“ [Kam10, S. 123] in einer Tageseinsatzplanung durchzuführenden Optimierungsvorgänge auch durch die nachträglich ergänzte, sehr spezialisierte lokale Suche nicht zufrieden stellend gelöst werden kann. Und schließlich zeigt die Evaluation des Ansatzes der elitären Gruppe, dass die hiermit durchgeführte nachgelagerte Optimierung zu einem Aufschaukeln von Kompensationsleistungen der Akteure führt, sodass im Endeffekt eine Teilmenge der Akteure mehr Strom verbraucht als sie müsste, und eine andere entsprechend mehr produziert als sie müsste [Kam10, S. 138 ff]. Diese Punkte zeigen, dass der vorgestellte Ansatz insgesamt wenig Potenzial besitzt und so nicht sinnvoll realisierbar ist.

1.3.4 Holonische virtuelle Kraftwerke

In seiner fast gleichzeitig zu Kamper (siehe oben) angefertigten Dissertation beschreibt Tröschel ein koordinationsstechnisch ähnliches Konzept, jedoch mit Fokus auf die reaktive Einsatzplanung, d. h. die Anpassung von Fahrplänen bei Prognoseabweichungen und verwandten Ereignissen [Trö10]. Hier werden auszugleichende Leistungsabweichungen innerhalb eines virtuellen Kraftwerks (VK) in Form von Aufträgen ausgeschrieben, welche an zentraler Stelle verwaltet und durch Verhandlungen mit den verbleibenden Akteuren des VK nach Möglichkeit erfüllt werden. Zur Reduktion des Koordinations- und Kommunikationsaufwandes führt Tröschel hier eine artifizielle Unterteilung des VK in sog. Holone (kooperative Untergruppen von Akteuren, die nach außen als Einheit auftreten) durch, und schafft somit eine Hierarchie von Akteuren, die es ermöglicht, Aufträge möglichst lokal (d. h. innerhalb einer Untergruppe) zu erfüllen. Die Besonderheit liegt hier in der dynamischen Adaption der Holonstruktur, sodass die resultierende Hierarchie zur Laufzeit stets hinsichtlich der schnellen Bearbeitung von Aufträgen optimiert wird.

Dieses Konzept stellt offensichtlich eine sinnvolle Überlegung für die reaktive

Einsatzplanung dar, da Ereignisse hier lokal kompensiert werden können, und ist explizit für diesen Anwendungsfall entwickelt worden. Tröschel weist in seiner Arbeit darauf hin, dass die Einsatzfähigkeit des Ansatzes für die prädiktive Einsatzplanung nicht untersucht wird. Da bei der prädiktiven Einsatzplanung eine Optimierung über alle Akteure erfolgen muss (d. h. eine lokale Separierbarkeit ist nicht gegeben), würde eine Einteilung in Holone keine Vorteile bringen.

1.3.5 AVPP

Im gleichen Jahr wie Kamper und Tröschel (siehe oben) schlagen Anders et al. das Konzept der *Autonomous Virtual Power Plants* (AVPP) vor [And+10]. Die zugrundeliegende Idee gleicht den obigen Konzepten: Es werden ebenfalls Gruppen von Energieanlagen gebildet, die einerseits intern ein *Supply-Demand-Matching* durchführen, und nach außen hin Regelpotenzial zur Verfügung stellen können. Ähnlich zu Tröschel wird eine adaptive Reorganisation der internen Gruppenstruktur in Untergruppen vorgesehen, um Teilplanungen lokal durchführen zu können und damit die Berechnungskomplexität zu reduzieren. Der dazugehörige Algorithmus ist in [Ste+13a] veröffentlicht. Eine wesentliche Neuerung im Vergleich zu den Ansätzen von Kamper und Tröschel besteht in der expliziten Modellierung von eigennützigen, mitunter nicht vertrauenswürdigen Akteuren. Hierzu wird ergänzend in [And+13a] ein Anreizmechanismus entwickelt, um auch unter entsprechend unsicheren Bedingungen eine möglichst gute Planung durchführen zu können. Zudem stellt das Verfahren eine Kombination aus prädiktiver und reaktiver Einsatzplanung dar, und kann damit als konzeptuelle Erweiterung der holonischen VK von Tröschel gesehen werden.

Die in den AVPP stattfindenden Einsatzplanungen werden im vorgestellten Verfahren mittels eines iterativen Marktalgorithmus durchgeführt. Dabei wird die Differenz zwischen aktuellem und gewünschtem Wirkleistungsverlauf innerhalb eines AVPP zentral in Form einer Auktion ausgeschrieben. Von den Angeboten der beteiligten Akteure zur Deckung dieser Differenz wird das beste ausgewählt, bevor in der nächsten Iteration mit einer weiteren Auktion bzgl. der nun noch verbleibenden Differenz fortgefahren wird. Dieses Koordinationskonzept entspricht im Wesentlichen dem iterativen Teil des weiter oben vorgestellten COBB Algorithmus (Abschnitt 1.3.1) und besitzt daher die gleichen Probleme aus Sicht der Dezentralität (abzüglich des

Speicheraufwandes, welcher bei COBB aus der *Vollständigkeit* des Algorithmus resultiert).

1.3.6 EPOS

Ein weiteres auf Hierarchisierung basierendes Konzept beschreibt Pournaras in seiner Dissertation [Pou13]. In dem Verfahren *Energy Plan Overlay Self-stabilization* (EPOS) werden Akteure in einer Baumstruktur organisiert, welche sich zur Laufzeit bzgl. benutzerdefinierten Kriterien selbst anpassen kann. Der Autor führt hier exemplarisch die Leistungsklasse einer Energieanlage sowie die Volatilität ihrer Wirkleistungsverläufe als mögliche Kriterien an, geht im Rahmen seiner Evaluation allerdings nicht näher auf den Aspekt der adaptiven Restrukturierung ein. Innerhalb dieser Baumstruktur wird dann eine *bottom-up* Planbildung durchgeführt, in welcher die Akteure, bei den Blättern des Baumes startend, ihre möglichen Fahrpläne an die übergeordnete Ebene senden. Hat ein übergeordneter Akteur alle möglichen Fahrpläne seiner ihm untergeordneten Akteure erhalten, ermittelt er durch Anwendung der Exhaustionsmethode (*brute-force search*) die beste Kombination dieser Pläne, und meldet die jeweils ausgewählten Fahrpläne an die ihm untergeordneten Akteure zurück. Anschließend sendet er wiederum diese ausgewählte Kombination samt all seiner eigenen möglichen Fahrpläne an den ihm übergeordneten Akteur, damit dort eine weitere Aggregation auf höherer Ebene stattfinden kann. Somit finden in den unteren Ebenen des Baumes diverse parallele Kombinationsbildungen von Plänen statt, welche schließlich an der Wurzel des Baumes zu einem Gesamtplan aggregiert werden. Da die in den verschiedenen Zweigen ermittelten Teilpläne nicht notwendigerweise kompatibel (hinsichtlich der zu optimierenden Zielfunktion) sein müssen, wird dieses Verfahren insgesamt iterativ wiederholt. Hierzu sendet der Akteur an der Wurzel mittels *broadcasting* den aktuellen Gesamtplan an alle Akteure der Baumstruktur, um eine erneute *bottom-up* Planbildung zur Verbesserung dieses Plans anzustoßen.

Der vorgestellte Ansatz verbindet (zumindest konzeptuell) das Prinzip der in den obigen Verfahren bereits vorgestellten adaptiven Anpassung der Organisationsstruktur mit einer ebenfalls aus den obigen Verfahren bekannten lokalen Planbildung, um die Berechnungskomplexität des Gesamtproblems zu reduzieren. Zusätzlich zu den oben bereits dokumentierten Schwierigkeiten dieses Koordinationsansatzes besteht

bei EPOS das Problem, dass Akteure ihre gesamten Suchräume an ihre übergeordneten Akteure in der Baumstruktur senden, was signifikante Auswirkungen auf den Kommunikationsaufwand, die Aspekte des Datenschutzes und Autonomie sowie den Berechnungsaufwand (in den einzelnen Akteuren) hat.

Als Ergänzung zu EPOS schlägt Pournaras das Verfahren *Adaptive Load Management by Aggregation* (ALMA) vor. Mit diesem soll es möglich sein, ein globales reaktives *Supply-Demand-Matching* durchzuführen, indem etwa bei signifikanten Lastspitzen der aktuelle Netzzustand mittels *gossiping* (stochastische Informationsausbreitung von Akteur zu Akteur) den beteiligten Akteuren bekannt gemacht wird, damit diese dann individuelle Ausgleichsmaßnahmen durchführen können. Dies entspricht jedoch keiner prädiktiven Planbildung und wird daher in der vorliegenden Arbeit nicht weiter betrachtet.

1.3.7 Stand der Technik (Zusammenfassung)

Seit den Beiträgen von Wu, Moslehi und Bose (2005) und von Ilić (2007) zum Thema dezentrale Kontrollstrukturen in Energieversorgungssystemen ([Abschnitt 1.1](#)) sind einige Arbeiten entstanden, in denen die direkte dezentrale Betriebsführung von dezentralen Energiewandlungsanlagen im Rahmen von Einsatzplanungsprozessen behandelt wird. Die ersten Arbeiten im Zeitraum 2006 – 2010 (COBB und *Stigspace*) versuchten, mittels der Verteilung der Entscheidungsprozesse auf alle Einzelanlagen und nachfolgender zentraler Zusammenführung der Ergebnisse eine Dezentralisierung der Koordination herbeizuführen. Beide wiesen – neben weiteren konzeptuellen Schwachstellen – jedoch noch essenzielle zentrale Komponenten auf. Um 2010 wurde das Konzept der Hierarchisierung intensiv untersucht, um zentrale Komponenten (mit globalem Wissen) zu vermeiden, und den Berechnungsaufwand weiter zu verteilen. Dies geschah mit unterschiedlichen Zielsetzungen im Detail, und mit unterschiedlichem Erfolg. Keiner dieser Ansätze ist jedoch zufriedenstellend für eine prädiktive, fahrplanbasierte Einsatzplanung dezentraler Akteure geeignet: Die restriktionslosen Gerätepools können wie im entsprechenden Abschnitt dargelegt aus verschiedenen Gründen konzeptuell nicht überzeugen, die holonischen VK sind nur für reaktive Maßnahmen sinnvoll anwendbar, die AVPP zeigen ihre Stärken in der Kompensation von Planungsunsicherheiten, führen jedoch eine nur semi-dezentrale Planbildung durch, was auch bei EPOS der Fall ist, wo zusätzlich Probleme bzgl. Kom-

munikationsaufwand, Datenschutz, Autonomie sowie Berechnungsaufwand identifiziert wurden.

Wie bereits am Ende von [Abschnitt 1.2](#) angedeutet, besteht also noch Forschungsbedarf hinsichtlich vollständig dezentraler Verfahren zur prädiktiven fahrplanbasierter Einsatzplanung, um die in [Abschnitt 1.1](#) identifizierten Anforderungen zu erfüllen.

1.4 ZIELSETZUNG DER ARBEIT

Im bisherigen Verlauf der Arbeit wurden vier aufeinander aufbauende Bereiche diskutiert: Zunächst wurde der aktuell stattfindende Transformationsprozess des Energieversorgungssystems beschrieben und hieraus ein Handlungsbedarf bzgl. der Integration dezentraler Energiewandlungsanlagen in Koordinationsstrategien zur *Nutzbarmachung ihrer Flexibilität* abgeleitet. Nachfolgend wurde dieser Handlungsbedarf auf Basis spezifischer Herausforderungen auf die Erforschung *dezentraler Kontrollstrukturen* fokussiert. Im Anschluss wurde dann hierzu ein exemplarischer Integrationsansatz anhand der Arbeiten des Forschungsverbundes Smart Nord vorgestellt und schließlich im Zusammenhang mit der Betrachtung verwandter Arbeiten Forschungsbedarf im Bereich der *prädiktiven Einsatzplanung* identifiziert. Aus diesen Punkten ergibt sich nun der Forschungsschwerpunkt der vorliegenden Arbeit:

Gesucht ist eine dezentrale Koordinationsstrategie für die Nutzbarmachung der Flexibilität dezentraler Energiewandlungsanlagen mittels fahrplanbasierter Wirkleistungsbereitstellung im Rahmen einer prädiktiven Einsatzplanung. Hierbei soll eine Dezentralisierung sowohl der Information als auch der Planung und Steuerung der betrachteten Anlagen erfolgen. Dabei sollen auf der einen Seite die Interessen der jeweiligen Anlagenbetreiber in Form von Autonomie und Datenschutz gewahrt bleiben. Auf der anderen Seite soll aber eine Planungssicherheit sowohl aus Sicht der Anlagenbetreiber, als auch aus Sicht der Netzbetreiber gewährleistet sein. Weiterhin soll der Ansatz eine hohe Effizienz bzgl. Berechenbarkeit, Zeitaufwand und Kommunikationsaufwand aufweisen. Hierzu ist es erforderlich, erzwungene Synchronität in der Koordinationsstrategie zu vermeiden, und stattdessen eine asynchrone Strategie

zu verfolgen. Gleichmaßen ist die Robustheit des Verfahrens in Bezug zur inhärenten Dynamik des Gesamtsystems zu betrachten, sodass starre Kommunikationsstrukturen zu vermeiden sind. Insgesamt soll also eine Heuristik gefunden werden, welche in möglichst kurzer Zeit möglichst gute Lösungen liefert und dabei die obigen Eigenschaften erfüllt.

Als motivierendem Anwendungsfall wird begleitend die in [Abschnitt 1.2](#) beschriebene dynamische Verbundbildung hinsichtlich des Handels von Wirkleistungsprodukten an einer Strombörse herangezogen. Wie später noch gezeigt werden wird, ist das in dieser Arbeit entwickelte Verfahren aber nicht auf diesen Anwendungsfall beschränkt.

1.5 GANG DER UNTERSUCHUNG

Die Arbeit beginnt im folgenden Kapitel mit einer kurzen Darstellung der Grundlagen, die für die Bearbeitung der Forschungsfrage wesentlich sind. Hier wird unter anderem auf die Konzepte der verteilten Optimierung sowie der Multiagentensysteme eingegangen. Anschließend wird das betrachtete Problem in [Kapitel 3](#) detailliert formalisiert und auf dieser Basis ein neuartiges Verfahren zur Lösung des gestellten Optimierungsproblems entwickelt. Dabei wird besonderen Wert auf eine strukturierte Herangehensweise im Rahmen eines etablierten Designprozesses gelegt, sodass die Entwicklung des Verfahrens in aufeinander aufbauenden Schritten dokumentiert wird. [Kapitel 4](#) stellt das resultierende Verfahren schließlich zusammenfassend in kompakter Form dar und bettet es in einen Ausführungskontext ein.

Im Anschluss daran beginnt der zweite Teil der Arbeit, in welchem das entwickelte Verfahren hinsichtlich der oben aufgestellten Kriterien evaluiert und kritisch untersucht wird. Hierzu wird in [Kapitel 5](#) eine formale Betrachtung der Konvergenzeigenschaften des Verfahrens vorgenommen. [Kapitel 6](#) dokumentiert dann eine prototypische technische Umsetzung des entwickelten Verfahrens, um die nachfolgend durchgeführten Simulationsstudien einzuleiten. Die Ergebnisse der Simulationsstudien werden in [Kapitel 7](#) vorgestellt, ausgewertet und interpretiert. Hier werden zunächst formale Performanz- und Effizienzkriterien anhand synthetischer Testdaten betrachtet, bevor schließlich verschiedene Szenarien aus dem oben beschriebenen Anwendungskontext der prädiktiven Einsatzplanung dezentraler Akteure im

Smart Grid untersucht werden. Die Arbeit schließt dann mit einer zusammenfassenden Betrachtung der erzielten Ergebnisse sowie möglicher Anknüpfungspunkte für nachfolgende Arbeiten.

1.6 EIGENE VORARBEITEN

Der Anfertigung der vorliegenden Arbeit gingen einige eigene Vorarbeiten voraus, die nachfolgend kurz dokumentiert werden.

Die grundlegende Idee des hier entwickelten Koordinationsverfahrens wurde 2011 im Rahmen eines Workshops vorgestellt [[HVS11](#)]. Aus den dort gewonnenen Erkenntnissen wurden mit [[Hin11](#)] im gleichen Jahr konkrete Anforderungen an das zu entwickelnde Verfahren formuliert und das Vorhaben der nationalen Fachwelt vorgestellt. Nachfolgend fand Anfang 2012 in [[And+12](#)] eine erste Realisierung des Verfahrens für das Problem der Steuerung eines Kraftwerksparks hinsichtlich der Frequenzhaltung im Elektrizitätsnetz statt. Hierbei handelte es sich um ein im Vergleich zur fahrplanbasierten prädiktiven Einsatzplanung einfacheres Optimierungsproblem, in welchem mit globalem Wissen (in Form der aktuellen Frequenzabweichung) gearbeitet wurde. Die Veröffentlichung erfolgte in Zusammenarbeit mit der Arbeitsgruppe von Herrn Prof. Dr. Reif am Lehrstuhl für Softwaretechnik der Universität Augsburg und resultierte in einem *Best Paper Award*. Das Verfahren wurde dann zu einem vollständig dezentralen Koordinationsansatz ausgebaut, welches nicht auf globales Wissen angewiesen ist und kombinatorische Optimierungsprobleme in verteilter Weise lösen kann, hierbei jedoch auf absolute Kooperationsbereitschaft der dezentralen Anlagen angewiesen ist. Dies wurde Mitte 2012 vorgestellt [[HLS14a](#)]. Eine detaillierte Evaluation dieses Ansatzes wurde mit [[HSL13](#)] vorgenommen. Hier fand auch eine erste Integration einer verteilten mehrkriteriellen Zielfunktion statt, um die Autonomie der dezentralen Akteure während des Optimierungsprozesses zu modellieren. Auf Anfrage wurde dies in einer *extended Version* mit [[HLS14b](#)] nochmals verfeinert. Zwischenzeitig wurden in [[HLS](#)] (noch nicht erschienen) anhand eines alternativen Optimierungsproblems adaptive Fähigkeiten des Verfahrens hinsichtlich dynamischer Optimierungsziele untersucht, und schließlich in [[HBS13](#)] eine Hybridisierung mit dem im Forschungsverbund Smart Nord entwickelten Ansatz der Suchraummodellierung vorgenommen.

Im Vorfeld fanden zudem einige vorbereitende Arbeiten statt, deren Ergebnisse nur indirekt in die vorliegende Arbeit eingeflossen sind. So wurden in [Hin08; HVS09] verschiedene Strategien zur Verhaltensoptimierung eines Anlagenparks hinsichtlich auftretender Lastgangoszillationen nach zentral ausgeübten Steuerungseingriffen untersucht. Dies wurde nachfolgend in einem Feldtest erprobt, die Auswertungen hierzu sind in [Hin+12] zu finden. Und schließlich wurden im Rahmen von studentischen Projektgruppen zwei Softwarewerkzeuge zur Lösung angrenzender Optimierungsprobleme im breiteren Themenfeld der Energieversorgung entwickelt. Diese wurden in [Eys+08] und [Aga+11] veröffentlicht und beide mit dem *Environmental Informatics Prize for Students* der Gesellschaft für Informatik (GI) ausgezeichnet.

2 | GRUNDLAGEN

Die vorliegende Arbeit trägt den Titel „Selbstorganisierte Einsatzplanung dezentraler Akteure im Smart Grid“. Während die Begriffe *Einsatzplanung*, *dezentrale Akteure* und *Smart Grid* im einleitenden Kapitel bereits aufgegriffen und beschrieben wurden, ist die Verwendung des Begriffes *selbstorganisiert* dort bewusst vermieden worden. Stattdessen sind für den in dieser Arbeit zu entwickelnden Ansatz unter anderem die Zieleigenschaften Informations-/Steuerungsdezentralität, Autonomie und Robustheit definiert worden. Im Folgenden wird argumentiert, dass Selbstorganisationsstrategien ein möglicher Weg sind, diese Eigenschaften umzusetzen. Hierzu wird zunächst auf das Thema der verteilten kombinatorischen Optimierung eingegangen, bevor Multiagentensysteme als Mittel zur technischen Umsetzung beschrieben werden und schließlich das Konzept der Selbstorganisation erläutert wird.

2.1 VERTEILTE KOMBINATORISCHE OPTIMIERUNG

Das Ziel der hier betrachteten prädiktiven fahrplanbasierten Einsatzplanung ist es, jeder beteiligten Energieanlage einen Fahrplan für einen begrenzten, in der Zukunft liegenden Zeitraum zuzuweisen, sodass die Menge aller zugewiesenen Fahrpläne eine bestimmte Zielfunktion maximiert (oder minimiert – durch Negation lässt sich das eine in das andere überführen; vereinfachend wird im Folgenden die Formulierung „optimieren einer Zielfunktion“ verwendet). Der betrachtete Zeitraum (auch als Planungshorizont bezeichnet) wird dabei in der Energiewirtschaft zur organisatorischen Vereinfachung in diskrete Zeitintervalle unterteilt und die von einer Energieanlage in einem Zeitintervall erbrachte Wirkleistung mittels Durchschnittsbildung als konstant angenommen [Cra04]. Da zudem die Menge der für eine Energieanlage zur Verfügung stehenden Fahrpläne diskreter Natur ist (bspw. kann ein nicht-modulierendes Mikro-Blockheizkraftwerk in einem Zeitpunkt genau zwei Zustände einnehmen: ein-

oder ausgeschaltet), ergibt sich hiermit insgesamt ein kombinatorisches Optimierungsproblem: Aus einer endlichen, enumerierbaren Menge von Elementen (den realisierbaren Fahrplänen aller Energieanlagen) muss eine Teilmenge ausgewählt werden (für jede Anlage ein Fahrplan), sodass eine gegebene Zielfunktion optimiert wird. Formal betrachtet kann für jeden möglichen Fahrplan eine binäre Entscheidungsvariable modelliert werden, die angibt, ob der entsprechende Fahrplan ausgewählt ist oder nicht. Die binäre Eigenschaft dieser Variablen kann realisiert werden, indem die Variablen als *ganzzahlig* im Wertebereich $[0, 1]$ definiert werden, und somit nur den Wert 0 oder den Wert 1 annehmen können. Solchermaßen diskretisierte Wertebereiche können nach [PS98, S. 313] durch Verwendung von speziellen Koeffizienten auch als Linearkombination dargestellt werden, sodass es sich insgesamt um ein *ganzzahlig lineares* Optimierungsproblem handelt, und damit NP-vollständig ist [PS98, S. 3] (eine vollständige Formulierung des Problems als ganzzahliges lineares Programm wird der Vollständigkeit halber nach der Formalisierung aller Entitäten in [Abschnitt 3.1.1](#) gegeben).

Die Schwierigkeit des Problems ergibt sich nun daraus, dass zwischen den möglichen Belegungen der Variablen Abhängigkeiten existieren. Diese werden einerseits in Form von Nebenbedingungen formuliert und trennen *gültige* von *ungültigen* Lösungen des Optimierungsproblems, und gehen andererseits in die Zielfunktion ein, welche damit verschiedenen Belegungen unterschiedlich *bewerten* kann. Im vorliegenden Fall etwa muss für eine gültige Lösung genau ein Fahrplan für jede Anlage ausgewählt werden, nicht mehr und nicht weniger. Die Aufgabe eines Optimierungsverfahrens ist es dann, aus der Menge der gültigen Lösungen diejenige herauszusuchen, welche die Zielfunktion optimiert. Zur Veranschaulichung hierzu ist es zweckmäßig, alle zu einer bestimmten Anlage gehörenden binären Entscheidungsvariablen unter einer Nebenbedingung zusammenzufassen. Sei bspw. eine Anlage a_i gegeben, und sei $\{x_{ij} \mid 0 \leq j < n_i\}$ die Menge aller binären Entscheidungsvariablen für a_i , wobei n_i die Anzahl der möglichen Fahrpläne von a_i ist. Dann kann mittels einer ganzzahligen Entscheidungsvariable v_i die folgende zusammenfassende Nebenbedingung N_i formuliert werden:

$$N_i : 0 \leq v_i < n_i, v_i \in \mathbb{N} \quad (2.1)$$

Jedem Fahrplan der Anlage a_i wird also ein Index zugeordnet, sodass aus der Be-

legung von v_i unmittelbar der für diese Anlage ausgewählte Fahrplan ersichtlich ist. Wird eine solche Nebenbedingung für jede beteiligte Energieanlage $a_0, \dots, a_{m-1}$ aufgestellt (m ist die Anzahl der beteiligten Anlagen), dann entspricht eine gültige Lösung des Optimierungsproblems einer gültigen Belegung aller Variablen $v_0, \dots, v_{m-1}$ entsprechend der Nebenbedingungen $N_0, \dots, N_{m-1}$. Aus diesen gültigen Belegungen ist nun diejenige herauszufinden, welche die Zielfunktion optimiert. Wie oben angedeutet, tragen die in der Zielfunktion kodierten Abhängigkeiten der Entscheidungsvariablen maßgeblich zur Schwierigkeit des Problems bei. Bei der hier betrachteten prädiktiven fahrplanbasierten Einsatzplanung ist eine Kombination von Einzelfahrplänen gesucht, welche in der Summe ein gewünschtes Wirkleistungsprodukt möglichst exakt abbildet (vgl. [Abschnitt 1.2](#)). Hieraus resultiert aber, dass im Optimierungsprozess alle in einer möglichen Lösung ausgewählten Fahrpläne in Relation zueinander bewertet werden müssen! Es ist somit nicht möglich, für eine einzelne Anlage eine optimale Fahrplanauswahl zu treffen, ohne die Fahrplanauswahlen aller anderen Anlagen zu berücksichtigen – das Problem ist *nicht separierbar*. Dies kann grafisch in Form eines sog. Constraint-Graphen dargestellt werden. Hierzu werden die Entscheidungsvariablen als Knoten, und die in der Zielfunktion kodierten, zwischen den Variablen bestehenden Abhängigkeiten (engl. *constraints*) als Kanten visualisiert. Für das vorliegende Problem ergibt sich für einen exemplarischen Verbund aus acht Anlagen der in [Abbildung 2.1](#) dargestellte Constraint-Graph. Dieser Graph ist vollvernetzt, da bzgl. der Zielfunktion jede Variable von allen anderen abhängig ist. Handelt es sich hierbei um immer denselben Constraint, so kann stattdessen auch von einem einzigen Constraint gesprochen werden, welcher sich über alle Variablen erstreckt (dies ist hier der Fall). Werden bspw. m Variablen betrachtet, so hat dieser Constraint die Arität m . Entsprechend ihrer Arität können Constraints in unterschiedliche Klassen eingeteilt werden, für welche jeweils spezifische Lösungsverfahren existieren (vgl. [\[HBS13\]](#)). Dies wird in den folgenden Abschnitten näher ausgeführt.

2.1.1 Restriktionen

Nebenbedingungen, welche nicht in Abhängigkeiten zwischen verschiedenen Variablen resultieren, sondern gültige von ungültigen Belegungen für eine einzelne Variable trennen, haben die Arität 0 und werden in dieser Arbeit als Restriktionen

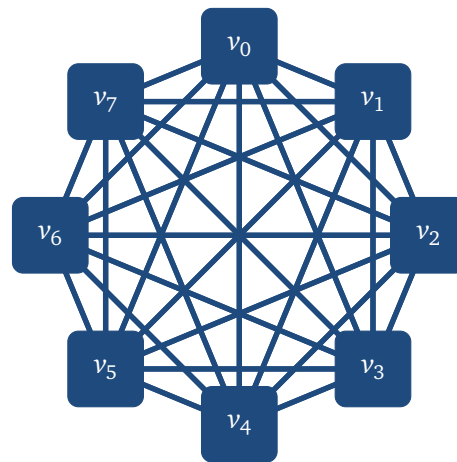


Abbildung 2.1: Constraint-Graph zum kombinatorischen Optimierungsproblem der prädiktiven fahrplanbasierten Einsatzplanung eines Verbundes aus acht Energieanlagen.

bezeichnet. Die oben in [Gleichung 2.1](#) definierte Nebenbedingung N_i ist ein Beispiel einer solchen Restriktion. Je nach verwendetem Optimierungsverfahren kann es problematisch bei diesen Bedingungen sein, dass sie nicht direkt in der Zielfunktion kodiert werden können, sondern stattdessen zusätzlich zur Evaluation der Zielfunktion überprüft werden müssen. Daher wurden verschiedene Möglichkeiten entwickelt, die Behandlung von Restriktionen in Optimierungsverfahren zu integrieren: Die Strafwert Methode erhebt einen Strafwert für jede verletzte Restriktion, welcher negativ in die zu optimierende Zielfunktion eingeht. Bei geschickter Gewichtung dieser Strafwerte in Relation zur restlichen Zielfunktion kann hiermit eine Formulierung gewonnen werden, die frei von Restriktionen ist. Bei bestimmten Optimierungsproblemen ist es alternativ möglich, ungültige Lösungen automatisch zu reparieren, und auf diese Weise eine effiziente Behandlung der Restriktionen zu erreichen [[LV90](#)]. Eine weitere Möglichkeit besteht darin, die Restriktionen als zusätzliche Zielfunktionen zu formulieren, und somit ein von Restriktionen freies Mehrzielproblem zu erhalten. Und schließlich wurde jüngst der sog. Dekoder Ansatz entwickelt, welcher es einem Optimierungsverfahren ermöglicht, den Raum der durchsuchten Lösungen unmittelbar auf gültige Lösungen zu beschränken [[BS12](#)].

2.1.2 Low-Arity Constraints

Nebenbedingungen über nur wenige Variablen werden in dieser Arbeit als Low-Arity Constraints bezeichnet. Hierunter fallen alle Constraints mit einer Arität zwischen 1 und 3. Optimierungsprobleme, welche ausschließlich diesen Constrainttyp beinhalten, sind auch unter dem Begriff *Distributed Constraint Optimization Problem* (DCOP) bekannt. Für diese Probleme existiert aufgrund ihrer praktischen Relevanz eine große Menge von Lösungsverfahren. Eine alternative Sichtweise auf diese Probleme ist mit der Spieltheorie gegeben, welche ein solides theoretisches Fundament zur Untersuchung solcher Probleme liefert. Es sei an dieser Stelle auf einen Beitrag von Chapman et al. [Cha+11] verwiesen, in welchem die etablierten Verfahren aus den Bereichen DCOP und Spieltheorie in strukturierter Weise vorgestellt und aufgearbeitet werden. Relevant ist auch ein Beitrag von Parunak et al. [PWS99], in welchem die Verbindung zu Marktmechanismen hergestellt wird. Diese Lösungsansätze lassen sich jedoch aus Komplexitätsgründen nicht für Constraints mit höherer Arität anwenden. Da zudem diese Art der Constraints in der vorliegenden Problemstellung nicht vorkommt, wird dieser Typus im Folgenden nicht näher betrachtet.

2.1.3 High-Arity Constraints

Erstrecken sich Nebenbedingungen über mehr als drei Variablen, werden sie in dieser Arbeit als High-Arity Constraints bezeichnet. Insbesondere in einem dezentralen Rahmen gehören diese zu den am schwierigsten zu behandelnden Constrainttypen, da sich mit steigender Arität hier ein erhöhter Koordinationsaufwand der beteiligten dezentralen Akteure ergibt. Um dieser Komplexität zu begegnen, sind verschiedene Koordinationsparadigmen denkbar, die je nach gegebener Problemstellung mehr oder weniger effektiv sind. Im folgenden Abschnitt wird eine detaillierte Aufstellung dieser Paradigmen gegeben.

2.1.4 Koordinationsparadigmen

Zur Lösung des gegebenen Optimierungsproblems sind verschiedene Ansätze denkbar, welche sich gemäß ihrer Koordinationsparadigmen kategorisieren lassen. Die folgenden Definitionen richten sich nach [Hou10; NLH10; McN12], Alternativen sind etwa in [WMB05; Pou13] zu finden.

Zentral: Es existiert eine zentrale Instanz mit globalem Wissen über alle Entscheidungsvariablen, deren Wertebereiche sowie der Zielfunktion. Diese Instanz entscheidet somit über alle Variablenbelegungen.

Dezentral: Es existiert eine zentrale Instanz, der sowohl die Zielfunktion des Problems als auch die Menge der Entscheidungsvariablen bekannt ist, die jedoch kein Wissen über die Wertebereiche der Variablen besitzt und auch keine Belegung der Variablen durchführen kann. Dazu existiert für jede Entscheidungsvariable eine lokale autonome Instanz, welche Wissen ausschließlich über die ihr zugeordnete Variable, deren Wertebereich und ggf. auch über die Zielfunktion besitzt, und Belegungen für diese Variable vornehmen kann. Diese lokalen Instanzen können mit der zentralen Instanz kommunizieren. Zur Lösung des Problems ist es erforderlich, dass die lokalen Instanzen ihre Entscheidungen der zentralen Instanz mitteilen, welche diese dann insgesamt bewerten und das Ergebnis wiederum den lokalen Instanzen mitteilen kann. Bestehen zwischen den Variablen Abhängigkeiten in Form von Low-Arity oder High-Arity Constraints, so muss dieser Koordinationsschritt ggf. mehrfach wiederholt werden, bis eine finale Lösung gefunden wurde.

Hierarchisch: Alle Instanzen werden in einer Baumtopologie organisiert, an deren Wurzel die zentrale Instanz steht. Eine Instanz darf hier nur mit den ihr bzgl. der Topologie untergeordneten Instanzen, und mit der ihr übergeordneten Instanz kommunizieren. Dies ist ein Spezialfall des dezentralen Paradigmas, da jede Instanz mit den ihr untergeordneten Instanzen als zentral bzgl. eines Teilproblems gesehen werden kann. Je nachdem, welche Abhängigkeiten (Constraints) zwischen den Variablen bestehen, kann durch die Anordnung der Instanzen eine Aufteilung in Teilprobleme geschaffen werden, welche lokal (d. h. innerhalb eines Zweiges des Baumes) optimiert, und durch Kommunikation mit den hinsichtlich der Topologie übergeordneten Instanzen zu einer Gesamtlösung aggregiert werden können. Je höher die Arität der auftretenden Constraints ist, desto weniger effektiv ist dieses Paradigma.

Verteilt: Dieses Koordinationsparadigma erweitert das dezentrale Paradigma um die Möglichkeit, dass lokale Instanzen auch untereinander kommunizieren können, statt nur mit einer zentralen Instanz. Analog zum hierarchischem

Paradigma können hierdurch Vorteile bzgl. Kommunikations- und Koordinationsaufwand erzielt werden, wenn Teilprobleme ggf. lokal (d. h. mit einer Teilmenge der lokalen Instanzen) optimiert und durch Kommunikation mit anderen lokalen Instanzen oder der zentralen Instanz zu einer Gesamtlösung aggregiert werden können. Im Vergleich zur Hierarchie ist dieses Paradigma etwas flexibler, da die Kommunikationstopologie keinen Baum bilden muss.

Vollständig verteilt: In diesem Paradigma existiert keine zentrale Instanz mehr. Die lokalen Instanzen müssen sich eigenständig koordinieren, und in vollständig verteilter Weise eine Gesamtlösung bilden.

Eine zentrale Koordination wird z. B. in der klassischen Kraftwerkseinsatzplanung (engl. *Unit Commitment*) durchgeführt. Marktmechanismen sind üblicherweise dezentral, und auch die in [Abschnitt 1.3](#) vorgestellten Ansätze COBB sowie *Stigspace* verfolgen das dezentrale Paradigma. Die Ansätze der holonischen VK, AVPP sowie EPOS fallen in die Kategorie der hierarchischen Koordination. Der Ansatz der restriktionslosen Gerätepools weist in seinen Einzelteilen Eigenschaften sowohl des zentralen, des dezentralen als auch des verteilten Paradigmas auf.

Die vollständig verteilte Koordination stellt die am schwierigsten umzusetzende Variante dar, insbesondere wenn High-Arity Constraints vorhanden sind, wie es bei der vorliegenden Problemstellung der Fall ist. Gleichwohl erfüllt dieses Paradigma als einziges zur Gänze die in der Zielsetzung dieser Arbeit angestrebten Eigenschaften (siehe [Abschnitt 1.4](#)). Daher besteht die Herausforderung der vorliegenden Arbeit darin, eben dieses Koordinationsparadigma für das zu entwickelnde Einsatzplanungsverfahren in Form einer Heuristik umsetzen. Entsprechend soll im Folgenden unter der *verteilten kombinatorischen Optimierung* ein kombinatorisches Optimierungsproblem verstanden werden, welches High-Arity Constraints aufweist und mittels eines vollständig verteilten Koordinationsparadigmas heuristisch gelöst wird.

2.2 MULTIAGENTENSYSTEME

Eine naheliegende Möglichkeit zur Realisierung einer verteilten kombinatorischen Optimierung sind Multiagentensysteme (MAS). In einem MAS werden die dezentralen Akteure der Problemstellung jeweils durch einen *Agenten* repräsentiert. Ein

Agent vertritt die Interessen des von ihm repräsentierten Akteurs in automatisierungstechnischer Form, und stellt damit die Verknüpfung der physischen Realität (hier: der Anlage bzw. ihr Betreiber) mit dem zu lösenden mathematischen Optimierungsproblem dar. Im Kontext der Energieversorgung ist die Anwendung von agentenbasierten Techniken eine noch relativ junge Idee [Bec+13], welche mit der Forderung nach Dezentralisierung des Systems aufkam. So ist etwa in einer Veröffentlichung des *Intelligent Systems* Unterkomitees der *IEEE Power Engineering Society* zu lesen [McA+07]:

Generators have a degree of autonomy and cannot be directly affected by external system actors, lending themselves to being represented by agents.

Eine wichtige Eigenschaft von Agenten ist, dass sie keinen gemeinsamen Zustand teilen, sondern durch Nachrichtenaustausch miteinander kommunizieren müssen [RN10, S. 161]. Dies hebt sie koordinations-technisch von klassischen verteilten Systemen ab, in welchen die Berechnungsprozesse mitunter auf einen gemeinsamen Speicher zugreifen können [AW04]. Davon abgesehen sind die existierenden Definitionen von Agenten ebenso vielfältig wie deren Anwendungen. Eine umfassende Übersicht hierzu liefert [DMW10], dort wurden die folgenden vier Punkte als am häufigsten genannte Eigenschaften von Agenten identifiziert:

Autonomie: Agenten sind autonom in dem Sinne, dass sie ihre Aktionen *selbst* auf Basis von internen Entscheidungsprozessen auswählen, und Kontrolle über ihren eigenen Zustand besitzen. Dies unterscheidet sie etwa von „normalen“ Objekten aus softwaretechnischer Sicht, deren Aktionen grundsätzlich durch externe Entitäten aufgerufen werden. Agenten besitzen somit ein *Verhalten*.

Soziales Verhalten: Agenten können mit ihrer Umwelt oder mit anderen Agenten in Kontakt treten, um ihre Aufgabe zu erfüllen. Dies entspricht der oben bereits formulierten Eigenschaft des Fehlens eines gemeinsamen Zustands.

Reaktivität: Agenten sind in der Lage, bestimmte Dinge wahrzunehmen und auf diese zu reagieren. Die betrifft insbesondere den Informationsaustausch mit anderen Agenten.

Proaktivität: Agenten besitzen ein Ziel, das sie verfolgen. Ihr Verhalten ist darauf ausgerichtet, dieses Ziel zu erreichen.

Die konkrete Definition der in der vorliegenden Arbeit modellierten Agenten wird im Rahmen der Problemformalisierung im folgenden Kapitel aufgestellt werden. In [Abschnitt 3.2](#) wird zudem näher auf die Umsetzung der obigen Eigenschaften, sowie auf das resultierende MAS als Ganzes und die damit verbundene Interaktionssituation zwischen Agenten eingegangen.

2.3 SELBSTORGANISATION

Ganz allgemein formuliert beschreibt das Konzept der Selbstorganisation (SO) solche Systeme, die ihre Organisation *aus sich selbst heraus* modifizieren. Der Begriff *Organisation* beschreibt hierbei die interne Struktur des Systems, welche sowohl physischer (bspw. die räumliche Anordnung von Systemelementen) als auch logischer Natur sein kann (bspw. organisatorische Beziehungen innerhalb des Systems). Gemäß dieser allgemeinen Formulierung könnte jedoch fast jedes beliebige System als selbstorganisierend bezeichnet werden, wenn der Betrachtungswinkel entsprechend gewählt wird.¹ Je nach Disziplin wird die SO daher mit unterschiedlichem Fokus betrachtet. So wird SO unter anderem in der Biologie mit der physischen Struktur- bildung in Zusammenhang gebracht, im Kontext von thermodynamischen Prozessen mit dem mathematischen Begriff der Entropie verknüpft und in der Psychologie als Erklärungsmodell zur Konstruktion mentaler Modelle verwendet. Referenzen zu diesen und weiteren Interpretationen sind bspw. in [[Hey01](#); [Ger07](#)] zu finden. Eine in Anbetracht dieser Vielfalt an Sichtweisen wichtige Erkenntnis zum Konzept der SO stammt von Ashby, der sich in [[Ash62](#)] unter anderem mit den Begriffen *System* und *Organisation* beschäftigt, und diese in Beziehung zum *Beobachter* setzt:

It follows that a substantial part of the theory of organization will be concerned with *properties that are not intrinsic to the thing but are relational between observer and thing*.

¹Je nach Standpunkt kann auch argumentiert werden, dass *kein* System selbstorganisierend sein kann, oder dass *alle* Systeme selbstorganisierend sind. Diese philosophische Debatte soll jedoch an anderer Stelle ausgetragen werden.

Nach Ashby ist „selbstorganisierend“ keine absolute Eigenschaft, sondern entsteht erst durch die Interpretation des Beobachters mittels der Attributierung von „guten“ und „schlechten“ Organisationen, bzw. von „organisiert“ und „unorganisiert“. Der Gedanke wird von Gershenson fortgeführt und in [Ger07] treffend formuliert:

Self-organization is a way of modeling systems, not a class of systems.

Von einem ingenieurstechnischen Standpunkt aus ist für Gershenson ein System genau dann selbstorganisierend, *wenn die Elemente des Systems dahingehend entwickelt werden, sodass sie dynamisch und autonom – durch ihr Verhalten und durch Interaktionen untereinander – auf globaler Systemebene ein Problem lösen bzw. eine Funktion ausführen*. Selbstorganisierende Systeme zeichnen sich nach dieser Beschreibung also durch eine Menge von Elementen innerhalb des Systems aus, die ein Verhalten besitzen und miteinander interagieren können, und diese Fähigkeiten nutzen, um ein globales Ziel (auf Ebene des Gesamtsystems) zu erreichen. Dieser Standpunkt hat starke Ähnlichkeit mit dem in der vorliegenden Arbeit angestrebten vollständig verteilten Koordinationsparadigma zur Lösung des gegebenen kombinatorischen Optimierungsproblems. Die Systemelemente entsprechen dabei den hier vorliegenden dezentralen Akteuren, und das globale zu erreichende Ziel ist die Optimierung der Zielfunktion, d. h. das Erfüllen eines gewünschten Wirkleistungsproduktes durch Auswahl entsprechender lokaler Fahrpläne für den Planungshorizont. Der Einsatz von Agenten (siehe [Abschnitt 2.2](#)) zur technischen Realisierung scheint ebenso passend, weisen diese doch gerade die Eigenschaften des autonomen Verhaltens und der Interaktion auf, welche auch von den Elementen eines selbstorganisierenden Systems gefordert werden. Aus diesem Grund wird das Konzept der Selbstorganisation in [Kapitel 3](#) als Herangehensweise zur Ausgestaltung des Einsatzplanungsverfahrens herangezogen.

2.3.1 Nichtlinearität

Wie oben beschrieben, besitzen die Elemente eines selbstorganisierenden Systems gemäß der hier verwendeten Definition ein bestimmtes Verhalten und interagieren in der Regel miteinander. Die hieraus resultierende inhärente Dynamik des Systems führt zum Effekt der Nichtlinearität, was es einerseits schwierig macht, das Systemverhalten auf globaler Ebene exakt vorherzusagen, und andererseits, eine

hinsichtlich einer gewünschten Systemfunktion zielführende Interaktionsstrategie zu entwickeln. Aus dem Wissenschaftsgebiet der komplexen Systeme, welche große Ähnlichkeit mit selbstorganisierenden Systemen aufweisen, kann zur Erklärung das Prinzip der Rückkopplung herangezogen werden [Cam+03; Ger07]: Die Effekte bzw. Ergebnisse eines Interaktionsprozesses innerhalb des Systems fließen direkt wieder in das System und gehen als Eingangsgröße in die darauffolgenden Interaktionsprozesse ein. Der Systemzustand zu einem bestimmten Zeitpunkt kann also nicht unmittelbar von den initialen Bedingungen, unter denen das System startete, und den Eingaben, die das System von außen erfährt, abgeleitet werden. Stattdessen ist der gesamte Verlauf des Systems seit dem Start zu betrachten, da dieser maßgeblich zum aktuellen Zustand beiträgt. Die hier auftretenden Rückkopplungen können in positiv (d. h. verstärkend) und in negativ (hemmend) unterteilt werden. Während positive Rückkopplung eine starke Sensitivität vom aktuellen Zustand beschreibt und zum *Chaos* führt, bewirkt negative Rückkopplung eine Dämpfung der Systemdynamik und resultiert im *Equilibrium*.

In Bezug zur Problemstellung der vorliegenden Arbeit weisen beide Effekte nützliche Eigenschaften auf. Da eine heuristische Herangehensweise entwickelt werden soll, ist eine geschickte Exploration des Lösungsraums des Optimierungsproblems durchzuführen. Hierzu kann die positive Rückkopplung sinnvolle Impulse geben. Auf der anderen Seite ist eine garantierte Konvergenz zu einer möglichst guten gemeinsamen Lösung in Form eines gültigen Einsatzplanes gefordert. Dieser finale Zustand stellt somit ein *Equilibrium* des Systems dar, was durch negative Rückkopplung erreicht werden kann. Beide Arten der Rückkopplung werden daher bei der Entwicklung des Lösungsansatzes im folgenden Kapitel wieder aufgegriffen.

2.3.2 Variation

Eine weitere potenzielle Quelle für die Dynamik eines selbstorganisierenden Systems ist die sog. *Variation* im Verhalten von ansonsten gleichartigen Systemelementen. Dabei handelt es sich um kleine Verhaltensabweichungen, die etwa durch verfälschte Informationen oder zeitliche Verzögerungen induziert werden und stochastischen (d. h. zufälligen) Ursprung haben. Ursprünglich von Ashby im Kontext der Selbstorganisation untersucht, wurde dieses Konzept im Rahmen von kooperativen Multiagentensystemen in [CRW11] wieder aufgegriffen, und dort als *inter-agent variation*

bezeichnet. Die Autoren zeigten, dass Variation den Selbstorganisationsprozess hinsichtlich der Erfüllung der globalen Systemfunktion positiv unterstützen kann. Dies wurde in [And+12] auf die energiewirtschaftliche Domäne übertragen und anhand eines Systems bestehend aus flexiblen Erzeugern zur kooperativen Frequenzhaltung im Elektrizitätsnetz demonstriert. Dazu wurden zwei verschiedene Koordinationsansätze untersucht, von denen einer völlig kommunikationslos arbeitete (und somit keine Interaktionen zwischen den Systemelementen beinhaltete), während der andere existenziell auf der direkten Koordination durch Kommunikation basierte. Für beide Verfahren wurde deutlich, dass die Einführung von moderater Variation zu einer größeren Zielerfüllung führte und sich somit jeweils vorteilhaft auf die selbstorganisierte Koordination auswirkte. Bei sehr starker Variation jedoch kehrte sich dieser positive Einfluss zumindest im kommunikationslosen Ansatz um.

In der vorliegenden Arbeit ist Variation durch die Autonomie der dezentralen Akteure gegeben. Hinsichtlich der Robustheit des Systems ist also zu untersuchen, welchen Einfluss diese Variation auf den entwickelten Koordinationsansatz hat. Dies wird im Rahmen der Evaluation in [Kapitel 7](#) durchgeführt.

3

SELBSTORGANISIERTE EINSATZPLANUNG

Die Problemstellung der vorliegenden Arbeit beinhaltet die Entwicklung einer Einsatzplanungsheuristik unter Verwendung des vollständig verteilten Koordinationsparadigmas. Wie im vorigen Kapitel beschrieben, besitzen selbstorganisierende Systeme Eigenschaften, die dies ermöglichen können. Das grundlegende Prinzip der Selbstorganisation ist jedoch nur schwer in eine ingenieurwissenschaftliche Herangehensweise zu integrieren, und es existiert kein für alle Situationen anwendbares Vorgehensmodell zur Konstruktion selbstorganisierender Systeme. Beispielsweise wird in [Ger07] eine abstrakte Methode vorgestellt, die auf der Reduktion von „negativer Interaktion“ (engl. *friction of interactions*) beruht. Es wird jedoch kein strukturiertes Vorgehen genannt, wie dies im Einzelnen erfolgen kann. In [SOR07] wird vorgeschlagen, die funktionalen Fähigkeiten der Akteure im System mittels einer selbstorganisierenden Kontrollebene dynamisch zu verwalten. Dies wird in [See11] mit dem *Restore Invariant Approach* zu einem Softwareentwicklungsansatz für selbstorganisierende Ressourcenflusssysteme ausgebaut. Der Fokus liegt hierbei auf der adaptiven Reorganisation der Aufgabenverteilung bei spontanen externen Ereignissen, um einen definierten Korridor von gewünschtem globalem Systemverhalten garantieren zu können. Dies ist für die vorliegende Problemstellung jedoch unpassend, da die Aufgaben aller Akteure fix sind. Die *Observer/Controller* Architektur aus dem Bereich des *Organic Computing* setzt diesen Gedanken für allgemeine selbstorganisierende Systeme fort und definiert eine Methode zur adaptiven Kontrolle eines observierten selbstorganisierenden Systems hinsichtlich gewünschter globaler Systemeigenschaften [Ste+13b]. Hier liegt der Fokus auf der Gestaltung einer adaptiven Kontrollschleife für ein observiertes System mit auf lokalem Verhalten basierender Dynamik, während das Ziel der vorliegenden Arbeit aber die Entwicklung eben dieses zugrundeliegenden Systems darstellt. Eine Reihe von verschiedenen Methodenbau-

steinen wie bspw. analytische Betrachtungen, Markov-Modelle oder evolutionäre Algorithmen wird in [Elm+09] vorgestellt. Das dabei gezeigte exemplarische Vorgehensmodell zum Design selbstorganisierender Systeme wirkt jedoch wie eine wahllose Aneinanderreihung einzelner Methodenbausteine. In der „Forschungs-Roadmap zur Konstruktion selbstadaptiver Systeme“ in [Che+09] werden Kontrollschleifen als grundlegende Methode vorgeschlagen. Die Kontrollschleifen ähneln stark dem prinzipiellen Verhaltensmodell von intelligenten Agenten (siehe etwa [RN10, Kap. 2] oder [Wei13, Kap. 2]). Ein darauf basierendes Vorgehensmodell wird nicht gegeben.

Eine deskriptive Herangehensweise auf Basis eines Fragenkataloges liefert [MGK05, S. 173]. Der Katalog umfasst die folgenden Bereiche:

1. Ziele des Systems
2. Umwelt
3. Externe Störereignisse
4. Entitäten
5. Interaktionen
6. Selbstorganisationsmechanismus

Der Katalog ist ursprünglich als Beschreibungsmodell selbstorganisierender Systeme gedacht, kann aber auch als Vorgehensmodell interpretiert werden, in welchem das Systemdesign anhand der Beantwortung der in den dargestellten Teilbereichen enthaltenen Fragen durchgeführt wird.

Im Gegensatz zu diesem recht allgemein gehaltenen Ansatz wird in [PB05] ein konkretes Modell vorgestellt, mit dem zielgerichtete Selbstorganisation erreicht werden soll. Der grundlegende Ablauf ist in [Abbildung 3.1](#) dargestellt. Hier wird mittels vier verschiedener Designphasen ein Interaktionsprotokoll definiert, dessen Durchführung die gewünschte globale Systemfunktion (engl. *overall networking function*) realisiert. Der Designprozess sieht dabei vor, ausgehend von der zu erfüllenden globalen Systemfunktion zunächst die beiden Paradigmen #1 (*Local behavior rules*) und #2 (*Implicit coordination*) anzuwenden, um den Kern des Selbstorganisationsmechanismus auszugestalten. Beide Paradigmen beinhalten sich iterativ wiederholende Designschritte. Anschließend wird dann Paradigma #3 (*Monitoring algorithm*) umgesetzt, um adaptive Fähigkeiten im Hinblick auf eine dynamische Umwelt zu

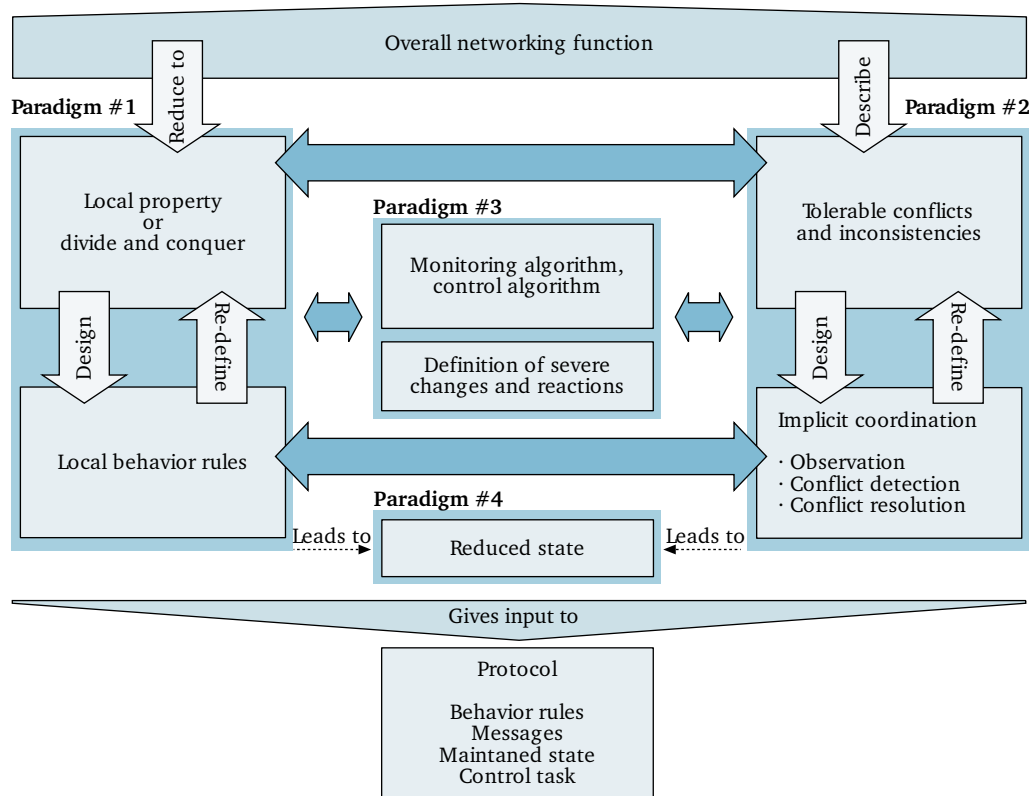


Abbildung 3.1: Designprozess eines selbstorganisierenden Systems, modifiziert nach [PB05].

erreichen. Als Konsequenz aus den ersten beiden Paradigmen soll sich mit Paradigma #4 (*Reduced state*) schließlich durch die Vemeidung von langlebigen (globalen) Zustandsinformationen ein höherer Grad an Selbstorganisation ergeben.

Im ersten Schritt (nach Spezifikation der globalen Systemfunktion) hat der Systemdesigner die Wahl, ob mit Paradigma #1 oder #2 begonnen wird. Diese Wahl hängt maßgeblich vom Typus des Zielsystems ab. Allgemein kann Paradigma #1 eher als *Top-Down* Vorgehen betrachtet werden, in welchem die globale Systemfunktion soweit in ihre Bestandteile zerlegt wird, dass sie mittels lokaler Verhaltensregeln erfüllt werden kann. Paradigma #2 hingegen ist eher ein *Bottom-Up* Vorgehen, da hier inhärentes lokales Verhalten als gegeben betrachtet wird, und die bei der Erfüllung der globalen Systemfunktion auftretenden Konflikte und Inkonsistenzen im Designprozess durch Anpassung dieses lokalen Verhaltens schrittweise aufgelöst werden.

In der vorliegenden Arbeit werden die beiden zuletzt genannten Ansätze zur Entwicklung selbstorganisierender Systeme kombiniert, da sie Methoden zur Verfügung stellen, um den grundlegenden Mechanismus zur Erlangung gewünschter selbstorganisierender Eigenschaften in zielgerichteter Weise zu gestalten. Hierzu werden zunächst die initial gegebenen Bestandteile und Eigenschaften des Systems inklusive der zu realisierenden globalen Systemfunktion mittels der Punkte 1.–4. des Fragenkataloges aus [MGK05] beschrieben und formalisiert (Abschnitt 3.1–3.3). Anschließend wird auf Basis der daraus abgeleiteten Freiheitsgrade bzgl. des Systemdesigns und mit Hilfe des Designprozesses aus [PB05] ein Interaktionsprotokoll erstellt, mit welchem das System selbstorganisierende Eigenschaften erlangt und im Zuge dessen die gewünschte globale Systemfunktion realisiert (Abschnitt 3.4–3.9). Dabei werden zugleich implizit die verbleibenden Punkte des Fragenkataloges ausgeführt.

3.1 DAS PRÄDIKTIVE EINSATZPLANUNGSPROBLEM

Wie in Abschnitt 1.4 beschrieben, wird der Handel von Wirkleistungsprodukten an einer Strombörse durch einen Verbund von Energieanlagen als motivierender Anwendungsfall für den zu entwickelnden Koordinationsansatz herangezogen. Hierzu wird sowohl der betrachtete Verbund aus Energieanlagen, als auch das zu erfüllende

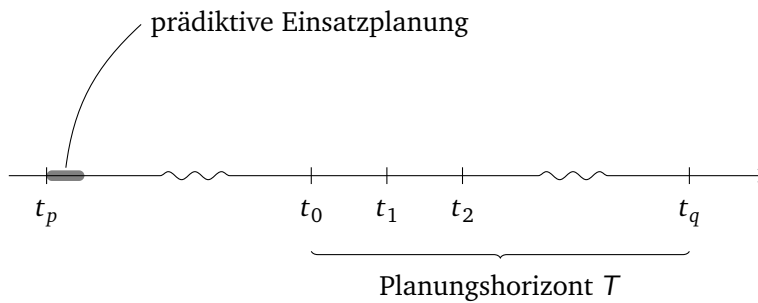


Abbildung 3.2: Zeitliche Einordnung der prädiktiven Einsatzplanung.

Wirkleistungsprodukt als gegeben vorausgesetzt. Das Wirkleistungsprodukt bezieht sich dabei auf eine in der Zukunft gelegene Zeitspanne (dem Planungshorizont). Das Ziel des Koordinationsansatzes ist es nun, für jede Anlage des Verbundes einen Fahrplan über dem Planungshorizont zu finden, sodass der resultierende Einsatzplan des Verbundes den durch das Wirkleistungsprodukt definierten Wirkleistungsverlauf produziert (die Begriffe werden nachfolgend noch genauer definiert, vgl. hierzu auch [Son+12]). Dabei sollen im Sinne des Autonomierhalts zudem lokale Präferenzen der Energieanlagen bzgl. ihrer Fahrpläne so weit wie möglich eingehalten werden. Dieser Prozess wird als *prädiktive Einsatzplanung* bezeichnet, da ausgehend von Annahmen über die realisierbaren Fahrpläne der Energieanlagen eine dem Planungshorizont vorgelagerte Planung vorgenommen wird. In [Abbildung 3.2](#) ist zur Veranschaulichung der zeitliche Ablauf dargestellt. Dieses prädiktive Einsatzplanungsproblem wird nun formal beschrieben. Hierzu werden zunächst die elementaren Systembestandteile wie Energieanlagen, Verbundteilnehmer etc. definiert, und darauf basierend das eigentliche Optimierungsproblem hergeleitet. Der Abschnitt schließt mit der formalen Aufstellung des prädiktiven Einsatzplanungsproblems in [Definition 3.1.13](#). Eine ähnliche Formalisierung, jedoch mit Fokus auf die reaktive Einsatzplanung, ist in [\[Trö10\]](#) zu finden.

Definition 3.1.1 (Energieanlage)

Als *Energieanlage* (oder kurz: Anlage) wird eine technische Anlage bezeichnet, die elektrische Energie wandeln oder speichern (und wieder abgeben) kann.

Der Fokus des vorliegenden Anwendungsfalles liegt auf dezentralen Kleinanlagen, wie etwa Kraft-Wärme-Kopplungsanlagen (KWK-Anlagen, bspw. Mikro-Blockheizkraftwerke). In dem hier betrachteten System wird jede Anlage stellvertretend für ihren Betreiber durch einen Agenten repräsentiert. Die folgende Definition eines Agenten ist auf den Anwendungsfall zugeschnitten, entspricht aber prinzipiell dem Verständnis der *Foundation for Intelligent Physical Agents* (FIPA) [DMW10, S. 3].

Definition 3.1.2 (Agent)

Ein *Agent* ist ein Softwareprogramm, welches Kenntnisse über die möglichen Betriebsweisen der von ihr repräsentierten Energieanlage besitzt und autonome Entscheidungen hinsichtlich der Steuerung der Anlage treffen kann.

Die möglichen Betriebsweisen einer Anlage werden auch als Handlungsspielraum (engl. *schedule space*, vgl. [BRS11a]) bezeichnet. Die genauen Kenntnisse und Handlungsoptionen eines Agenten in Bezug auf die entsprechende Energieanlage sowie das sie umgebende System stellen die Freiheitsgrade des zu entwerfenden selbstorganisierenden Systems dar und werden in den nächsten Abschnitten näher spezifiziert.

Der Einfachheit halber wird der Begriff Energieanlage im Folgenden synonym zu dem sie vertretenden Agenten verwendet. Zusammen bilden diese einen Verbundteilnehmer.

Definition 3.1.3 (Verbund, Verbundteilnehmer)

Sei $A = \{a_0, \dots, a_{m-1}\}$ mit $m \in \mathbb{N}$ eine endliche Menge von Bezeichnern, wobei jeder Bezeichner eine Energieanlage eineindeutig identifiziert. O.B.d.A. wird hier angenommen, dass A total geordnet ist.

Stehen die Anlagen aus A hinsichtlich eines gemeinsam zu erfüllenden Ziels in einer wohlgesinnten Haltung zueinander, bildet A einen *Verbund* der Größe $m = |A|$.

Die einzelnen Energieanlagen a_i mit $0 \leq i < m$ werden dann als *Verbundteilnehmer* bezeichnet.

Aus der Totalordnung der in A enthaltenen Bezeichner ergibt sich insbesondere, dass ein Verbundteilnehmer eindeutig über die Position seines Bezeichners in A identifiziert werden kann. Im Folgenden werden daher die einem Verbundteilnehmer zugesprochenen Eigenschaften vereinfachend mit dem Index i versehen, um die Zugehörigkeit zu einem Verbundteilnehmer a_i anzudeuten.

Definition 3.1.4 (Verbundsprecher)

Sei A ein Verbund. Um den Verbund nach außen hin zu repräsentieren, sei ein einzelner Verbundteilnehmer $\hat{a} \in A$ der *Verbundsprecher*. Im Gegensatz zu den anderen Verbundteilnehmern ist dem Verbundsprecher die Menge der Bezeichner aller Verbundteilnehmer seines Verbundes bekannt. Zudem besitzt er initial als einziger Verbundteilnehmer Kenntnis über das zu realisierende Wirkleistungsprodukt (vgl. [Definition 3.1.7](#) weiter unten). Der Verbundsprecher fungiert somit als Verwalter des Verbundes, sowie als Kommunikationsschnittstelle für externe Systeme.

Mögliche verteilte Verfahren zur Wahl eines Verbundsprechers sind im Bereich der *Leader Election* Algorithmen zu finden. Die konkrete Vorgehensweise bei der Wahl des Verbundsprechers ist für das vorliegende System jedoch nicht relevant, da der Verbund mit seinen Eigenschaften als gegeben betrachtet wird. Daher wird an dieser Stelle nicht näher auf solche Verfahren eingegangen, sondern auf entsprechende Literatur verwiesen (z. B. [[Mat89](#); [Lyn96](#)]). Außer den beschriebenen Verbundteilnehmern (von denen einer zusätzlich die Aufgabe des Verbundsprechers übernimmt) existieren keine weiteren Entitäten im System.

Die Existenzgrundlage eines Verbundes ist nach [Definition 3.1.3](#) ein für die Verbundteilnehmer gemeinsames Ziel. Im Kontext des betrachteten Systems ist dies die Realisierung eines Wirkleistungsproduktes, welches sich aus einem Planungshorizont und einem darüber zu produzierenden Wirkleistungsverlauf zusammensetzt.

Definition 3.1.5 (Planungsintervall, Planungshorizont)

Sei $T = (t_0, \dots, t_{q-1})$ eine endliche Folge von äquidistanten Zeitpunkten mit $t_k \in \mathbb{N}$ für $0 \leq k < q$, $q \in \mathbb{N}$. Das halboffene Intervall $[t_k, t_{k+1})$ beschreibt hierbei ein *Planungsintervall* der Länge Δt , identifiziert durch t_k .

Dann ist T ein *Planungshorizont* der Länge $q = |T|$ (vgl. [Abbildung 3.2](#)).

Ein in der Energiewirtschaft häufig vorkommender Planungshorizont ist beispielsweise in der Tageseinsatzplanung zu finden [[Cra04](#)] und umfasst 24 Stunden in einer zeitlichen Auflösung von Viertelstundenintervallen, sodass sich 96 Planungsintervalle $T = (t_0, \dots, t_{95})$ mit $\Delta t = 15$ Minuten ergeben [[Trö10](#)].

Definition 3.1.6 (Wirkleistungsverlauf)

Ein *Wirkleistungsverlauf* ist eine Abbildung π , die jedem Planungsintervall eines Planungshorizontes eine konstante Wirkleistung $P_{el} \in \mathbb{R}$ zuordnet:

$$\pi : T \rightarrow \mathbb{R}, \quad t_k \mapsto P_{el}$$

Damit kann π verkürzt als reellwertiger Vektor im $|T|$ -dimensionalen Raum betrachtet werden:

$$\pi \in \mathbb{R}^{|T|}$$

Die vereinfachte Darstellung von Wirkleistungsverläufen als $|T|$ -dimensionale Vektoren ermöglicht es, diese in einem technischen System auf einfache Weise zu speichern und zu kommunizieren.

Definition 3.1.7 (Wirkleistungsprodukt)

Ein *Wirkleistungsprodukt* ist ein spezieller Wirkleistungsverlauf, der einem Verbund als Zielvorgabe dient. Dieser Zielverlauf wird mit ζ bezeichnet.

Um ein Wirkleistungsprodukt zu realisieren, muss für den Verbund ein entsprechender Einsatzplan gefunden werden. Dieser setzt sich aus den Fahrplänen der einzelnen Verbundteilnehmer zusammen.

Definition 3.1.8 (Fahrplan)

Sei a_i ein Verbundteilnehmer, und sei T der betrachtete Planungshorizont. Ein von a_i über T produzierbarer Wirkleistungsverlauf heißt *Fahrplan* und wird mit θ_i bezeichnet.

Streng genommen handelt es sich bei einem Fahrplan um eine Vorgabe, die im Erfüllungszeitraum (also nach der Planungsphase, bei Eintritt in den Planungshorizont) von der Anlage in einen tatsächlichen Wirkleistungsverlauf umgesetzt werden muss. Dabei kann es aufgrund von technischen Einschränkungen, Prognoseabweichungen oder unerwarteten Ereignissen zu Abweichungen vom Fahrplan kommen. Dies liegt jedoch außerhalb der prädiktiven Planung und muss in einer nachgelagerten reaktiven (oder auch: kontinuierlichen, vgl. [Son+12]) Einsatzplanung behandelt werden. In [Abschnitt 3.8](#) wird hierzu in kurzer Form auf die mögliche Adaptivität eines selbstorganisierenden Systems zu diesem Zwecke eingegangen. Für die folgenden

Betrachtungen soll ein Fahrplan zunächst aber als ein uneingeschränkt realisierbarer Wirkleistungsverlauf betrachtet werden, der sich auf eine einzelne Energieanlage bezieht.

Definition 3.1.9 (Einsatzplan, Verbundfahrplan)

Sei A ein Verbund der Größe m und seien $\{\theta_0, \dots, \theta_{m-1}\}$ die Fahrpläne der einzelnen Verbundteilnehmer. Dann wird $\{\theta_0, \dots, \theta_{m-1}\}$ als *Einsatzplan* bezeichnet. Die Summe aller Fahrpläne des Einsatzplanes ist mit $\Theta = \sum \theta_i$ gegeben und heißt *Verbundfahrplan*.

Das zu lösende Optimierungsproblem ergibt sich nun daraus, dass die an einem Verbund beteiligten Energieanlagen Freiheitsgrade bezüglich ihrer realisierbaren Fahrpläne besitzen. Beispielsweise ist ein modulierendes Mikro-Blockheizkraftwerk in der Lage, im Rahmen der anlagentechnischen Restriktionen pro Planungsintervall unterschiedliche Leistungsstufen zu fahren [Tho07]. Die Nutzung eines Pufferspeichers für die produzierte thermische Energie liefert hierbei durch die Entkopplung vom Wärmebedarf weitere Freiheitsgrade. Diese Flexibilität von Verbundteilnehmern ermöglicht es, durch geschickte Kombination von Fahrplänen einen Verbundfahrplan zu erzeugen, der das Wirkleistungsprodukt möglichst genau widerspiegelt. Um diesen Sachverhalt unabhängig von bestimmten Anlagentypen zu modellieren, wird jedem Verbundteilnehmer ein abstraktes Suchraummodell zugeordnet, innerhalb dessen der entsprechende Agent nach passenden Fahrplänen für die von ihm repräsentierte Energieanlage hinsichtlich der Realisierung des gemeinsamen Wirkleistungsproduktes suchen kann.

Definition 3.1.10 (Suchraummodell, Suchraum)

Sei a_i ein Verbundteilnehmer. Dann beschreibt die Funktion σ_i das zu a_i gehörende *Suchraummodell*, welches einen hypothetischen Wirkleistungsverlauf π auf einen möglichst ähnlichen, durch die Anlage realisierbaren Fahrplan abbildet:

$$\sigma_i : \mathbb{R}^{|T|} \rightarrow S_i \subset \mathbb{R}^{|T|}, \quad \pi \mapsto \theta_i$$

Dabei bezeichnet die endliche Menge S_i der von der Anlage realisierbaren Fahrpläne den *Suchraum* des Verbundteilnehmers.

Es wird im Rahmen des vorliegenden Systems also davon ausgegangen, dass für jeden Verbundteilnehmer eine restriktionsfreie Darstellung des Handlungsspielraumes der jeweiligen Anlage in Form eines Suchraums vorliegt, in welchem über das Suchraummodell nach passenden Fahrplänen gesucht werden kann. Restriktionsfrei bedeutet hier, dass dieses Suchraummodell in Bezug auf den betrachteten Planungshorizont ausschließlich realisierbare Fahrpläne enthält (siehe [Abschnitt 2.1.1](#)). Auf diese Weise wird von anlagentechnischen Nebenbedingungen (wie bspw. minimale/maximale Laufzeiten eines Erzeugers innerhalb bestimmter Planungshorizonte, vgl. [[Tho07](#); [Trö10](#); [BRS10](#)]) abstrahiert. Auf die tatsächliche Realisierung des Suchraummodells sowie den Begriff der Ähnlichkeit von Wirkleistungsverläufen wird in [Abschnitt 6.4](#) näher eingegangen.

Die Freiheitsgrade der Verbundteilnehmer ermöglichen es somit, verschiedene realisierbare Einsatzpläne für den Verbund kombinatorisch zu erzeugen. Diese Einsatzpläne können hinsichtlich der Erfüllung des Wirkleistungsproduktes mittels einer globalen Gütefunktion bewertet werden.

Definition 3.1.11 (Globale Gütefunktion, globale Güte)

Eine *globale Gütefunktion* ist eine Abbildung f , die einem Tupel aus Wirkleistungsprodukt und Verbundfahrplan einen Gütewert zuordnet:

$$f : \mathbb{R}^{|T|} \times \mathbb{R}^{|T|} \rightarrow \mathbb{R}, \quad (\zeta, \Theta) \mapsto v$$

Der Funktionswert $f(\zeta, \Theta) = v$ heißt *globale Güte* von Θ hinsichtlich ζ .

Die Maximierung dieser globalen Güte stellt somit das primäre Optimierungsziel des Systems dar und beinhaltet im vorliegenden Fall ausschließlich die möglichst genaue Reproduktion des Wirkleistungsproduktes. Prinzipiell ist jedoch auch die zusätzliche Betrachtung von ökonomischen oder ökologischen Kriterien denkbar. Dazu könnte die obige Funktion etwa durch eine *utility function* ersetzt werden, welche eine Gewichtung einzelner Kriterien vornimmt. Auch eine lexikografische Zielfunktion zur Integration mehrerer Kriterien ist möglich, siehe hierzu bspw. [[EG00](#)].

In Ergänzung zu dem globalen Ziel besitzen Verbundteilnehmer zudem lokale Präferenzen bzgl. ihrer Fahrpläne. Diese können ebenfalls ökonomischer/ökologischer, oder aber auch technischer Natur sein, und stellen sekundäre Optimierungsziele des Systems dar. Exemplarische Beispiele hierzu sind: Minimierung der Betriebskosten

(etwa verursacht durch Rohstoffverbrauch), Minimierung des Schadstoffausstoßes oder Minimierung des Verschleißes durch multiple Ein-/Ausschaltvorgänge (für weitere Beispiele siehe [Tho07; Trö10; Win09]). Analog zu [Trö10] werden Präferenzen im Folgenden anhand ihres Erfüllungsgrades bewertet.

Definition 3.1.12 (Erfüllungsfunktion bzgl. lokaler Präferenzen, lokale Güte)

Sei a_i ein Verbundteilnehmer. Dann ist die *Erfüllungsfunktion bzgl. lokaler Präferenzen* des Verbundteilnehmers eine Abbildung g_i , die einem Fahrplan einen lokalen Gütewert zuordnet:

$$g_i : \mathbb{R}^{|T|} \rightarrow \mathbb{R}, \quad \theta_i \mapsto v$$

Der Funktionswert $g_i(\theta_i) = v$ heißt *lokale Güte* von θ_i hinsichtlich der Präferenzen des Verbundteilnehmers.

Aus der Aggregation der globalen Gütefunktion sowie aller lokalen Erfüllungsfunktionen ergibt sich schließlich das prädiktive Einsatzplanungsproblem (PEP).

Definition 3.1.13 (Prädiktives Einsatzplanungsproblem, PEP)

Das *prädiktive Einsatzplanungsproblem* besteht darin, für einen Verbund A einen Verbundfahrplan Θ zu finden, sodass sowohl die globale Zielfunktion f hinsichtlich eines gegebenen Wirkleistungsproduktes ζ , als auch alle existierenden Erfüllungsfunktionen g_i bzgl. lokaler Präferenzen der jeweiligen Verbundteilnehmer a_i optimiert werden.

Die gleichzeitige Optimierung der globalen Gütefunktion sowie der lokalen Erfüllungsfunktionen ist in dieser Formulierung bewusst unspezifisch gehalten, da es dem Lösungsverfahren überlassen wird, dafür eine sinnvolle Herangehensweise zu wählen. In [Abschnitt 3.6.4](#) wird näher auf diesen Aspekt eingegangen.

3.1.1 Formulierung als ganzzahliges lineares Optimierungsproblem

Das PEP ist nach obiger Definition ein kombinatorisches Optimierungsproblem und lässt sich daher nach Einführung von binären Entscheidungsvariablen für die Wahl von Anlagenfahrplänen als ganzzahliges lineares Optimierungsproblem darstellen (siehe [Abschnitt 2.1](#)). Werden die lokalen Präferenzen als Zielkriterium außer Acht

gelassen, ähnelt es stark dem Multiple-Choice Subset-Sum Problem (MC-SSP, siehe [Pis99]) mit mehrdimensionalen Eingaben: Aus einer Menge von Klassen soll je ein (mehrdimensionales) Element ausgewählt werden, sodass die Aggregation der ausgewählten Elemente möglichst genau einem vorgegeben (mehrdimensionalen) Zielelement entspricht, dieses wertmäßig jedoch nicht überschreitet. Für das PEP kann diese Problemstellung verallgemeinert werden, sodass sich das in [HLS14a] vorgestellte Multiple-Choice Combinatorial Optimization Problem (MC-COP) ergibt:

$$\begin{aligned} \min \quad & \left\| \zeta - \sum_{i=1}^{|A|} \sum_{j=1}^{|S_i|} (\theta_{ij} \cdot x_{ij}) \right\|_1 \\ \text{mit} \quad & \sum_{j=1}^{|S_i|} x_{ij} = 1, \quad i = 1 \dots |A| \\ \text{und} \quad & x_{ij} \in \{0, 1\}, \quad i = 1 \dots |A|, \quad j = 1 \dots |S_i| \end{aligned} \tag{3.1}$$

Hier ist es im Gegensatz zum MC-SSP erlaubt, das Zielelement wertmäßig zu überschreiten. Das Optimierungsziel besteht also lediglich darin, das Zielelement (das Wirkleistungsprodukt) so weit wie möglich anzunähern. In dieser Formulierung wurde exemplarisch die Summennorm (1-Norm) als Metrik zur Bewertung der Ähnlichkeit von Wirkleistungsprodukt und Verbundfahrplan verwendet. Diese liefert ein Bewertungsmaß für die kumulierten Abstände über alle Planungsintervalle und soll minimiert werden. Es sind jedoch auch alternative Metriken denkbar. Sollen beispielsweise extreme Abweichungen in einzelnen Planungsintervallen bevorzugt vermieden werden, so würde sich etwa die Euklidische Norm (2-Norm) anbieten, da die Abstände hier quadratisch in das Bewertungsmaß einfließen. Eine Betrachtung weiterer Metriken in diesem Kontext liefert [Bor01, Abs. 6.2]. Die Werte x_{ij} sind binäre Entscheidungsvariablen, die angeben ob ein bestimmtes Element θ_{ij} ausgewählt ist ($x_{ij} = 1$) oder nicht ($x_{ij} = 0$). Die Nebenbedingungen stellen sicher, dass für eine gültige Lösung aus jeder Klasse (d. h. jedem Suchraum) exakt ein Element ausgewählt wird. Neben der Vernachlässigung lokaler Präferenzen stellt das obige MC-COP aber zudem ein Modell aus der Sicht eines zentralen Optimierers dar. Es soll an dieser Stelle nur zur Veranschaulichung des kombinatorischen Optimierungsproblems dienen und wird daher in den folgenden Abschnitten nicht näher

ausgearbeitet.

Zu erwähnen bleibt, dass das MC-SSP in die Klasse der schwach NP-vollständigen Probleme fällt und nur unter bestimmten Umständen (zentrales Lösungsverfahren mit globalem Wissen, sowie alle Wertebereiche durch bekannte Konstanten beschränkt) mit polynomielltem Berechnungsaufwand lösbar ist [Pis99]. Da das MC-COP (und damit auch das PEP) eine Verallgemeinerung des MC-SSP ist, fällt auch dieses in die Klasse der schwach NP-vollständigen Probleme (vgl. [Abschnitt 2.1](#)).

3.2 AGENTEN

Insgesamt spezifizieren die Definitionen der vorangegangenen Abschnitte das Eingabe-/Ausgabeverhalten des zu erstellenden Systems. Da die beschriebenen Akteure des Systems dezentrale Entitäten mit individuellen Fähigkeiten und Zielen darstellen, welche ein gemeinsames Ziel zu erreichen versuchen, handelt es sich um ein Multiagentensystem (MAS, vgl. [Abschnitt 2.2](#)). Für einen einzelnen Agenten lässt sich das verfolgte Ziel wie folgt formulieren:

Sobald ein Agent das vom Verbund zu realisierende Wirkleistungsprodukt kennt, besteht sein Ziel darin, einen Fahrplan derart auszuwählen und der ihm zugeordneten Energieanlage zuzuweisen, sodass der vom Verbund insgesamt gebildete Einsatzplan den vorgegebenen Wirkleistungsverlauf so gut wie möglich realisiert, dabei aber zugleich die lokalen Präferenzen der Anlage so gut wie möglich erfüllt.

Diese Zielformulierung entspricht dem lokalen Anteil eines Agenten am PEP aus [Definition 3.1.13](#) und beinhaltet an dieser Stelle ebenfalls noch bewusst unspezifisch die gleichzeitige Optimierung der globalen Gütefunktion sowie der lokalen Erfüllungsfunktion. Die Aufgabe der nachfolgenden Abschnitte wird es sein, ein konkretes Verfahren zu entwerfen, welches diese beiden potenziell konfliktären Teilziele auf verteilte Weise lösen kann.

Da zumindest die Optimierung der globalen Gütefunktion von allen Agenten gemeinsam beeinflusst wird, müssen sich die Agenten des MAS mittels Interaktionen untereinander abstimmen und sich koordinieren. Die hierbei bestehende Interaktionssituation (engl. *interaction situation*) kann nach Ferber [Fer99] als eine Staffellung aus Kollaboration und Wettbewerb interpretiert werden.

3.2.1 Kollaboration

Ein gemeinsames Teilziel aller Agenten ist die kombinatorische Optimierung bzgl. des Wirkleistungsproduktes. Auf dieser Ebene sind die Ziele der Agenten kompatibel zueinander. Es stehen genügend Ressourcen (in Form von Suchräumen) für die Agenten zur Verfügung, um dieses gemeinsame Teilziel zu erfüllen. Die Agenten müssen dazu jedoch kollaborieren, da es keinen Agenten gibt, der das Teilziel alleine erreichen kann. Ferber bezeichnet diese Situation als einfache Kollaboration (engl. *simple collaboration*), in welcher die Agenten einander gleichgültig gegenüberstehen (engl. *indifference*). Letzteres bedeutet insbesondere, dass die Agenten in Bezug auf die Erfüllung des gemeinsamen Teilziels weder im Wettbewerb zueinander stehen, noch altruistisch gesinnt sein müssen (in dem Sinne, dass einzelne Agenten Nachteile zugunsten der Zielerfüllung hinnehmen; dies wird von Ferber mit dem Begriff *cooperation* bezeichnet).

3.2.2 Wettbewerb

Zusätzlich verfolgen die Agenten aber sich voneinander unterscheidende Teilziele bzgl. der maximalen Erfüllung lokaler Präferenzen. Diese können, bei gleichzeitiger Betrachtung des gemeinsamen Ziels, inkompatibel zueinander sein. Eine exemplarische Situation dazu liefert das in [Beispiel 3.2.1](#) (Seite 49) beschriebene Szenario. Die Erfüllung des gemeinsamen Ziels steht in solch einer Situation diametral zur Maximierung der individuellen lokalen Gütwerte. Ein Agent kann nun entweder seine lokalen Gütwerte priorisieren und sich damit hinsichtlich dieser egoistisch verhalten, oder das gemeinsame Ziel favorisieren und sich trotz der damit verbundenen lokalen Nachteile altruistisch verhalten (was durch entsprechende Anreize aber ebenfalls positiv vergütet werden kann, siehe hierzu [Abschnitt 4.4.2](#)). Auch Mischformen sind denkbar, in denen ein Kompromiss aus lokaler und globaler Zielerfüllung gefunden wird.

Im betrachteten Anwendungsfall soll es im Sinne des Autonomieerhalts jedem Agenten frei stehen, die lokalen Präferenzen gegenüber dem globalen gemeinsamen Ziel in eigenem Ermessen zu priorisieren. An den zu entwickelnden Selbstorganisationsmechanismus wird daher die Anforderung gestellt, auch unter diesen Umständen

Beispiel 3.2.1

Gegeben sei ein Wirkleistungsprodukt ζ und ein Verbund $A = \{a_0, a_1\}$, für den die vier Verbundfahrpläne

$$\Theta_w = \theta_{0,0} + \theta_{1,0}$$

$$\Theta_x = \theta_{0,0} + \theta_{1,1}$$

$$\Theta_y = \theta_{0,1} + \theta_{1,0}$$

$$\Theta_z = \theta_{0,1} + \theta_{1,1}$$

realisierbar sind. Die Agenten a_0, a_1 haben also je zwei mögliche Fahrpläne $\theta_{\cdot,0}, \theta_{\cdot,1}$ anzubieten. Es gelte nun

$$f(\zeta, \Theta_w) = f(\zeta, \Theta_z) < f(\zeta, \Theta_x) = f(\zeta, \Theta_y)$$

und zugleich

$$\forall i \in \{0, 1\} : g_i(\theta_{i,0}) > g_i(\theta_{i,1}).$$

Um ζ optimal anzunähern, müsste also genau einer der beiden Agenten eine schlechtere lokale Güte in Kauf nehmen.

das PEP lösen zu können. Hierauf wird vom Standpunkt der mehrkriteriellen Optimierung aus detailliert im „4. Designschritt“ der Ausgestaltung der Verhaltensregeln der Agenten in [Abschnitt 3.6.4](#) eingegangen.

3.2.3 Fähigkeiten

Die grundlegende Funktionsweise eines Agenten kann nach [[Wei13](#), Kap. 2] durch die Schritte Wahrnehmung (engl. *perception*), Entscheidung (engl. *decision*) und Aktion (engl. *action*) beschrieben werden. Im vorliegenden System ergibt sich daraus das folgende Gerüst der Fähigkeiten eines Agenten.

Wahrnehmung Agenten können miteinander kommunizieren. Von anderen Agenten erhaltene Nachrichten können als lokales Wissen gespeichert werden.

Entscheidung Wie in [Abschnitt 3.1](#) beschrieben, repräsentiert jeder Agent eine Energieanlage, deren mögliche Fahrpläne für den betrachteten Planungshorizont in Form eines Suchraummodells vorliegen. Dieses Suchraummodell ist dem jeweiligen Agenten bekannt, sodass dieser darin nach bestimmten realisierbaren Fahrplänen für die ihm zugeordnete Anlage suchen kann. Zudem kennt jeder Agent $a_i \in A$ die Berechnungsvorschriften der Bewertungsfunktionen f und g_i ([Definition 3.1.11](#) und [3.1.12](#)), mit deren Hilfe Fahrpläne ausgewählt werden können.

Aktion Sofern ein Agent die Bezeichner anderer Verbundteilnehmer kennt, kann er Nachrichten an diese schicken.

Für den betrachteten Anwendungsfall wird vorausgesetzt, dass Agenten *vertrauenswürdig* sind, d. h. sie verfolgen keine weiteren als die bereits beschriebenen Ziele oder üben Handlungen aus, welche dem Verbund schaden. Eine detailliertere Betrachtung dieser Anforderung kann erst nach dem erfolgten Entwurf des Einsatzplanungsverfahrens erfolgen und ist in [Abschnitt 4.4](#) zu finden.

3.3 KOMMUNIKATION

Im vorigen Abschnitt wurde beschrieben, dass die Lösung des PEP durch ein Multi-agentensystem realisiert wird, in welchem die Agenten mittels Interaktionen eine Koordination herbeiführen. Diese Interaktionen geschehen über den Austausch von Nachrichten über ein Kommunikationsnetz. Insbesondere für die Umsetzbarkeit des Systems in einer Produktivumgebung ist die genaue Betrachtung des Kommunikationsnetzes von signifikanter Bedeutung. So wird bspw. geschätzt, dass bei der Realisierung bestehender Ansätze für virtuelle Kraftwerke bis zu 10% der Betriebskosten durch die Kommunikation verursacht werden könnten [Peh+06, S. 15]. Insbesondere der Aspekt der Nicht-Zuverlässigkeit wird in Forschungsarbeiten zu verteilten Systemen zudem oftmals ignoriert [Lyn96, S. 459]. Daher wird in der vorliegenden Arbeit eine Abbildung realer, kostengünstiger Kommunikationsnetze angestrebt. Im Folgenden wird das dazu erforderliche Kommunikationsnetz definiert.

Definition 3.3.1 (Kommunikationsnetz)

Sei A ein Verbund. Dann repräsentiert der Digraph $C = (A, A \times A)$ das dem Verbund zugrundeliegende *Kommunikationsnetz*. Dieses ermöglicht es prinzipiell beliebigen Verbundteilnehmern, miteinander zu kommunizieren. Es besitzt folgende Eigenschaften:

1. Kommunikation erfolgt durch *Nachrichten* der Form (Ziel, Nutzdaten). Als Zielangabe genügt es, den Bezeichner des Empfängers einzutragen.
2. Die Übertragung erfolgt *asynchron*, d. h. Agenten können Nachrichten zu beliebigen Zeitpunkten absenden.
3. Die Übertragung erfolgt *verbindungslos*, d. h. für verschickte Nachrichten erhält der Sender keine Empfangsbestätigung.
4. Die Übertragung erfolgt *nicht-blockierend*, d. h. nach dem Absenden einer Nachricht kann der Sender direkt mit seinen Aktionen fortfahren.
5. Das Kommunikationsnetz kann die Zustellung von Nachrichten individuell (aber endlich) *verzögern*. Daraus resultiert, dass sequenziell versendete Nachrichten nicht notwendigerweise in der gleichen Reihenfolge eintreffen, in der sie abgeschickt wurden.

Das Kommunikationsnetz modelliert somit eine verbindungslose Punkt-zu-Punkt Kommunikation (engl. *unicast* oder *point-to-point*). Aus den Eigenschaften *nicht-blockierend* und *asynchron* folgt insbesondere, dass Nachrichten jederzeit zugestellt werden können müssen (wie bspw. bei der Übertragung per *User Datagram Protocol* [Pos80] im Internet). Dies kann etwa über eine Kombination aus Pufferspeicher auf Netzwerkebene (für Nachrichten, die zeitgleich bei einem Empfänger eintreffen) und einem stets empfangsbereiten Pufferspeicher im Empfängeragenten (in welchem Nachrichten so lange vorgehalten werden, bis der Agent sie bearbeitet) realisiert werden, vgl. [Mat89, Abs. 1.3]. Für den nachfolgenden Designprozess des Selbstorganisationsmechanismus ist also die Berücksichtigung der durch das nicht-zuverlässige Kommunikationsnetz induzierten Schwierigkeiten von besonderem Interesse.

3.4 PLANUNG DES SYSTEMDESIGNS

Nach der Aufstellung der Rahmenbedingungen soll nun das eigentliche Verfahren für die selbstorganisierte Einsatzplanung entwickelt werden. Wie eingangs bereits erwähnt, wird dazu der Designprozess für selbstorganisierende Systeme nach Prehofer und Bettstetter eingesetzt. Dieser setzt sich aus vier ineinander greifenden Paradigmen mit jeweils unterschiedlichen Zielen zusammen (vgl. [Abbildung 3.1](#) auf S. 37). Gemäß [PB05] soll zunächst der Kern des Selbstorganisationsmechanismus anhand der Paradigmen #1 (*Local behavior rules*) und #2 (*Implicit coordination*) ausgestaltet werden. Beide Paradigmen haben das Ziel, die gewünschte globale Systemfunktion so auf die individuellen Systembestandteile abzubilden, dass ein selbstorganisierender Prozess sichtbar wird, der ausgehend von lokalen Interaktionen die globale Systemfunktion produziert. Dabei entspricht Paradigma #1 einem *Top-Down* Vorgehen, bei dem durch die Zerlegung der globalen Systemfunktion auf lokal abgrenzbare Bestandteile das individuelle Verhalten der Systembestandteile definiert wird. Paradigma #2 hingegen eignet sich dann als Herangehensweise, wenn bereits lokales Verhalten der Systembestandteile gegeben ist, welches durch die Auflösung von Konflikten dahingehend modifiziert wird, dass es schließlich die globale Zielfunktion abbildet. Welches der beiden Paradigmen als initiale Herangehensweise gewählt wird, ist dem Systemdesigner überlassen und muss individuell entschieden werden. Im Anschluss werden mit Paradigma #3 (*Monitoring algorithm*) die

adaptiven Eigenschaften des Systems ausgestaltet. Die in Paradigma #4 (*Reduced state*) betrachtete Reduktion langlebiger Zustandsinformationen zur Begünstigung des Selbstorganisationsprozesses lässt sich unmittelbar aus den in den Paradigmen #1 und #2 aufgestellten Eigenschaften und Regeln ableiten. Der Designprozess endet mit der finalen Definition des angestrebten Systemverhaltens, welches sich aus den Ergebnissen der zuvor beschriebenen Vorgänge zusammensetzt: Lokale Verhaltensregeln, ausgetauschte Nachrichten, vorgehaltene Zustandsinformationen sowie Kontrollmechanismen zur Adaptivität.

Für die vorliegende Problemstellung wird Paradigma #1 als maßgebliche Herangehensweise gewählt, da kein initiales lokales Verhalten der Verbundteilnehmer vorgegeben ist, und zudem die Modellierung von Konflikten und Inkonsistenzen in dem gegebenen Optimierungsproblem als schwierig erachtet wird (da gemäß der Formalisierung im vorangegangenen Abschnitt kein *Constraint Satisfaction Problem*, sondern ein reines *Constraint Optimization Problem* vorliegt, vgl. [Abschnitt 2.1](#)). Im Folgenden wird daher in [Abschnitt 3.5](#) zunächst als Voraussetzung des Designprozesses die gewünschte globale Systemfunktion aufgestellt. Anschließend folgt in [Abschnitt 3.6](#) die iterative Ausgestaltung des lokalen Verhaltens der Verbundteilnehmer anhand Paradigma #1 des Designprozesses. Hierzu sind im vorliegenden Fall vier Iterationen erforderlich (nachfolgend als Designschritte bezeichnet), in welchen das Verhalten der Verbundteilnehmer Schritt für Schritt verfeinert wird. In [Abschnitt 3.7](#) wird das definierte Verhalten mit Paradigma #2 auf Konflikte und Inkonsistenzen geprüft. Auf die Ausgestaltung der adaptiven Eigenschaften des Systems wird dann mit Paradigma #3 in [Abschnitt 3.8](#) eingegangen. Schließlich folgt mit Paradigma #4 in [Abschnitt 3.9](#) die aus den ersten beiden Paradigmen abgeleitete Reduktion von langlebigen Zustandsinformationen.

3.5 GLOBALE SYSTEMFUNKTION

Die globale Systemfunktion wurde detailliert in [Abschnitt 3.1](#) beschrieben und besteht in der Lösung des PEP aus [Definition 3.1.13](#).

3.6 PARADIGMA #1: LOKALE VERHALTENSREGELN

Um passende lokale Verhaltensregeln für die Erfüllung der globalen Systemfunktion zu finden, sollte diese nach Prehofer und Bettstetter auf ihre lokal handhabbaren Bestandteile untersucht werden. Die Autoren schlagen hierzu unter anderem das teile-und-herrsche Prinzip vor (engl. *divide-and-conquer concept*, [PB05, S. 81]), nach welchem die einzelnen Agenten Informationen lokal sammeln und bearbeiten, und ihre Erkenntnisse dann mit anderen Agenten austauschen. Für die vorliegende Problemstellung scheint dieser Ansatz besonders geeignet zu sein, da die Verbundteilnehmer bereits inhärent lokales Wissen im Form ihres Suchraumes besitzen, welches auf effiziente Weise über alle Verbundteilnehmer aggregiert werden muss, um einen optimalen gemeinsamen Verbundfahrplan zu erzeugen. Vom Standpunkt der Metaheuristiken aus betrachtet entspricht dieses Vorgehen dem *Cooperative Algorithmic-Level Parallel Model* [Tal09, Abs. 6.1.1.2].

Im Folgenden werden nun die lokalen Verhaltensregeln für die Verbundteilnehmer iterativ mittels jeweils aufeinander aufbauender Designschritte ausgestaltet. Hierbei wird in den ersten drei Schritten zunächst nur das gemeinsame Teilziel der Verbundteilnehmer (die Maximierung der globalen Güte) als einzelnes Kriterium für das zu lösende Optimierungsproblem betrachtet. In Designschritt 4 (Abschnitt 3.6.4) wird das Verhalten dann auf den mehrkriteriellen Fall ausgeweitet und um die Erfüllung lokaler Präferenzen ergänzt.

3.6.1 Designschritt 1: Iterative Annäherung

Als grundlegende Herangehensweise wird eine iterative Annäherung an das Wirkleistungsprodukt gewählt, in welcher die Verbundteilnehmer ihren eigenen Fahrplan dahingehend wählen, sodass dieser unter Berücksichtigung ihrer jeweils aktuellen Informationen bzgl. anderer Verbundteilnehmer das Wirkleistungsprodukt maximal annähert. Ein Verfahren mit vergleichbarem Ansatz wurde von Li, Poulton und James im Jahr 2008 zum Patent angemeldet [LPJ08]. Dieses stützt sich jedoch auf einen zentralen Informationsspeicher, dort als *Stigspace* bezeichnet, in welchem die Agenten ihre jeweiligen Fahrplanwechsel veröffentlichen (siehe [LPJ10] sowie Abschnitt 1.3.2 für nähere Details). Das zugrundeliegende Kommunikationsprinzip wird auch als *Blackboard* bezeichnet, siehe hierzu bspw. [Fer99]. In der vorliegen-

Beispiel 3.6.1

Ein Fisch in einem Schwarm hat das individuelle Ziel, sich innerhalb des Schwarms zu befinden, und sich dabei in die gleiche Richtung wie der Schwarm zu bewegen sowie einen wohldefinierten Abstand zu anderen Schwarmteilnehmern zu halten. Ein Fisch reagiert daher auf Positions- und Richtungsänderungen von solchen anderen Fischen im Schwarm, welche sich innerhalb eines fest definierten Wahrnehmungsradius des Fisches befinden, da diese Aktionen unmittelbare Auswirkungen auf sein individuelles Ziel haben. Unterschreitet ein anderer Schwarmteilnehmer den minimal einzuhaltenden Abstand, oder entfernt sich der Fisch zu weit von der Mehrzahl der ihn umgebenden Schwarmteilnehmer, bewegt sich der Fisch jeweils in die entgegengesetzte Richtung. Zugleich versucht er, in die gleiche Richtung wie die ihn umgebenden Fische zu schwimmen (vgl. [Rey87]).

den Arbeit sollen die bereits diskutierten Nachteile eines solchen zentralisierten Kommunikationsmodells vermieden, und eine vollständig verteilte Lösung gefunden werden.

Als Inspirationsquelle für ein solches verteiltes Verfahren dient das *Schwarmverhalten*, genauer die reaktive Anpassung der Bewegungsrichtung eines Schwarmteilnehmers an die ihn umgebenden Individuen [Cam+03]. Im Bereich der *Distributed Constraint Optimization* wird dies als iterative Annäherung nach besten Möglichkeiten (engl. *iterative approximate best-response algorithm*) bezeichnet [Cha+11]. In der Theorie der Multiagentensysteme ist diese Vorgehensweise als reaktive Koordination (engl. *reactive coordination* oder *eco-problem solving*) bekannt [Fer99]. Das Prinzip ist in [Beispiel 3.6.1](#) anhand eines Fischeschwarms verdeutlicht. Der Freiheitsgrad der einzelnen Schwarmteilnehmer in diesem Beispiel besteht in der Möglichkeit, ihre eigene Richtung und Geschwindigkeit zu modifizieren. Übertragen auf das vorliegende System, soll auch ein Verbundteilnehmer im Rahmen seiner Freiheitsgrade (d. h. seiner Fahrplanauswahl) stets auf die Aktionen anderer Verbundteilnehmer reagieren. So soll er seine Fahrplanauswahl immer dann revidieren, wenn er aktualisierte Informationen über die Fahrplanauswahl anderer Verbundteilnehmer erhält, und ein anderer Fahrplan als der derzeit gewählte ggf. eine bessere globale Güte gemäß [Definition 3.1.11](#) erreichen würde. Genauer betrachtet müsste ein Verbundteilnehmer hier auf Änderungen der Fahrplanauswahl derjenigen anderen Verbundteilnehmer

reagieren, welche eine unmittelbare Auswirkung auf seine eigenen individuellen Ziele haben. Diese Teilmenge der observierten anderen Verbundteilnehmer würde so die lokale Umgebung des Verbundteilnehmers darstellen (und damit seinen Wahrnehmungsradius, wie im obigen Beispiel). Daraus ergibt sich das erste konzeptuelle Gerüst des lokalen Verhaltens von Verbundteilnehmern, bestehend aus den in [Abschnitt 3.2](#) eingeführten Teilen *Wahrnehmung*, *Entscheidung* und *Aktion*.

Lokale Verhaltensregel, Version 1

Erhält ein Verbundteilnehmer a_i eine Nachricht bzgl. einer Fahrplanauswahl eines anderen Verbundteilnehmers ($\rightarrow$ Wahrnehmung), und beeinflusst diese Fahrplanauswahl das lokale Ziel von a_i (wie in [Abschnitt 3.2](#) beschrieben), so soll a_i auf diese aktualisierten Informationen reagieren und die eigene Fahrplanauswahl dahingehend modifizieren, sodass das lokale Ziel maximiert wird ($\rightarrow$ Entscheidung). Im Anschluss soll a_i die eigene (neue) Fahrplanauswahl denjenigen anderen Verbundteilnehmern mitteilen, deren lokales Ziel davon betroffen sein könnte ($\rightarrow$ Aktion).

Wie bereits in [Abschnitt 2.1](#) erläutert, bilden die Abhängigkeiten des betrachteten verteilten Optimierungsproblems jedoch einen vollvernetzten Constraint-Graphen, sodass die durch die Abhängigkeiten induzierte Definition einer Nachbarschaftsrelation ausschließlich zu globalen Nachbarschaften, und damit zu globalen Wahrnehmungsradien führen würde. Bei direkter Umsetzung in einem vollständig verteilten System (d. h. ohne zentralen Informationsspeicher) würde die Kommunikation hier daher zu einem *Broadcasting* degenerieren: Jeder Verbundteilnehmer würde mit jeder Änderung seines Fahrplans eine Nachricht an alle anderen Verbundteilnehmer senden. Da Fahrplanänderungen eben durch solche Nachrichten verursacht werden, was wiederum zum Versenden von Nachrichten an alle anderen Verbundteilnehmer führt, würde der Kommunikationsaufwand somit quadratisch mit der Anzahl der Verbundteilnehmer wachsen.

3.6.2 Designschritt 2: Overlay-Netz

Um den bei Verwendung der *Broadcasting* Kommunikation entstehenden quadratischen Kommunikationsaufwand zu reduzieren, wird im vorliegenden System ein heuristischer Ansatz gewählt, der die Menge der Nachbarn begrenzt, auf deren Fahr-

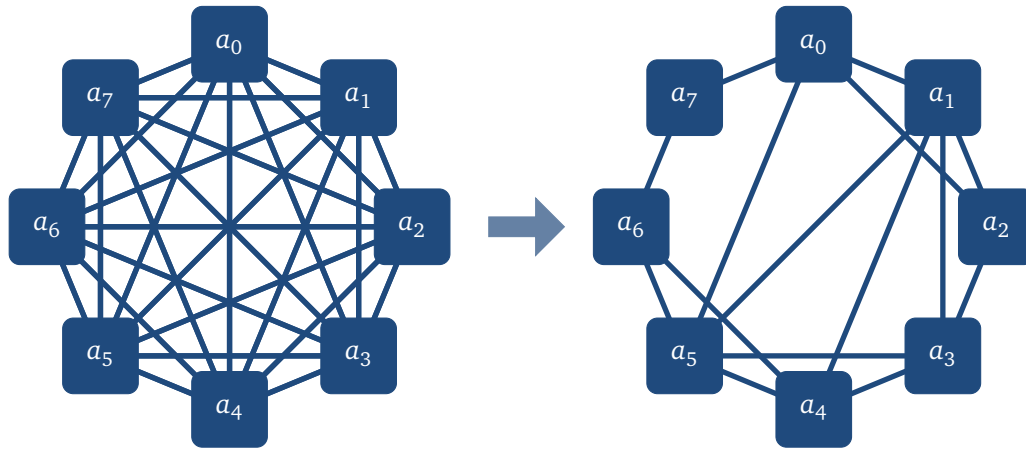


Abbildung 3.3: Exemplarisches Overlay-Netz (rechts) für ein vollvermaschtes Kommunikationsnetz (links) mit acht Agenten.

planauswahl ein Verbundteilnehmer unmittelbar reagiert. Dazu werden die Nachbarschaftsbeziehungen der Verbundteilnehmer artifizuell mittels eines *Overlay-Netzes* auf dem nach [Definition 3.3.1](#) vollvermaschten Kommunikationsnetz gebildet. Gemäß [\[DO03\]](#), referenziert in [\[Van06\]](#), ist ein Overlay-Netz eine logische (d. h. virtuelle) Topologie, die über eine bestehende (z. B. physikalische) Topologie gelegt wird. In einem solchen Netz kommunizieren ausschließlich solche Agenten direkt miteinander, welche in der virtuellen Topologie benachbart sind. Der eigentliche Nachrichtentransfer erfolgt dabei weiterhin über das ursprüngliche Netzwerk. [Abbildung 3.3](#) zeigt exemplarisch ein mögliches Overlay-Netz für einen aus acht Teilnehmern bestehenden Verbund $A = \{a_0, \dots, a_7\}$.

Definition 3.6.1 (Schwarmbildendes Overlay-Netz)

Sei $C = (A, A \times A)$ das Kommunikationsnetz für einen Verbund A . Dann ist ein Teilgraph $O = (A, E)$ mit $E \subseteq A \times A$ ein *Overlay-Netz* auf C .

Ist O zudem zusammenhängend, irreflexiv und symmetrisch, sodass gilt

$$\forall a_i, a_j \in A : W = (a_i, \dots, a_j) \text{ ist ein Weg in } O,$$

$$\forall (a_i, a_j) \in E : a_i \neq a_j,$$

$$\forall (a_i, a_j) \in E : (a_j, a_i) \in E,$$

dann wird O als *schwarmbildend* bezeichnet.

Auf die konkrete Bildung des Overlay-Netzes wird in [Kapitel 4](#) und [Abschnitt 7.3.1](#) näher eingegangen. Für die folgenden Abschnitte sei lediglich angenommen, dass ein solches schwarmbildendes Overlay-Netz O definiert sei (im Folgenden auch verkürzt als „Netz“ bezeichnet), welches jedem Verbundteilnehmer eine Nachbarschaft zuordnet.

Definition 3.6.2 (Nachbarschaft)

Sei $O = (A, E)$ ein schwarmbildendes Overlay-Netz. Dann ist für jeden Verbundteilnehmer $a_i \in A$ mit $N_i = \{a_j \mid (a_i, a_j) \in E\}$ eine *Nachbarschaft* von anderen Verbundteilnehmern gegeben, mit denen a_i kommunizieren kann.

Die weiter oben beschriebene [Lokale Verhaltensregel, Version 1](#) wird nun mit den durch die Nachbarschaften eingeschränkten Wahrnehmungsradien von Verbundteilnehmern ergänzt.

Lokale Verhaltensregel, Version 2

Jeder Verbundteilnehmer verwaltet einen lokalen Wissensspeicher, in welchem er von anderen Verbundteilnehmern empfangene Informationen abspeichert. Erhält ein Verbundteilnehmer a_i von einem seiner Nachbarn $a_j \in N_i$ eine Nachricht bzgl. einer geänderten Fahrplanauswahl θ_j , so wird zunächst der lokale Wissensspeicher von a_i mit dieser neuen Information aktualisiert ($\rightarrow$ Wahrnehmung). Nun prüft a_i , ob unter Berücksichtigung dieser neuen Informationslage eine andere als die derzeitige eigene Fahrplanauswahl eine höhere Erfüllung des eigenen Zieles (wie in [Abschnitt 3.2](#) beschrieben) bewirkt. Ist dies der Fall, so soll a_i die eigene Fahrplanauswahl dahingehend modifizieren, sodass das Ziel maximiert wird ($\rightarrow$ Entscheidung). Anschließend teilt a_i die neue Fahrplanauswahl jedem seiner Nachbarn mit ($\rightarrow$ Aktion).

Ein wichtiger Aspekt hierbei ist, dass sich Nachbarschaften aufgrund der schwarmbildenden Eigenschaft „überlappen“, d. h. die Menge der Nachbarschaften über A stellt keine Partitionierung von A dar. Dies hat zur Folge, dass die Reaktionen von Verbundteilnehmern auf Ereignisse in ihrer Umgebung wellenförmig durch das Netz kaskadieren, und analog zu [Beispiel 3.6.1](#) ein Selbstorganisationsprozess sichtbar wird. Dies entspricht dem Konzept der positiven Rückkopplung aus [Abschnitt 2.3.1](#) und resultiert in einer chaotisch anmutenden, aber dennoch zielgerichteten Exploration des Lösungsraumes des Optimierungsproblems.

Im Vergleich zu einem Fischschwarm besteht hier jedoch der Unterschied, dass lokal isolierte (Re-)Aktionen nicht ausreichen, um das System zu einem Equilibrium zu führen. Denn während in einem Fischschwarm die lokalen Anpassungen der Schwarmteilnehmer an ihre unmittelbaren Umgebungsbedingungen genügen, um die Zielerreichung jedes einzelnen Schwarmteilnehmers zu maximieren, muss im vorliegenden System ein globaler *Konsens* gefunden werden.

3.6.3 Designschritt 3: Konsens

Um einen Konsens bzgl. der global optimalen Fahrplanauswahl jedes einzelnen Verbundteilnehmers zu erreichen, muss ein Mechanismus konstruiert werden, der eine über lokale Nachbarschaften hinausgehende Informationssynthese erlaubt. Das Ziel hierbei ist es, einem Verbundteilnehmer die Reaktion auf Fahrplanauswahlen von Verbundteilnehmern zu ermöglichen, die sich nicht in seiner unmittelbaren Nachbarschaft befinden. Außerdem sollte dieser Mechanismus dazu führen, dass das System *konvergiert*, also dass sich im Laufe des Optimierungsprozesses alle Verbundteilnehmer nach und nach auf eine gemeinsame Lösung einigen. Zudem sollte der resultierende Algorithmus die sog. *anytime* Eigenschaft besitzen, nach welcher unmittelbar nach dem Start des Verfahrens gültige Lösungen vorliegen, welche mit zusätzlich zur Verfügung stehender Laufzeit inkrementell verbessert werden [[Cha+11](#), S. 433]. Dabei darf die durch das Overlay-Netz im vorigen Designschritt realisierte Reduktion der Kommunikationskomplexität jedoch nicht beeinträchtigt werden, sodass Ansätze ausgeschlossen werden, die etwa auf dem Vergrößern von Nachbarschaften oder Ähnlichem basieren.

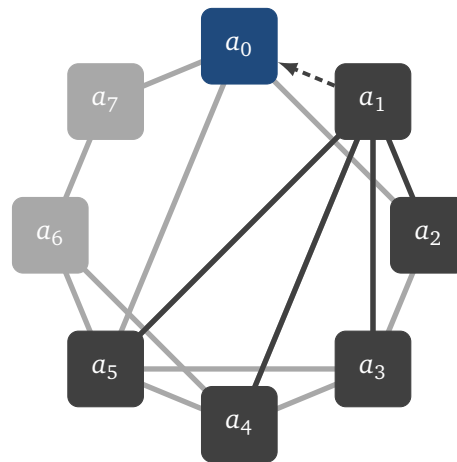


Abbildung 3.4: Über lokale Nachbarschaften hinausgehende Informationssynthese am Beispiel einer Nachricht von Verbundteilnehmer a_1 an a_0 , welche Informationen bzgl. a_1 sowie dessen Nachbarn $\{a_2, a_3, a_4, a_5\}$ enthält.

Informationssynthese

Eine über lokale Nachbarschaften hinausgehende Informationssynthese kann erreicht werden, indem die im vorigen Abschnitt identifizierte Überlappung von Nachbarschaften ausgenutzt wird. Die Idee ist es, dass ein Verbundteilnehmer in seiner Nachricht an seine Nachbarn bzgl. einer aktualisierten eigenen Fahrplanauswahl *zusätzlich* sein aktuelles Wissen über die ihm bekannten Fahrplanauswahlen anderer Verbundteilnehmer inkludiert. [Abbildung 3.4](#) veranschaulicht dies anhand des in obigem Abschnitt eingeführten Overlay-Netzes exemplarisch für einen Verbundteilnehmer a_0 . Empfängt a_0 wie dargestellt beispielsweise eine Nachricht von a_1 , so würde diese Nachricht Informationen zur aktuellen Fahrplanauswahl von a_1 sowie die a_1 derzeit vorliegenden Informationen zur Fahrplanauswahl anderer Verbundteilnehmer (hier $\{a_2, a_3, a_4, a_5\}$) beinhalten. Es ist offensichtlich, dass sich durch diesen Mechanismus mehrere redundante Wege im Overlay-Netz bilden, über die ein Verbundteilnehmer Informationen bzgl. anderer Verbundteilnehmer erhält. Insbesondere kann dabei durch die vorausgesetzten Eigenschaften des Kommunikationsnetzes (vgl. [Definition 3.3.1](#)) nicht mehr davon ausgegangen werden, dass die bei einem Verbundteilnehmer eintreffenden Nachrichten den jeweils aktuellsten

Informationsstand beinhalten. Daher ist ein Entscheidungskriterium erforderlich, welches es einem Verbundteilnehmer für jede erhaltene Information ermöglicht, deren Aktualität in Bezug zu bereits gespeichertem Wissen zu bewerten. Dies kann mit einer einfachen Ergänzung erreicht werden: Jeder Verbundteilnehmer verwaltet eine Zählvariable, welche bei jedem Fahrplanwechsel inkrementiert und der vom Verbundteilnehmer verschickten Information bzgl. seiner eigenen Fahrplanauswahl beigefügt wird. Mit dieser Ergänzung lässt sich nun die vollständige Definition einer Fahrplanauswahl aufstellen.

Definition 3.6.3 (Fahrplanauswahl)

Eine *Fahrplanauswahl* eines Verbundteilnehmers a_i ist ein Tupel $\omega_i = (a_i, \theta_i, \lambda_i)$, wobei a_i gemäß Definition 3.1.3 den Bezeichner des Verbundteilnehmers und θ_i gemäß Definition 3.1.8 einen von a_i produzierbaren Fahrplan darstellen. Die dritte Komponente λ_i beinhaltet den zum Zeitpunkt der Erstellung von ω_i momentanen Wert einer Zählvariablen, welche von a_i mit jedem Fahrplanwechsel inkrementiert wird. Auf diese Weise können alle von a_i im Laufe des Prozesses erzeugten Fahrplanauswahlen ω_i durch ihre jeweiligen λ_i Werte in eine eindeutige zeitliche Reihenfolge gebracht werden.

Durch die soeben eingeführte Informationssynthese beinhalten ausgetauschte Nachrichten üblicherweise die Fahrplanauswahlen mehrerer Verbundteilnehmer. Es ist daher zweckmäßig, diese unter einem Oberbegriff zusammen zu fassen.

Definition 3.6.4 (Konfiguration, Schlüsselmenge, vollständige Konfiguration)

Sei A ein Verbund. Eine *Konfiguration* ist eine Menge $\Omega = \{\omega_i, \omega_j, \dots\}$ von Fahrplanauswahlen, für die gilt

$$\begin{aligned} \forall \omega_i, \omega_j \in \Omega : i &\neq j, \\ \forall \omega_i, \omega_j \in \Omega : a_i, a_j &\in A. \end{aligned}$$

Die Menge $K_\Omega = \{a_i \mid \omega_i \in \Omega\} \subseteq A$ beinhaltet die Bezeichner aller Verbundteilnehmer, deren Fahrplanauswahlen in Ω enthalten sind und wird als *Schlüsselmenge* der Konfiguration Ω bezeichnet.

Eine *vollständige Konfiguration* ist dann eine Konfiguration Ω mit $K_\Omega = A$.

Eine Konfiguration beinhaltet maximal eine Fahrplanauswahl für jeden Verbundteilnehmer. Insbesondere ist es zulässig, dass eine Konfiguration weniger als $|A|$ Elemente enthält, und somit *unvollständig* ist. Der Begriff der Konfiguration erlaubt es nun, den von einem Verbundteilnehmer wahrgenommenen Systemzustand formal zu beschreiben.

Definition 3.6.5 (Wahrgenommener Systemzustand)

Jeder Verbundteilnehmer a_i verwaltet in seinem Wissensspeicher eine Konfiguration Ω_i . Sobald mittels Nachrichtenaustausch Fahrplanauswahlen $\{\omega_j, \dots\}$ anderer Verbundteilnehmer empfangen werden, aktualisiert a_i die Konfiguration Ω_i entsprechend. Dabei wird eine Fahrplanauswahl ω_j eines Verbundteilnehmers a_j in Ω_i übernommen, wenn eine der beiden folgenden Bedingungen erfüllt ist:

1. Die Konfiguration Ω_i enthält noch keine Fahrplanauswahl des Verbundteilnehmers a_j , sodass $\forall \omega_z \in \Omega_i : z \neq j$.
2. Die Konfiguration Ω_i enthält bereits eine Fahrplanauswahl des Verbundteilnehmers a_j , welche aber älter als die soeben empfangene Fahrplanauswahl $\omega_j = (a_j, \theta_j, \lambda_j)$ ist, sodass $\exists \omega_z = (a_z, \theta_z, \lambda_z) \in \Omega_i : j = z \wedge \lambda_j > \lambda_z$. In diesem Fall wird ω_z gelöscht.

Da die Konfiguration Ω_i zu jedem Zeitpunkt das aktuellste bekannte Wissen von a_i bzgl. anderer Verbundteilnehmer enthält, wird Ω_i auch als *wahrgenommener Systemzustand* eines Verbundteilnehmers a_i bezeichnet.

An dieser Stelle ist hervorzuheben, dass jeder Verbundteilnehmer seine eigene unabhängige Zählvariable zur Markierung seiner sukzessiven Fahrplanauswahlen verwaltet, und durch diesen Mechanismus keineswegs ein globaler Zeitbegriff in das betrachtete verteilte System eingeführt wird.

Konvergenz

Das zweite Ziel des aktuellen Designschrittes, die Konvergenz des Systems, wird ebenfalls über eine Erweiterung der Kommunikation zwischen Verbundteilnehmern erreicht. Die Idee hierbei ist die Einführung von Lösungskandidaten für das gestellte Optimierungsproblem, welche von den Verbundteilnehmern individuell anhand

ihrer wahrgenommenen Systemzustände ermittelt und dann ihren jeweiligen Nachbarn zur Verfügung gestellt werden. Da sich aus einer vollständigen Konfiguration unmittelbar ein Einsatzplan für den Verbund ableiten lässt (vgl. [Definition 3.1.9](#)), können Lösungskandidaten durch Konfigurationen modelliert werden.

Definition 3.6.6 (Lösungskandidat)

Sei A ein Verbund. Ein *Lösungskandidat* für das A gestellte PEP ist ein Tupel $\gamma = (a_x, \Omega)$, wobei Ω eine vollständige Konfiguration ist und a_x denjenigen Verbundteilnehmer bezeichnet, welcher Ω initial ermittelt hat.

Jeder Verbundteilnehmer $a_i \in A$ verwaltet zu jedem Zeitpunkt maximal einen Lösungskandidaten in seinem Wissensspeicher. Dieser Lösungskandidat wird mit γ_i bezeichnet. Dabei ist zu beachten, dass a_i nicht notwendigerweise der Erzeuger von γ_i sein muss: $\gamma_i = (a_x, \Omega) \nRightarrow i = x$. Empfängt a_i mittels Nachrichtenaustausch einen Lösungskandidaten $\gamma_j = (a_y, \Omega')$ eines anderen Verbundteilnehmers a_j , so aktualisiert a_i seinen Lösungskandidaten $\gamma_i = (a_x, \Omega)$ wie folgt:

1. Wenn für die Schlüsselmengen gilt $K_\Omega \subset K_{\Omega'}$, dann ersetze γ_i durch γ_j .
2. Wenn für die Schlüsselmengen gilt $K_\Omega \not\subseteq K_{\Omega'}$, dann ergänze Ω mit den neuen Elementen $\{\omega_z \mid a_z \in K_{\Omega'} - K_\Omega\}$ und erzeuge daraus einen neuen Lösungskandidaten $\gamma_i' = (a_i, \Omega)$.
3. Wenn für die Schlüsselmengen gilt $K_\Omega = K_{\Omega'}$, die empfangene Konfiguration Ω' jedoch eine höhere Güte bzgl. der globalen Gütefunktion von a_i als Ω erreicht, dann ersetze γ_i durch γ_j .
4. Wenn für die Schlüsselmengen gilt $K_\Omega = K_{\Omega'}$, beide Konfigurationen die gleiche Güte bzgl. der globalen Gütefunktion von a_i erreichen, und für die Bezeichner a_x und a_y gilt $x > y$, dann ersetze γ_i durch γ_j .

In allen anderen Fällen wird γ_i unverändert beibehalten.

Die Aktualisierungsregeln 1 und 2 in obiger Beschreibung führen dazu, dass Lösungskandidaten zunächst soweit wie möglich vervollständigt werden. Die Regeln 3 und 4 bewirken schließlich die Konvergenz des Systems, da hierdurch eine Konsensbildung stattfindet: Die Nachbarn eines Verbundteilnehmers übernehmen einen per Nachricht übertragenen Lösungskandidaten in ihrem eigenen Wissensspeicher, sofern

dieser eine größere Zielerreichung für das PEP als ihr eigener derzeitiger Lösungskandidat erreichen würde. Auf diese Weise setzt sich die jeweils beste gefundene Lösung gegenüber schlechteren durch und wird nach und nach allen Verbundteilnehmern bekannt.

Bevor nun die nächste Version der lokalen Verhaltensregel eines Verbundteilnehmers aufgestellt wird, muss zunächst der Begriff des Wissensspeichers eines Verbundteilnehmers genauer betrachtet werden. Es wird zwischen *privatem* und *nicht-privatem* Wissen unterschieden. Ersteres beinhaltet das einem Verbundteilnehmer inhärent bei Prozessbeginn vorliegende Wissen, wie etwa das seiner Energieanlage zugeordnete Suchraummodell σ_i oder die lokalen Präferenzen, abgebildet durch die Erfüllungsfunktion g_i . Diese Informationen sind zudem *statisch*, d. h. sie können im Verlauf des Optimierungsprozesses nicht modifiziert werden. Der nicht-private Teil des Wissensspeichers eines Verbundteilnehmers hingegen beinhaltet dynamische Informationen, welche im Verlauf des Optimierungsprozesses kontinuierlich aktualisiert werden, und zudem mit anderen Verbundteilnehmern über Nachrichten geteilt werden können. Dieser Teil des Wissensspeichers wird als Arbeitsgedächtnis bezeichnet.

Definition 3.6.7 (Arbeitsgedächtnis)

Sei a_i ein Verbundteilnehmer. Das *Arbeitsgedächtnis* von a_i ist ein Tupel $\kappa_i = (\zeta, \Omega_i, \gamma_i)$ bestehend aus Wirkleistungsprodukt, wahrgenommenem Systemzustand und Lösungskandidat. Das Arbeitsgedächtnis ist veränderbar in dem Sinne, dass a_i die Inhalte von κ_i gemäß empfangener Nachrichten von anderen Verbundteilnehmern aktualisieren kann.

Zwischen Verbundteilnehmern ausgetauschte Nachrichten beziehen sich somit ausschließlich auf ihre Arbeitsgedächtnisse, und können daher wie folgt formalisiert werden.

Definition 3.6.8 (Nachricht)

Seien a_i und a_j Verbundteilnehmer. Eine *Nachricht* von a_i an a_j ist ein Tupel $\mu_i = (a_j, \kappa_i)$, wobei die erste Komponente des Tupels gemäß [Definition 3.3.1](#) den Empfänger der Nachricht bezeichnet, und die zweite Komponente die zu versendenden Nutzdaten. Im vorliegenden Fall veröffentlichen Verbundteilnehmer ihre vollständigen Arbeitsgedächtnisse.

In einer technischen Umsetzung des Systems kann der Nachrichtenaustausch dahingehend optimiert werden, dass statt vollständiger Arbeitsgedächtnisse jeweils nur aktualisierte Informationen versendet werden, um die Nachrichtengröße gering zu halten. Auf die korrekte Funktionsweise des Systems hat dies keinen Einfluss. Die beschriebene Modellierung zeigt zudem, dass der entwickelte Koordinationsansatz ausschließlich realisierbare Fahrpläne als individuelle Information bzgl. der Anlagen veröffentlicht – lokale Präferenzen oder sonstiges privates Wissen bleiben geschützt.

Insgesamt ergibt sich nach Betrachtung der Mechanismen für die Informationssynthese und für die Konvergenz die folgende erweiterte lokale Verhaltensregel.

Lokale Verhaltensregel, Version 3

Empfängt ein Verbundteilnehmer a_i eine Nachricht $\mu_j = (a_i, \kappa_j)$ mit $\kappa_j = (\zeta, \Omega_j, \gamma_j)$ von einem benachbarten Verbundteilnehmer a_j , so aktualisiert a_i zunächst sein Arbeitsgedächtnis κ_i mit den in κ_j enthaltenen Informationen:

- *Enthält das eigene Arbeitsgedächtnis κ_i noch kein Wirkleistungsprodukt ζ , speichere das Wirkleistungsprodukt aus μ_j in κ_i ab.*
- *Aktualisiere den eigenen wahrgenommenen Systemzustand Ω_i mit den in Ω_j enthaltenen Informationen gemäß den in [Definition 3.6.5](#) aufgestellten Regeln.*
- *Aktualisiere den eigenen Lösungskandidaten γ_i mit den in γ_j enthaltenen Informationen gemäß den in [Definition 3.6.6](#) aufgestellten Regeln.*

Hierbei kann es vorkommen, dass μ_j keine neuen Informationen beinhaltet, sodass sich das Arbeitsgedächtnis von a_i nicht ändert. In diesem Fall sind die folgenden Phasen überflüssig und werden nicht ausgeführt.

Wurde das Arbeitsgedächtnis im vorigen Schritt ($\rightarrow$ Wahrnehmung) modifiziert, optimiert der Verbundteilnehmer anschließend seine eigene Fahrplanauswahl ($\rightarrow$ Entscheidung). Sei für die folgende Betrachtung $\omega_i = (a_i, \theta_i, \lambda_i) \in \Omega_i$ die aktuelle Fahrplanauswahl von a_i , und $\tilde{\omega}_i = (a_i, \tilde{\theta}_i, \tilde{\lambda}_i) \in \gamma_i$ die im Lösungskandidaten enthaltene Fahrplanauswahl von a_i . Der Verbundteilnehmer ermittelt nun eine Fahr-

planauswahl $\omega_i' = (a_i, \theta_i', \lambda_i')$ mit $\lambda_i' = \lambda_i + 1$, welche in Bezug auf den aktuell wahrgenommenen Systemzustand $\Omega_i \setminus \{\omega_i\}$ gemäß [Definition 3.1.11](#) die höchste Güte erreichen würde. a_i vergleicht nun die Güte, die sich aus dem wahrgenommenen Systemzustand mit dieser neu ermittelten Fahrplanauswahl ω_i' ergibt, mit der Güte, die sich aus dem derzeitigen Lösungskandidaten γ_i ergibt. Ist die Güte von $(\Omega_i \setminus \{\omega_i\}) \cup \{\omega_i'\}$ größer als die von γ_i , so hat a_i einen neuen Lösungskandidaten gefunden: $\gamma_i' = \Omega_i' = (\Omega_i \setminus \{\omega_i\}) \cup \{\omega_i'\}$. Ist die Güte des Lösungskandidaten γ_i jedoch größer, so verwirft a_i die soeben ermittelte Fahrplanauswahl ω_i' wieder, und wählt stattdessen die in γ_i enthaltene Fahrplanauswahl $\tilde{\omega}_i$ aus: $\Omega_i' = (\Omega_i \setminus \{\omega_i\}) \cup \{\tilde{\omega}_i\}$. Der Lösungskandidat γ_i wird in letzterem Fall unverändert beibehalten: $\gamma_i' = \gamma_i$.

Die getroffenen Entscheidungen werden schließlich im Arbeitsgedächtnis gespeichert: $\kappa_i := (\zeta, \Omega_i', \gamma_i')$. Zuletzt teilt a_i das Arbeitsgedächtnis seinen Nachbarn mit, indem jeweils eine Nachricht $\mu_i = (a_k, \kappa_i)$ an alle benachbarten Verbundteilnehmer $a_k \in N_i$ verschickt wird ($\rightarrow$ Aktion).

Gemäß dieser Verhaltensregel sucht ein Verbundteilnehmer nach der Aktualisierung seines Wissens zunächst nach einer optimalen Fahrplanauswahl in Bezug auf den aktuell wahrgenommenen Systemzustand. Diese muss dann jedoch mit dem aktuellen Lösungskandidaten verglichen werden, da der Lösungskandidat die beste Lösung des Optimierungsproblems repräsentiert, die der Verbundteilnehmer bisher kennt. Kann hierbei eine noch bessere Zielerfüllung als die der momentan besten bekannten Lösung erreicht werden, so wurde eine neue beste Lösung gefunden, welche dann nachfolgend den Nachbarn des Verbundteilnehmers mitgeteilt wird und sich sukzessive ausbreitet. Kann der Lösungskandidat hingegen nicht verbessert werden, ist der Verbundteilnehmer verpflichtet, die sich auf ihn beziehende Fahrplanauswahl aus dem bestehenden Lösungskandidaten anzunehmen. In beiden Fällen beinhalten nach der Entscheidungsphase sowohl der aktuell wahrgenommene Systemzustand Ω_i als auch der Lösungskandidat γ_i die aktuelle Fahrplanauswahl eines Verbundteilnehmers a_i . Somit repräsentiert Ω_i zu jedem Zeitpunkt die aktuellsten Informationen des Verbundteilnehmers über den globalen Systemzustand, wohingegen die dem Verbundteilnehmer beste bekannte Lösung γ_i zu jedem Zeitpunkt den aktuell angestrebten Konsens des Verbundteilnehmers repräsentiert.

Die Berücksichtigung dieser Informationen stellt die in [Abschnitt 2.3.1](#) beschriebene Dämpfung durch negative Rückkopplung dar und führt schließlich zum Equi-

librium des Systems. Die Konvergenz des Systems zu diesem Equilibrium kann vorweggreifend durch folgende Argumentation verdeutlicht werden: Bestehende Lösungskandidaten werden ausschließlich durch Lösungskandidaten überschrieben, welche eine bessere Zielerfüllung erreichen. Dies entspricht bereits der geforderten *anytime* Eigenschaft. Irgendwann findet ein Verbundteilnehmer einen Lösungskandidaten, welcher sich mittels der obigen Informationssynthese über den gesamten Verbund ausbreitet, und von keinem Verbundteilnehmer weiter verbessert werden kann. Dieser Lösungskandidat sei mit γ^* bezeichnet. Aufgrund der Verpflichtung zur Einhaltung der Fahrplanvorgabe aus γ_i für alle a_i , und da nach vollständiger Ausbreitung von γ^* gilt $\forall a_i \in A: \gamma_i = \gamma^*$, wurde mit γ^* eine finale Lösung des PEP gefunden, auf die sich alle Verbundteilnehmer geeinigt haben. Somit wurde ein Konsens erzielt. Das Verfahren konvergiert also stets gegen den besten bisher identifizierten Lösungskandidaten. Sobald es keinem Verbundteilnehmer möglich ist, diesen noch weiter zu verbessern, werden keine weiteren Nachrichten mehr erzeugt, und das System befindet sich in einem Equilibrium (und terminiert damit, siehe [Abschnitt 4.3](#)). Eine detailliertere, formale Herleitung der Konvergenzeigenschaften inklusive der Berücksichtigung der im folgenden Abschnitt eingeführten freien Gesinnungswahl wird in [Kapitel 5](#) durchgeführt.

3.6.4 Designschritt 4: Freie Gesinnungswahl

Bisher wurde in den Betrachtungen bzgl. der Qualität oder Güte eines Lösungskandidaten stets vereinfacht von „Zielerfüllung“ gesprochen. Den Verbundteilnehmern soll es jedoch frei gestellt sein, sich kooperativ oder antagonistisch zu verhalten (siehe [Abschnitt 3.2](#)). Bei den Zielfunktionen f und $\forall a_i \in A: g_i$ aus [Definition 3.1.11](#) und [3.1.12](#) handelt es sich damit je nach Gesinnung der Verbundteilnehmer um potenziell konfliktäre Ziele (vgl. [Beispiel 3.2.1](#)). Daher muss das Entscheiden für eine Fahrplanauswahl eines Verbundteilnehmers genauer spezifiziert werden.

Wie auch in [\[HLS14b\]](#) dargelegt, handelt es sich bei dem PEP um ein multikriterielles kombinatorisches Optimierungsproblem (engl. *Multiple Objective Combinatorial Optimization*, kurz MOCO) mit der Zielfunktion f als erstem Optimierungskriterium, und der aggregierten Menge aller Zielfunktionen g_i als zweitem Kriterium. Die Aufgabe ist es, eine hinsichtlich beider Kriterien möglichst optimale Lösung zu finden. Dies entspricht der Suche nach *effizienten*, d. h. Pareto-optimalen Lösungen [\[EG00\]](#).

Eine solche Lösung ist dabei so definiert, dass sie im Zielfunktionsraum in keiner Dimension von einer anderen Lösung dominiert wird, d. h. dass keine Lösung existiert, die bzgl. mindestens eines Kriteriums eine bessere Güte erreicht. Daraus folgt jedoch, dass es üblicherweise eine Menge von mehreren Pareto-optimalen Lösungen gibt, von denen schließlich eine Lösung ausgewählt werden muss. Zentrale Lösungsverfahren für multikriterielle Probleme können je nach Herangehensweise in drei verschiedene Kategorien eingeteilt werden [EG00, S. 430]:

a priori Alle Präferenzen des Entscheidungsträgers bzgl. der Gewichtung der Kriterien sind zu Beginn des Prozesses bekannt, sodass unmittelbar hinsichtlich dieser Präferenzen nach einer optimalen Lösung gesucht werden kann.

a posteriori Es wird zunächst die (exakte oder approximierte) Menge der Pareto-optimalen Lösungen bestimmt. Im Nachgang wird aus dieser Menge eine finale Lösung hinsichtlich der Präferenzen des Entscheidungsträgers ausgewählt.

interaktiv Der Entscheidungsträger wirkt selbst am Optimierungsprozess mit und lässt seine Präferenzen an passenden Stellen in den Entscheidungsprozess einfließen.

Im vorliegenden System liegen die Zielfunktionen g_i , welche zusammen das zweite Optimierungskriterium bilden, verteilt vor. Sie sind zwar bereits zu Beginn des Optimierungsprozesses bekannt, stellen aber privates Wissen dar und können nicht für eine globale Gewichtung der Kriterien in einer *a priori* Herangehensweise verwendet werden. Ebenso wenig anwendbar ist die *a posteriori* Herangehensweise, da auch hier globales Wissen über alle Präferenzen zur Entscheidung erforderlich ist. Daher wird die *interaktive* Herangehensweise für die vorliegende Problemstellung adaptiert. Im Vergleich zu einem zentralen Verfahren existiert hier allerdings nicht ein einzelner Entscheidungsträger. Stattdessen kann jeder Verbundteilnehmer als eigenständiger Entscheidungsträger betrachtet werden, dessen Entscheidungsgewalt sich auf die von ihm beigetragene Fahrplanauswahl beschränkt.

Ein Verbundteilnehmer soll somit für jeden möglichen Fahrplan selbst entscheiden können, ob dieser hinsichtlich seiner lokalen Präferenzen akzeptabel ist oder nicht. In die Entscheidung kann die Erfüllung der globalen Zielfunktion f mit einfließen: Ein Fahrplan, welcher die Zielerfüllung bzgl. Funktion f sehr stark maximiert, wird bei einer beispielsweise eher kooperativen Gesinnung des Verbundteilnehmers ggf.

trotz geringer Erfüllung von Funktion g_i ausgewählt werden. Diese Entscheidung hängt dann jedoch vom aktuell wahrgenommenen Systemzustand ab, und muss mit jeder Aktualisierung des Wissensspeichers neu getroffen werden. Konkret bedeutet dies, dass jeder Verbundteilnehmer mit jeder Aktualisierung seines Wissens ein eigenes multikriterielles Optimierungsproblem lösen muss: Basierend auf dem aktuellen wahrgenommenen Systemzustand Ω_i einerseits, und den lokalen Präferenzen andererseits, muss a_i einen Fahrplan $\theta_i \in S_i$ derart auswählen, dass beide Zielfunktionen f und g_i möglichst maximal erfüllt sind, wobei je nach Gesinnung eine Priorisierung der Zielfunktionen vorgenommen werden kann.

Eine mögliche Vorgehensweise ist hier die Betrachtung der gewichteten Summe aller Optimierungskriterien als zusammengesetzte Zielfunktion, im vorliegenden Fall wäre dies etwa $z_i = \alpha \cdot f + (1 - \alpha) \cdot g_i$, wobei $\alpha \in [0, 1]$ von a_i frei gewählt werden kann. Problematisch hierbei ist, dass aufgrund der diskreten Natur kombinatorischer Probleme Teile des Lösungsraumes mit dieser Modellierung unerreichbar bleiben [EG00, S. 429]. Eine weitere Möglichkeit ist die sog. *lexikografische* Zielfunktion, in welcher zunächst eine Priorisierung der Optimierungskriterien in ein primäres und ein sekundäres Kriterium vorgenommen wird. Ein Fahrplan θ_i ist hierbei immer dann einem anderen Fahrplan θ_i' vorzuziehen, wenn θ_i bzgl. des primären Kriteriums eine größere Güte erreicht als θ_i' . Erst wenn beide bzgl. des primären Kriteriums eine identische Güte erlangen, wird das sekundäre Kriterium als Entscheidungsgrundlage herangezogen. Eine Variante hiervon ist die sog. *min-max* Zielfunktion, bei welcher ausgehend von einer vorliegenden Situation immer das momentan am schlechtesten abschneidende Kriterium priorisiert wird.

Aufgrund der Forderung einer freien Gesinnungswahl wird für das vorliegende System keine konkrete Vorgehensweise zur Lösung der lokalen multikriteriellen Optimierungsprobleme vorgeschrieben. Um dennoch unter diesen Bedingungen die Konvergenz zu einer global möglichst optimalen Lösung garantieren zu können, werden folgende Richtlinien aufgestellt:

1. Hat ein Verbundteilnehmer a_i im Rahmen seines Entscheidungsprozesses eine für ihn akzeptable Fahrplanauswahl ω_i identifiziert und per Nachricht an andere Verbundteilnehmer kommuniziert, so kann diese Fahrplanauswahl von a_i im Laufe des Optimierungsprozesses nicht mehr widerrufen werden. (Im Falle eines *kritischen Ereignisses* kann diese Richtlinie eingeschränkt aufgehoben

werden, mehr dazu in [Abschnitt 3.8.](#))

2. Sobald ein Verbundteilnehmer a_i das zu erfüllende Wirkleistungsprodukt mitgeteilt bekommt, muss er eine Fahrplanauswahl treffen.

Auf diese Weise erhält jeder Verbundteilnehmer die Freiheit, seinen persönlichen Lösungsraum aus realisierbaren Fahrplänen nach eigenem Ermessen einzuschränken, solange eine einmal publizierte Fahrplanauswahl für den gesamten nachfolgenden Zeitraum des Optimierungsprozesses als potenziell verpflichtend anerkannt wird (Regel 1). Zu Beginn des Verfahrens ist ein Verbundteilnehmer jedoch verpflichtet, eine beliebige Fahrplanauswahl zu treffen (Regel 2). Dies bewirkt, dass alle Verbundteilnehmer mindestens einen Fahrplan auswählen und veröffentlichen. Hieraus ergibt sich die folgende erweiterte Verhaltensregel:

Lokale Verhaltensregel, Version 4

Empfängt ein Verbundteilnehmer a_i eine Nachricht μ_j , so wird zunächst das Arbeitsgedächtnis wie in [Lokale Verhaltensregel, Version 3](#) beschrieben aktualisiert ($\rightarrow$ Wahrnehmung).

Für die darauf folgende Optimierung der Fahrplanauswahl ($\rightarrow$ Entscheidung) sei wiederum $\omega_i = (a_i, \theta_i, \lambda_i) \in \Omega_i$ die aktuelle Fahrplanauswahl von a_i , und $\tilde{\omega}_i = (a_i, \tilde{\theta}_i, \tilde{\lambda}_i) \in \gamma_i$ die im Lösungskandidaten enthaltene Fahrplanauswahl von a_i . Der Verbundteilnehmer ermittelt nun eine Fahrplanauswahl $\omega_i' = (a_i, \theta_i', \lambda_i')$ mit $\lambda_i' = \lambda_i + 1$, welche eine für ihn in der derzeitigen Situation (hinsichtlich Ω_i und seiner lokalen Präferenzen) optimale Wahl darstellt. Die Vorgehensweise hierzu sei dem Verbundteilnehmer freigestellt, insbesondere kann er die Zielfunktionen f und g_i aus [Definition 3.1.11](#) bzw. [3.1.12](#) nach eigenem Ermessen priorisieren. Ist die aktuelle Fahrplanauswahl ω_i derzeit leer (d. h. μ_j ist die erste Nachricht, die a_i im Rahmen des Einsatzplanungsprozesses empfängt), so ist a_i dazu verpflichtet, einen Fahrplan auswählen und damit eine neue Fahrplanauswahl ω_i' zu erstellen. Ist ω_i jedoch derzeit nicht leer und a_i findet keinen akzeptablen abweichenden Fahrplan im Suchraummodell, so darf die aktuelle Fahrplanauswahl mit $\omega_i' = \omega_i$ auch unverändert beibehalten werden.

*Für den nun durchzuführenden Vergleich darf ausschließlich die **globale Güte***

hinsichtlich Zielfunktion f betrachtet werden: Ist die globale Güte von $(\Omega_i \setminus \{\omega_i\}) \cup \{\omega_i'\}$ größer als die von γ_i , so hat a_i einen neuen Lösungskandidaten gefunden: $\gamma_i' = \Omega_i' = (\Omega_i \setminus \{\omega_i\}) \cup \{\omega_i'\}$. Ist die globale Güte des Lösungskandidaten γ_i jedoch größer oder gleich groß, so verwirft a_i die soeben ermittelte Fahrplanauswahl ω_i' wieder, und wählt stattdessen ungeachtet seiner lokalen Präferenzen die in γ_i enthaltene Fahrplanauswahl $\tilde{\omega}_i$ aus: $\Omega_i' = (\Omega_i \setminus \{\omega_i\}) \cup \{\tilde{\omega}_i\}$. Der Lösungskandidat γ_i wird in letzterem Fall unverändert beibehalten: $\gamma_i' = \gamma_i$.

Die Veröffentlichung des resultierenden aktualisierten Arbeitsgedächtnisses erfolgt anschließend wie in [Lokale Verhaltensregel, Version 3](#) beschrieben ($\rightarrow$ Aktion).

Ein Verbundteilnehmer kann also bei jeder neu aufzustellenden Fahrplanauswahl eine beliebige Priorisierung seiner persönlichen Präferenzen gegenüber dem globalen Optimierungsziel (oder umgekehrt) vornehmen, und seine möglichen Fahrpläne diesbezüglich bewerten. Die im vorigen Designschritt eingeführte Verpflichtung zur Einhaltung der Fahrplanvorgabe aus γ_i bleibt davon aber unberührt: Eine einmal veröffentlichte Fahrplanauswahl muss von einem Verbundteilnehmer ungeachtet seiner lokalen Präferenzen zu jedem zukünftigen Zeitpunkt akzeptiert werden, wenn diese zu einer höheren globalen Güte beiträgt. Die Konsequenz für das resultierende selbstorganisierende System ist somit lediglich ein potenziell durch die lokalen Präferenzen der Verbundteilnehmer eingeschränkter Lösungsraum – die Konsensbildung ist weiterhin gegeben (siehe [Abschnitt 5.3](#) für eine formale Betrachtung).

Mit diesem Designschritt ist die Ausgestaltung der lokalen Verhaltensregeln, und damit Paradigma #1 des Vorgehensmodells nach Prehofer und Bettstetter abgeschlossen.

3.7 PARADIGMA #2: IMPLIZITE KOORDINATION

Wie bereits in [Abschnitt 3.4](#) erwähnt, dienen die Paradigmen #1 und #2 des Vorgehensmodells nach Prehofer und Bettstetter der Ausgestaltung des Kerns des Selbstorganisationsmechanismus, wobei eines der beiden Paradigmen als initiale Herangehensweise gewählt werden sollte. Im vorliegenden Fall wurde Paradigma #1 als grundlegende Herangehensweise gewählt, und mit dem daraus resultierenden Ergebnis bereits der wesentliche Teil des Selbstorganisationsmechanismus erarbeitet.

Die Anwendung von Paradigma #2 erbringt hier keine weiteren sinnvollen Ergänzungen. Die in diesem Paradigma fokussierte implizite Koordination ist inhärent in den aufgestellten lokalen Verhaltensregeln enthalten: Es werden keine direkten Verträge oder Absprachen zwischen Verbundteilnehmern ausgehandelt, sondern es wird eine iterative reaktive Anpassung vorgenommen, welche in Verbindung mit den aufgestellten Regeln zur Ausbreitung des Wissens und zur Auswahl neuer Lösungskandidaten implizit eine Koordination herbeiführt.

3.8 PARADIGMA #3: ADAPTIVITÄT

Selbstorganisierenden Systemen wird allgemein die Fähigkeit zur Adaptivität zugesprochen [Ash47; Hey01; Ger07; VDE10], d. h. diese Systeme können so konstruiert werden, dass ihre Funktion auch bei auftretenden Störungen (bis zu einer gewissen Schwere) gewährleistet werden kann. Solche Störungen können bspw. durch das Fehlverhalten einzelner Entitäten verursacht werden. Auch eine Überlastung des Systems aufgrund großer Datenströme aus der Umgebung des Systems stellt eine solche Störung dar. Zur Reaktion auf diese Störungen werden in [PB05] drei Ebenen der Adaptivität in aufsteigender Reihenfolge der erforderlichen Anpassungen identifiziert: (1) Der Selbstorganisationsmechanismus kann so konstruiert sein, dass Störungen implizit durch den normalen Betrieb des Systems ausgeglichen werden. (2) Das System kann einzelne Betriebsparameter bei Störungen dahingehend selbstständig anpassen, dass weiterhin maximale Performanz erreicht wird. (3) Der grundlegende Mechanismus des Systems kann vom System auf verschiedene Störungen hin angepasst oder ersetzt werden.

Im vorliegenden Anwendungsfall der selbstorganisierten Einsatzplanung ist insbesondere das spontane Fehlverhalten von Energieanlagen als Störungsursache zu betrachten. Jeder Agent im betrachteten System repräsentiert eine Energieanlage, für welche eine Planung vorgenommen werden soll. Hierzu werden alle für diese Anlage über dem Planungshorizont realisierbaren Fahrpläne, in einem Suchraummodell gekapselt, dem Agenten als Planungsgrundlage zur Verfügung gestellt. Diese Menge der realisierbaren Fahrpläne wurde dabei auf Basis von Annahmen bzgl. des erwarteten Verhaltens der Anlage erzeugt. Im Falle eines wärmegeführten Mikro-Blockheizkraftwerkes werden bspw. Annahmen über den erwarteten Wärmebedarf

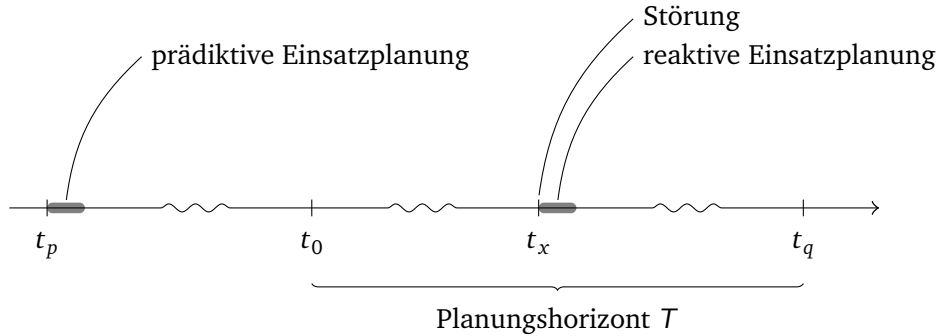


Abbildung 3.5: Zusammenhang zwischen prädiktiver und reaktiver Einsatzplanung.

getroffen, um aus den verbleibenden Freiheitsgraden mögliche realisierbare Fahrpläne berechnen zu können. Treffen diese Annahmen aufgrund unerwarteter Ereignisse später nicht mehr zu, muss das entsprechende Suchraummodell für diese Energieanlage neu berechnet werden (vgl. [Abschnitt 6.4](#)), und stellt damit im Sinne der Selbstorganisation eine Störung des Systems dar. Neben aktualisierten Suchraummodellen sind prinzipiell aber auch andere Störungsarten denkbar, wie etwa eine Modifikation des Wirkleistungsproduktes (z. B. durch eine Veränderung des Planungshorizontes).

Solche Situationen sind zwar nicht mehr Teil der prädiktiven Einsatzplanung, sondern müssten wie bereits in [Abschnitt 3.1](#) erläutert im Rahmen einer reaktiven Einsatzplanung behandelt werden. In [Abbildung 3.5](#) ist zur Veranschaulichung der zeitliche Zusammenhang zwischen prädiktiver und reaktiver Einsatzplanung visualisiert. Dennoch soll im Folgenden demonstriert werden, dass das vorliegende Verfahren mit leichten Anpassungen auch für einen nachgelagerten reaktiven Einsatz verwendet werden kann. Hierzu wird exemplarisch die soeben beschriebene Aktualisierung von Suchraummodellen als Störungstyp herangezogen.

Definition 3.8.1 (Störung)

Sei a_i ein Verbundteilnehmer und T ein Planungshorizont. Dann ist eine *Störung* $\Delta = (t_x, a_i, \sigma_i')$ ein Ereignis, das zu einem im Planungshorizont liegenden Planungsintervall $t_x \in T$ auftritt und eine Aktualisierung des Suchraummodells σ_i des Verbundteilnehmers a_i bewirkt: $\sigma_i := \sigma_i'$.

Das in den vorangegangenen Abschnitten entwickelte Verfahren zur Einsatzplanung

ist genau dann terminiert, wenn alle Verbundteilnehmer sich auf einen identischen Lösungskandidaten γ^* geeinigt haben (auf Möglichkeiten zur verteilten Terminierungserkennung wird in 4.3 noch näher eingegangen). Die Agenten befinden sich dann in einem *passiven* Zustand: Da sie keine Nachrichten mehr von benachbarten Agenten empfangen, gibt es keinen Anlass ihrerseits Nachrichten zu versenden. Angenommen, zu einem im Planungshorizont liegenden Planungsintervall $t_x \in T$ tritt nun eine Störung $\Delta = (t_x, a_i, \sigma_i')$ gemäß obiger Definition auf, und die resultierende Aktualisierung des Suchraummodells des betreffenden Verbundteilnehmers bewirkt, dass die vereinbarte Fahrplanauswahl $\omega_i \in \gamma^*$ nicht mehr eingehalten werden kann. In diesem Fall müsste das System reaktiviert werden und erneut eine Einsatzplanung vornehmen, um die Störung zu kompensieren (dies ist die *reaktive Einsatzplanung* in [Abbildung 3.5](#)).

Die durch die Störung induzierte reaktive Einsatzplanung zum Zeitpunkt t_x kann dabei entweder als Neuaufwurf der Planung durchgeführt werden, oder es kann eine Umplanung vorgenommen werden. Im Folgenden werden beide Möglichkeiten kurz beleuchtet, wobei mit den dafür jeweils erforderlichen Anpassungen des Selbstorganisationsmechanismus eine Adaptivität der einleitend beschriebenen Ebene 2 erzielt wird, d. h. dem System ist es erlaubt, einzelne Betriebsparameter zu rekonfigurieren, um einer Störung entgegen zu wirken. Der grundlegende Mechanismus soll dabei jedoch erhalten bleiben.

3.8.1 Neuaufwurf der Planung

Der Neuaufwurf einer Planung ist dadurch gekennzeichnet, dass bei Auftreten einer Störung eine erneute vollständige Einsatzplanung durchgeführt wird. Im vorliegenden System kann dies wie folgt realisiert werden: Jeder Agent überwacht die ihm assoziierte Energieanlage auf Störungen gemäß [Definition 3.8.1](#). Tritt eine Störung $\Delta = (t_x, a_i, \sigma_i')$ auf, wird diese dem Verbundsprecher $\hat{a}$ per Nachricht mitgeteilt. Der Verbundsprecher erstellt nun eine Kopie des Wirkleistungsproduktes (wobei diese Kopie lediglich noch den verbleibenden Teil $\{t_x, t_{x+1}, \dots, t_{q-1}\}$ des Planungshorizontes T umfassen muss) und markiert diese als *neu*. Anschließend leert er sein Arbeitsgedächtnis κ , und befolgt anschließend die in den vorigen Abschnitten aufgestellten Verhaltensregeln. Auf diese Weise beginnt ein neues Einsatzplanungsverfahren.

Für diese Erweiterung des Verfahrens wären folgende Anpassungen gegenüber

den in diesem Kapitel aufgestellten Definitionen und Verhaltensregeln erforderlich:

- Jedem Verbundteilnehmer muss der Bezeichner des Verbundsprechers $\hat{a}$ bekannt sein, damit dieser im Falle einer Störung unabhängig von den durch das Overlay-Netz induzierten Nachbarschaftsbeziehungen kontaktiert werden kann. Dies ist ohne weiteres möglich, da das zugrundeliegende Kommunikationsnetz prinzipiell die Kommunikation zwischen beliebigen Verbundteilnehmern erlaubt (vgl. [Definition 3.3.1](#)).
- Dem Wirkleistungsprodukt muss, analog zu einer Fahrplanauswahl in [Definition 3.6.3](#), eine Zählvariable mitgegeben werden. Diese darf nur vom Verbundsprecher modifiziert werden und wird immer dann inkrementiert, wenn der Verbundsprecher ein neues Wirkleistungsprodukt aufgrund einer Störung ausschreiben möchte. Dies ist erforderlich, damit Verbundteilnehmer ein empfangenes Wirkleistungsprodukt als *neu* erkennen.
- Sobald ein Verbundteilnehmer a_i ein neues Wirkleistungsprodukt erhält, leert er sein Arbeitsgedächtnis κ_i und berechnet sein Suchraummodell σ_i unter Berücksichtigung der vom jetzigen Zeitpunkt an realisierbaren Fahrpläne der Anlage neu, bevor er mit den existierenden Verhaltensregeln fortfährt.

Diese Anpassungen genügen, um bei einer Störung eine neue Einsatzplanung für den verbleibenden Planungshorizont zu initialisieren. Es ist jedoch leicht ersichtlich, dass nicht immer ein Neuaufwurf erforderlich ist. Insbesondere bei geringfügigen Störungen ist eine lokale Umplanung zu bevorzugen.

3.8.2 Umplanung

Eine Umplanung unterscheidet sich von einem Neuaufwurf dadurch, dass üblicherweise nicht die gesamte Planung neu erstellt wird. Stattdessen werden einige wenige Anpassungen vorgenommen, welche die aufgetretene Störung bereits kompensieren. Hierbei lassen sich verschiedene sekundäre Optimierungsziele definieren, bspw. kann eine Umplanung mit möglichst wenig Planungsschritten erwünscht sein, um zeitnah reagieren zu können. Alternativ kann die Anzahl der erforderlichen Anpassungen als Kriterium verwendet werden, um etwa durch lokale Umplanungen

verursachte Kosten zu minimieren (siehe [Trö10; Son+12] für weitere Details zur reaktiven/kontinuierlichen Einsatzplanung).

Die Realisierung der Umplanung im vorliegenden System ist etwas aufwändiger als der im vorigen Abschnitt beschriebene Neuaufwurf. Ein möglicher Ansatz hierzu wäre, dass im Gegensatz zum Neuaufwurf die Arbeitsgedächtnisse der Verbundteilnehmer nicht geleert, sondern soweit wie möglich wiederverwendet werden, um mit möglichst wenig Fahrplananpassungen eine Kompensation der Störung zu erzielen. Dabei wären aber insbesondere folgende Fragestellungen zu klären: Darf jeder Verbundteilnehmer bei einer auftretenden Störung lokal ein aktualisiertes Wirkleistungsprodukt ausschreiben? Wie wird der noch aktuelle Lösungskandidat γ^* als ungültig markiert? Wie kann erreicht werden, dass die Reaktionen auf eine Neuausschreibung lokal begrenzt sind? Wie kann hierbei eine Netzverträglichkeit mit berücksichtigt werden? Was passiert, wenn mehrere Störungen parallel auftreten? Die Beantwortung dieser Fragen liegt jedoch außerhalb des Kontextes der vorliegenden Arbeit. Ein aktuell verfolgter Ansatz hierzu ist etwa in [NS13] zu finden.

3.9 PARADIGMA #4: REDUZIERTER ZUSTANDSINFORMATIONEN

Nach Prehofer und Bettstetter führt die Vorhaltung langlebiger Zustandsinformationen, beispielsweise induziert durch starke Abhängigkeiten zwischen den aktiven Systemelementen, zu einem geringeren Grad an Selbstorganisation. Die in Paradigma #4 fokussierte Reduktion solch langlebiger Informationen wird üblicherweise nicht direkt angewendet, sondern ergibt sich als Konsequenz aus den Paradigmen #1 und #2. Dies ist auch im vorliegenden System der Fall: Die Einführung des Overlay-Netzes führt unmittelbar dazu, dass ein Verbundteilnehmer selbst bei aktiver Kommunikation mit anderen Verbundteilnehmern keine verlässlichen globalen Informationen besitzt. Stattdessen stellen seine gesammelten Informationen kurzlebige Annahmen über das System dar, die mit jeder empfangenen Nachricht aktualisiert werden. Eine weitere potenzielle Quelle langlebiger Informationen ist nach [PB05] der Bedarf an Synchronisation, denn sobald die Entitäten eines verteilten Prozesses synchronisiert sind, haben sie einen gemeinsamen globalen Zustand eingenommen. Im vorliegenden System ist Synchronisation in Form der Einigung aller

Verbundteilnehmer auf eine gemeinsame Lösung vorhanden. Dies ist aber ein *Ergebnis* des Prozesses, und nur geringfügig im Optimierungsprozess selbst erforderlich. Insgesamt weist das System somit einen bzgl. des Selbstorganisationsmechanismus geringen Grad an Abhängigkeiten zwischen den Verbundteilnehmern auf. Langlebige Zustandsinformationen werden erst bei Konvergenz zur finalen Lösung aufgebaut.

4

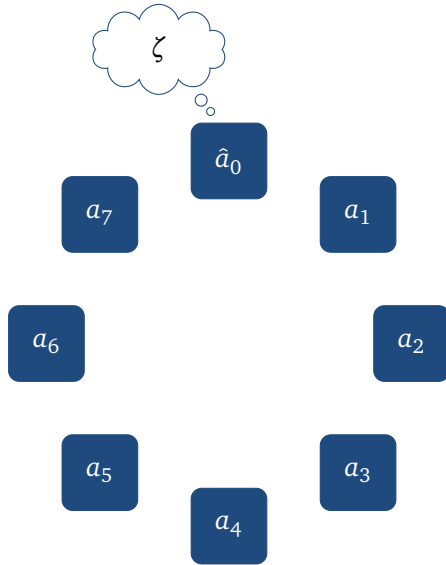
HEURISTIK: COHDA

In [Kapitel 3](#) wurde ein Ansatz zur Lösung verteilter kombinatorischer Optimierungsprobleme mit speziellem Fokus auf die prädiktive Einsatzplanung im Smart Grid entwickelt. Im Folgenden wird die dabei entstandene Heuristik COHDA (*Combinatorial Optimization Heuristic for Distributed Agents*) noch einmal zusammenfassend unter Verwendung der in [Kapitel 3](#) aufgestellten Definitionen dargestellt und in einen Ausführungskontext eingebettet.

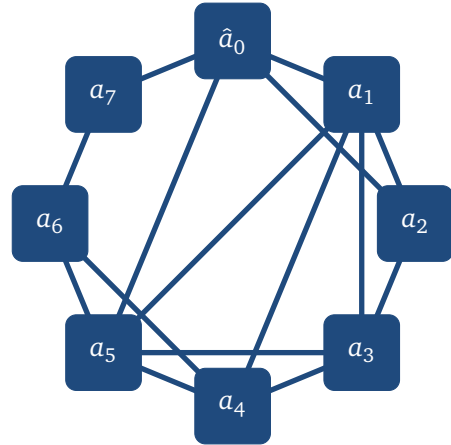
4.1 START DER HEURISTIK

Die Ausgangslage des von COHDA zu lösenden Optimierungsproblems besteht aus einem Verbund A von kommunikationstechnisch vernetzten Energieanlagen, jeweils durch einen Softwareagenten repräsentiert, sowie einem zu erfüllenden Wirkleistungsprodukt ζ . Sowohl A als auch ζ werden von einem vorgelagerten Verbundbildungsprozess generiert, siehe dazu bspw. [\[BSA11; Son+12\]](#). Die Aufgabe von COHDA ist es nun, für jeden Verbundteilnehmer einen hinsichtlich systemglobaler sowie anlagenlokaler Kriterien möglichst optimalen Fahrplan zu finden. Aufgrund der Verteiltheit des gegebenen Systems wird hierzu ein Verfahren mit selbstorganisierenden Eigenschaften eingesetzt. Dieses bildet zunächst ein Overlay-Netz zur Reduktion des Kommunikationsaufwandes und führt dann einfache, individuelle Entscheidungsprozesse auf Ebene der Verbundteilnehmer aus, um auf diese Weise zu einem Konsens zu gelangen, der die genannten Kriterien erfüllt.

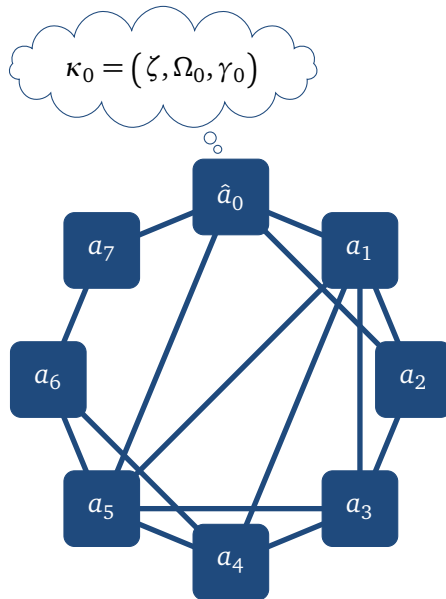
In [Abbildung 4.1](#) ist der Startvorgang der Heuristik anhand eines Beispiels visualisiert. Der dargestellte Verbund $A = \{\hat{a}_0, a_1, \dots, a_7\}$ besteht aus acht Verbundteilnehmern, wobei $\hat{a}_0$ der Verbundsprecher ist. Letzterer fungiert als Repräsentant des Verbundes sowie als Schnittstelle zu externen Systemen – daher besitzt er initial alleinige Kenntnis über das zu erfüllende Wirkleistungsprodukt ζ (siehe [Abbildung 4.1\(a\)](#)).



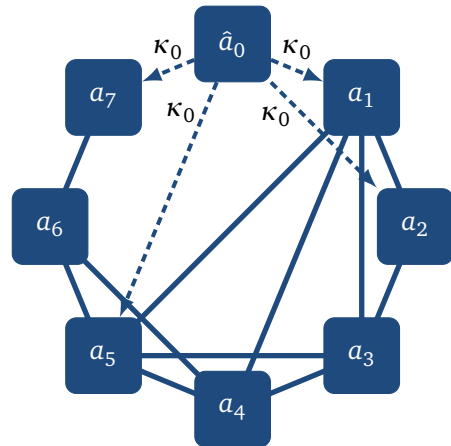
(a) Initial kennt nur der Verbundsprecher $\hat{a}_0$ das Wirkleistungsprodukt ζ .



(b) Der Verbundsprecher $\hat{a}_0$ bildet ein Overlay-Netz und informiert alle Verbundteilnehmer a_i über deren jeweilige Nachbarschaften N_i .



(c) Der Verbundsprecher $\hat{a}_0$ erzeugt die erste Fahrplanauswahl für seine Energieanlage, und füllt sein Arbeitsgedächtnis κ_0 entsprechend.



(d) Indem $\hat{a}_0$ sein Arbeitsgedächtnis κ_0 an seine Nachbarn sendet, wird die eigentliche Konsensfindung gestartet.

Abbildung 4.1: Die initialen Schritte zum Starten der COHDA Heuristik anhand eines Beispiels mit acht Verbundteilnehmern.

Der Verbundsprecher bildet nun auf dem existierenden Kommunikationsnetz (hier nicht dargestellt) ein Overlay-Netz, um die Kommunikationskomplexität der nachfolgenden Konsensfindung zu reduzieren ([Abbildung 4.1\(b\)](#)). In dem visualisierten Beispiel wurde eine *Small-World* Topologie nach der Konstruktionsvorschrift des verbesserten Modells von Watts und Strogatz [[Str01](#)] als Overlay-Netz verwendet. Dieser Topologie-Typ zeichnet sich durch eine kurze durchschnittliche Weglänge zwischen zwei beliebigen Knoten aus, was zu einer verbesserten Signalausbreitung führt und damit weitere positive Eigenschaften wie etwa eine schnellere Synchronisation oder eine höhere Berechnungsleistung (des Gesamtsystems) bewirken kann [[Str01](#), S. 273]. Es sind hier jedoch prinzipiell beliebige Topologien anwendbar, solange sie ein schwarmbildendes Overlay-Netz gemäß [Definition 3.6.1](#) darstellen (in [Abschnitt 7.3.1](#) werden verschiedene Topologien bzgl. ihrer Effekte untersucht). Der Verbundsprecher informiert anschließend alle Verbundteilnehmer bzgl. des Overlay-Netzes, sodass diesen nun ihre jeweiligen Nachbarschaften N_i bekannt sind.

4.2 VERHALTEN DER HEURISTIK

Nach diesen vorbereitenden Schritten beginnt der Verbundsprecher $\hat{a}_0$ mit der heuristischen Einsatzplanung, indem er das in [Kapitel 3](#) modellierte Verhalten ausführt. Dazu werden die drei Phasen *Wahrnehmung*, *Entscheidung* und *Aktion* sequenziell durchlaufen. Der Vollständigkeit halber werden diese Phasen nun noch einmal zusammenfassend aus der Sicht eines beliebigen Verbundteilnehmers a_i dargestellt:

Wahrnehmung Empfängt ein Verbundteilnehmer a_i eine Nachricht $\mu_j = (a_j, \kappa_j)$ mit $\kappa_j = (\zeta, \Omega_j, \gamma_j)$ von einem benachbarten Verbundteilnehmer a_j , so aktualisiert a_i zunächst sein Arbeitsgedächtnis κ_i mit den in κ_j enthaltenen Informationen:

- Enthält das eigene Arbeitsgedächtnis κ_i noch kein Wirkleistungsprodukt ζ , speichere das Wirkleistungsprodukt aus μ_j in κ_i ab.
- Aktualisiere den eigenen wahrgenommenen Systemzustand Ω_i mit den in Ω_j enthaltenen Informationen nach den Regeln aus [Definition 3.6.5](#).
- Aktualisiere den eigenen Lösungskandidaten γ_i mit den in γ_j enthaltenen Informationen nach den Regeln aus [Definition 3.6.6](#).

Hierbei kann es vorkommen, dass μ_j keine neuen Informationen beinhaltet, sodass sich das Arbeitsgedächtnis von a_i nicht ändert. In diesem Fall sind die folgenden Phasen überflüssig und werden *nicht* ausgeführt.

Entscheidung Wurde das Arbeitsgedächtnis im vorigen Schritt modifiziert, optimiert der Verbundteilnehmer nun seine eigene Fahrplanauswahl. Sei dazu $\omega_i = (a_i, \theta_i, \lambda_i) \in \Omega_i$ die aktuelle Fahrplanauswahl von a_i , und $\tilde{\omega}_i = (a_i, \tilde{\theta}_i, \tilde{\lambda}_i) \in \gamma_i$ die im Lösungskandidaten enthaltene Fahrplanauswahl von a_i . Der Verbundteilnehmer ermittelt nun eine Fahrplanauswahl $\omega_i' = (a_i, \theta_i', \lambda_i')$ mit $\lambda_i' = \lambda_i + 1$, welche eine für ihn in der derzeitigen Situation (hinsichtlich Ω_i und seiner lokalen Präferenzen) optimale Wahl darstellt. Die Vorgehensweise hierzu sei dem Verbundteilnehmer freigestellt, insbesondere kann er die Zielfunktionen f und g_i aus [Definition 3.1.11](#) bzw. [3.1.12](#) nach eigenem Ermessen priorisieren. Ist die aktuelle Fahrplanauswahl ω_i derzeit leer (d. h. μ_j ist die erste Nachricht, die a_i im Rahmen des Einsatzplanungsprozesses empfängt), so ist a_i dazu verpflichtet, einen Fahrplan auszuwählen und damit eine neue Fahrplanauswahl ω_i' zu erstellen. Ist ω_i jedoch derzeit nicht leer und a_i findet keinen akzeptablen abweichenden Fahrplan im Suchraummodell, so darf die aktuelle Fahrplanauswahl mit $\omega_i' = \omega_i$ auch unverändert beibehalten werden.

Für den nun durchzuführenden Vergleich darf ausschließlich die **globale Güte** hinsichtlich Zielfunktion f betrachtet werden: Ist die globale Güte von $(\Omega_i \setminus \{\omega_i\}) \cup \{\omega_i'\}$ größer als die von γ_i , so hat a_i einen neuen Lösungskandidaten gefunden: $\gamma_i' = \Omega_i' = (\Omega_i \setminus \{\omega_i\}) \cup \{\omega_i'\}$. Ist die globale Güte des Lösungskandidaten γ_i jedoch größer oder gleich groß, so *verwirft* a_i die soeben ermittelte Fahrplanauswahl ω_i' wieder, und wählt stattdessen ungeachtet seiner lokalen Präferenzen die in γ_i enthaltene Fahrplanauswahl $\tilde{\omega}_i$ aus: $\Omega_i' = (\Omega_i \setminus \{\omega_i\}) \cup \{\tilde{\omega}_i\}$. Der Lösungskandidat γ_i wird in letzterem Fall unverändert beibehalten: $\gamma_i' = \gamma_i$. Die getroffenen Entscheidungen werden schließlich im Arbeitsgedächtnis gespeichert: $\kappa_i := (\zeta, \Omega_i', \gamma_i')$.

Aktion Wurde das Arbeitsgedächtnis in der Phase *Wahrnehmung* modifiziert, teilt a_i das Arbeitsgedächtnis (samt der ggf. in der Phase *Entscheidung* entstandenen Modifikationen) seinen Nachbarn mit, indem jeweils eine Nachricht $\mu_i = (a_k, \kappa_i)$ an alle benachbarten Verbundteilnehmer $a_k \in N_i$ verschickt wird.

Da $\hat{a}_0$ als erster Akteur des Verfahrens noch keine Nachrichten von anderen Verbundteilnehmern erhalten hat, entfällt die Phase *Wahrnehmung* an dieser Stelle. Es wird somit direkt der Schritt *Entscheidung* ausgeführt, woraus das initiale Arbeitsgedächtnis κ_0 (bestehend aus Wirkleistungsprodukt, wahrgenommenem Systemzustand und Lösungskandidat) von $\hat{a}_0$ resultiert (siehe [Abbildung 4.1\(c\)](#)). Es folgt die Phase *Aktion*, in welcher $\hat{a}_0$ sein Arbeitsgedächtnis per Nachricht an seine unmittelbaren Nachbarn kommuniziert ([Abbildung 4.1\(d\)](#)). Diese führen beim Empfang der Nachrichten ebenfalls das beschriebene Verhalten aus.

4.3 TERMINIERUNG DER HEURISTIK

Auf die oben beschriebene Weise wird nun eine verteilte Einsatzplanung durchgeführt, welche genau dann terminiert, sobald sich alle Verbundteilnehmer auf den gleichen Lösungskandidaten geeinigt haben und somit ein Konsens gebildet wurde. Zu diesem Zeitpunkt besitzen alle Verbundteilnehmer identische Arbeitsgedächtnisse, und kein Verbundteilnehmer wird neue Nachrichten erzeugen. Die Terminierung entspricht im vorliegenden System somit einem Gleichgewichtszustand (Equilibrium), in welchem kein Akteur mehr aktiv ist und alle Nachrichten zugestellt wurden. In dieser Situation ist allen Verbundteilnehmern der von ihnen aufzuführende Fahrplan bekannt (da im Lösungskandidaten enthalten) und kann bei Eintritt in den Erfüllungszeitraum (vgl. [Abbildung 3.2](#) auf S. 39) von der assoziierten Energieanlage umgesetzt werden. Insofern muss die Terminierung für den beschriebenen Anwendungsfall nicht gesondert festgestellt werden, solange davon ausgegangen werden kann, dass der Terminierungszeitpunkt vor dem Beginn des Erfüllungszeitraumes liegt. Es sind jedoch Anwendungsfälle denkbar, in denen die Terminierung der Heuristik explizit detektiert werden muss. Dies ist beispielsweise der Fall, wenn der resultierende Einsatzplan vom Verbundsprecher an eine externe Entität zur Netzverträglichkeitsprüfung übermittelt werden muss, bevor er umgesetzt werden darf. Die Terminierungserkennung eines verteilten Verfahrens ist allerdings nicht trivial, insbesondere wenn es sich wie im vorliegenden Fall um ein asynchrones System mit variabler Nachrichtenverzögerung handelt. In [\[Mar13\]](#) wurden hierzu verschiedene Verfahren untersucht. Dabei wurden die Algorithmen *Computation Tree* nach [\[SF86\]](#), *Static Tree* nach [\[MD07\]](#) und *Splitting-Vektor* nach [\[Mat89\]](#) als geeignete Verfahren

für das vorliegende System identifiziert, anschließend prototypisch umgesetzt und schließlich hinsichtlich ihrer Effizienz bewertet. Im Ergebnis konnten prinzipiell alle drei Verfahren überzeugen, wobei sie in Bezug auf die Kriterien Nachrichtenkomplexität, Nachrichtengröße, Speicheraufwand sowie Verzögerung der Terminierungserkennung jeweils unterschiedliche Vor- und Nachteile besitzen. Insgesamt scheint die *Splitting-Vektor* Methode aufgrund ihrer geringen Nachrichtenkomplexität für das vorliegende System besonders geeignet zu sein und kann – wie in [Mar13] prototypisch gezeigt – bei Bedarf leicht realisiert werden.

4.4 VERHALTEN DER AGENTEN

Ein essenzieller Teil der Motivation für das in dieser Arbeit entwickelte Einsatzplanungsverfahren betrifft die Tatsache, dass die an einem Verbund teilnehmenden Akteure als Individuen mit privaten Interessen betrachtet werden müssen. Es ist daher zweckmäßig, genau zu spezifizieren, welche Anforderungen das durch das Einsatzplanungsverfahren induzierte Verhaltensprotokoll an das Verhalten der die Akteure repräsentierenden Agenten stellt. In [Abschnitt 3.2](#) wurde dazu als Vorbereitung auf den Entwurf des Einsatzplanungsverfahrens das prinzipielle Verhaltensmodell der Agenten vorgestellt. Dabei wurde auf die Punkte *Kollaboration*, *Wettbewerb* und *Fähigkeiten* eingegangen. Im Zuge des Systemdesigns in den Abschnitten [3.4](#) bis [3.9](#) wurde die Spezifikation des Agentenverhaltens sukzessive erweitert, bis es dem in diesem Kapitel beschriebenen Stand entsprach. Die darin enthaltenen Freiheiten und Pflichten eines Agenten sollen nun noch einmal strukturiert zusammengefasst werden. Im Zuge dessen wird die Aufstellung um weitere Details ergänzt, die in der bisherigen Diskussion nicht von Relevanz waren, für die nachfolgenden Kapitel aber von Interesse sind. Zusätzlich wird dabei auf mögliche Auswirkungen von Regelverletzungen bzgl. der verbleibenden Durchführbarkeit des Verfahrens eingegangen, um die Robustheit des Verfahrens gegen fehlerhafte oder böswillige Akteure zu diskutieren.

4.4.1 Pflichten

Lebendigkeit Das in dieser Arbeit entwickelte Verfahren ist nicht in der Lage, den Wegfall von Agenten aus dem betrachteten Verbund zu detektieren. Daher sind alle

Agenten verpflichtet, nach dem Start der Einsatzplanung bis zu deren Ende ihre Mitgliedschaft an dem Verbund aufrechtzuerhalten. Dies impliziert insbesondere, dass auch ein Anlagenausfall bzw. ein technischer Ausfall des sie repräsentierenden Agenten nicht detektiert werden kann (dies ist in asynchronen Netzen generell nicht möglich, siehe [FLP85]). Bei Eintritt eines solchen Ereignisses würde sich im einfachsten Fall die resultierende Planungsgüte vermindern, da mit dem Beitrag des nicht mehr vorhandenen Akteurs weitergeplant wird. Im schlimmsten Fall jedoch würde der Wegfall eines Agenten zu einem nicht mehr zusammenhängenden Overlay-Netz führen, wodurch die Konvergenz des Verfahrens nicht mehr möglich wäre und die Einsatzplanung nicht abgeschlossen werden könnte. Um diesen Effekt zu vermeiden, wäre ein fehlerdetektierendes Basis-Kommunikationsprotokoll erforderlich, welches bspw. über periodische Statusanfragen (engl. *heartbeats*) die Lebendigkeit der Verbundteilnehmer überwacht.

Sprecherrolle Wie zu Beginn dieses Kapitels erläutert, nimmt ein einzelner Verbundteilnehmer die Rolle des Verbundsprechers ein. Das vorgestellte Einsatzplanungsverfahren schreibt keine Methode vor, nach welcher diese Auswahl zu treffen ist, sondern geht von einem fertig gebildeten Verbund samt auserwähltem Verbundsprecher aus. Dieser Sprecher hat dann die Aufgabe, das Einsatzplanungsverfahren durch Bildung des Overlay-Netzes sowie Verteilung des Wirkleistungsproduktes anzustoßen. Wird diese Aufgabe nicht erfüllt, so kann die Planung nicht durchgeführt werden. Daher soll an dieser Stelle eine mögliche Alternative aufgezeigt werden: Statt einem Verbundteilnehmer die Rolle des Sprechers zu übertragen, kann diese Aufgabe auch von einer externen Entität übernommen werden. Wenn etwa davon ausgegangen werden kann, dass für die vorgelagerte Verbundbildung eine technische Plattform existiert, an der sich die potenziellen Verbundteilnehmer registrieren, so kann diese Plattform zugleich auch entsprechende Management-Funktionen bereitstellen, die für die gebildeten Verbünde die Aufgaben des postulierten Verbundsprechers ausführen und als Schnittstelle zu externen Systemen wie etwa Handelsplattformen fungieren.

Nachbarschaftserhalt Jeder Agent muss die durch das ihm mitgeteilte Overlay-Netz induzierten Nachbarschaftsbeziehungen einhalten. Wird diese Regel verletzt, indem ein Verbundteilnehmer Nachrichten an andere als die ihm zugeteilten Nach-

barn versendet, so würde im einfachsten Fall aufgrund der modifizierten Kommunikationstopologie die Effizienz des Verfahrens beeinflusst werden (siehe hierzu [Abschnitt 7.3.1](#)). Ein Ausnutzen dieses Effektes bspw. für Dienstverweigerungsattacken (engl. *Denial of Service*) kann vermieden werden, indem Verbundteilnehmer nur Nachrichten in Empfang nehmen, die von legalen Nachbarn stammen. Im schlimmsten Fall jedoch wäre durch die Modifikation der Nachbarschaftsbeziehungen das Overlay-Netz nicht mehr zusammenhängend, wodurch sich der gleiche Effekt wie bei Verletzung der Lebendigkeits-Eigenschaft oben ergeben würde. Im Gegensatz zu der Lebendigkeits-Eigenschaft kann eine mutwillige Unterbrechung des Overlay-Netzes jedoch nicht so einfach durch ein fehlerdetektierendes Basis-Kommunikationsprotokoll auf technischer Ebene erfasst werden. Stattdessen wäre eine übergeordnete, überwachende Instanz erforderlich (z. B. der Verbundsprecher), welche solche Auffälligkeiten detektieren und geeignete Maßnahmen ergreifen kann.

Informationserhalt Die mittels Nachrichtenübertragung empfangenen Informationen müssen vom Empfänger im Sinne der Informationssynthese an seine jeweiligen Nachbarn weitergegeben werden. Dabei dürfen weder Wirkleistungsprodukt noch von anderen Agenten erstellte Fahrplanauswahlen oder Lösungskandidaten modifiziert werden. Hält ein Agent Informationen zurück, so würde sich im einfachsten Fall die Konvergenzgeschwindigkeit des Verfahrens reduzieren. Dieser Fall tritt ein, wenn das Overlay-Netz bzgl. der durch den betreffenden Agenten verlaufenden Kommunikationspfade redundante alternative Pfade besitzt. Im schlimmsten Fall käme das Zurückhalten von Informationen einer Unterbrechung des Overlay-Netzes, und damit der Verletzung der Lebendigkeits-Eigenschaft wie oben beschrieben gleich. Gegenmaßnahmen bzgl. des Zurückhaltens von Nachrichten wurden bereits diskutiert. Eine Modifikation von Nachrichteninhalten würde ebenfalls im einfachsten Fall zur Reduktion der Konvergenzgeschwindigkeit führen. Dieser Fall tritt ein, wenn die Modifikationen nicht schwerwiegend sind (bspw. eine Veränderung von einzelnen Fahrplanauswahlen, ohne dabei die Zählervariablen zu modifizieren) und damit im weiteren Verlauf des Verfahrens von korrekten Informationen überschrieben werden. Im schlimmsten Fall jedoch würden sich die fehlerhaften Informationen ausbreiten und einen fehlerfreien Abschluss des Verfahrens verhindern (bspw. wenn ein perfekter, aber aus nicht realisierbaren Fahrplänen bestehender Lösungskandidat erzeugt wird). Ein Schutz gegen Informationsmanipulation kann etwa durch auf asynchroner

Verschlüsselung basierende Signierung von Informationen geschehen.

Lösungsbeitrag Jeder Agent muss mindestens einen Fahrplan als Lösungselement in das Verfahren einbringen. Käme ein Agent dieser Verpflichtung nicht nach, so könnte kein vollständiger Lösungskandidat erzeugt werden. Zusätzlich ist jeder Agent verpflichtet, den letztlich im finalen Lösungskandidaten nach Abschluss des Verfahrens enthaltenen eigenen Fahrplan im Erfüllungszeitraum auszuführen. Somit sind alle von einem Agenten im Laufe des Verfahrens veröffentlichten eigenen Fahrpläne als potenziell verpflichtend anzusehen. Dies hat zwar keine unmittelbaren Auswirkungen auf den korrekten Abschluss der Einsatzplanung, würde bei Nichteinhaltung im Erfüllungszeitraum jedoch zu einer unvollständigen Produktlieferung führen. Es ist daher durch geeignete Maßnahmen (wie bspw. leistungsorientierte Verteilung des resultierenden Erlöses des Wirkleistungsproduktes) sicherzustellen, dass die aus dem Einsatzplanungsverfahren resultierenden Fahrpläne von den Akteuren im Erfüllungszeitraum tatsächlich umgesetzt werden.

Lösungsfortschritt Jeder Agent ist verpflichtet, im Rahmen der Entscheidungsfindung ausgewählte eigene Fahrpläne bzgl. der dann sichtbaren Systemkonfiguration gegen die globale Zielfunktion zu evaluieren und, sofern eine höhere als die bisher bekannte Güte erreicht werden kann, daraus neue Lösungskandidaten zu erzeugen und im Sinne der Informationssynthese den jeweiligen Nachbarn mitzuteilen. Eine Nichteinhaltung dieser Verpflichtung würde im einfachsten Fall (d. h. wenn nur wenige Agenten vereinzelt dagegen verstoßen) keine oder nur geringe Auswirkungen auf die Konvergenzgeschwindigkeit oder die finale Lösungsgüte haben. Mit zunehmender Anzahl der Verstöße würden sich die Auswirkungen statistisch betrachtet immer stärker auswirken. Im schlimmsten Fall würden gar keine Lösungskandidaten erzeugt werden und das Verfahren würde ohne Lösung terminieren. Gegen diesen Effekt kann das Verfahren nicht abgesichert werden, da es sich in diesem Fall um eine kollektive Verweigerung der Lösungsfindung handeln würde.

4.4.2 Freiheiten

Freie Gesinnungswahl Es ist jedem Agenten gestattet, im Rahmen der Entscheidungsfindung die eigenen Fahrpläne zusätzlich zur globalen Zielfunktion hinsichtlich

privater Kriterien zu bewerten. Diese Kriterien dürfen bzgl. der globalen Zielfunktion beliebig stark gewichtet werden. Auf diese Weise kann ein Agent seinen Lösungsraum beliebig stark einschränken, sofern mindestens ein realisierbarer Fahrplan in das Verfahren eingebracht wird (siehe Absatz *Lösungsbeitrag* oben). Der resultierende Effekt hieraus ist, dass mit zunehmenden Einschränkungen der Lösungsräume von Agenten die insgesamt erreichbare Lösungsgüte der Einsatzplanung verringert wird. Diesem Verhalten kann durch Anreizsysteme (bspw. beitragsorientierte Prämien aus dem resultierenden Erlös des Wirkleistungsproduktes) entgegengewirkt werden.

Private Wissenspeicher Die in der Einsatzplanungsheuristik COHDA enthaltenen Verhaltensregeln sehen vor, dass Agenten erhaltene Informationen bzgl. anderer Agenten in ihrem jeweiligen Arbeitsgedächtnis ablegen und für den Lösungsfortschritt sinnvoll nutzen. Während die Aktualisierung des Arbeitsgedächtnisses in den Verhaltensregeln genau spezifiziert ist, steht es den Agenten jedoch frei, die im Laufe des Verfahrens gesammelten Informationen in ihrem privaten Wissenspeicher dauerhaft abzulegen (vgl. [Abschnitt 3.6.4](#)). Jeder Agent stimmt bei Teilnahme an dem Verfahren somit zu, dass die von ihm veröffentlichten Informationen von anderen Agenten erfasst und abgelegt werden können. Eine Unterbindung der Informationsspeicherung ist in dem betrachteten System nicht möglich. Da die Agenten aber in der Regel nur einen sehr kleinen Teil ihrer Lösungsräume veröffentlichen, die betreffenden Informationen stark situationsabhängig sind, und sich die in einem gemeinsamen Verbund befindlichen Akteure zur kooperativen Zusammenarbeit entschlossen haben, sollte dies aus datenschutzrechtlichen Gründen unbedenklich sein.

Unabhängige Aktionsgeschwindigkeit Das für das betrachtete System modellierte Kommunikationsnetz wurde als potenziell verzögerungsbehaftet beschrieben (siehe [Definition 3.3.1](#)). Aus diesen möglichen Verzögerungen ergibt sich, dass die Reaktionen der Agenten auf an sie verschickte Nachrichten nicht unmittelbar, sondern ggf. erst zu einem unbekannten späteren Zeitpunkt erfolgen. Das entworfene Einsatzplanungsverfahren wurde demzufolge als asynchrones Verfahren gestaltet, in welchem die relativen Aktionszeitpunkte der Akteure zueinander irrelevant sind, solange keine Nachrichten unendlich lange verzögert werden (was einem Nachrichtenverlust gleichkäme). Die Asynchronität des Verfahrens kann nun ebenso als Freiheit

für die Agenten aufgefasst werden, welche ihnen erlaubt, in ihrer jeweils eigenen Geschwindigkeit zu handeln. Es ist den Agenten somit gestattet, die für die Konsensbildung erforderlichen Aktionen wie Informationsauswertung, Fahrplanberechnung usw. in beliebigen Geschwindigkeiten auszuführen, sofern sie nicht unendlich lange verzögert werden (was einer Verletzung der Lebendigkeit gleichkäme, siehe oben). Dies vereinfacht insbesondere die Realisierbarkeit des Verfahrens auf technischer Ebene, da keine Synchronisierungspunkte oder Zeitabgleiche im Betrieb erforderlich werden. Zudem wird hierdurch die in [Abschnitt 2.3.2](#) beschriebene *Variation* in das System eingeführt, was im günstigsten Fall zu einer Beschleunigung der Koordination führen kann. Welche Auswirkungen dies tatsächlich auf Performanz und Effizienz des Verfahrens hat, wird in [Kapitel 7](#) untersucht.

Insgesamt wird damit deutlich, dass das Verfahren gegen Regelverstöße einzelner Akteure bereits robust ist oder mittels der beschriebenen Maßnahmen abgesichert werden kann. Grundsätzlich wurde das Verfahren jedoch als kooperative Lösungsfindung gestaltet, welche auf einem zielgerichteten Zusammenschluss von individuellen Akteuren beruht. Zugleich wurde dabei aber beachtet, dass kein Akteur *altruistisch* handeln muss. Es wird somit von keinem Verbundteilnehmer erwartet, zum Wohle der anderen Akteure eigene Nachteile in Kauf zu nehmen. Verbundteilnehmer können daher selbst bestimmen, welchen Lösungsbeitrag sie in das Verfahren einbringen möchten. Hierdurch wird die grundsätzliche Stabilität des Verfahrens nicht beeinträchtigt. Um jedoch zu einem hinsichtlich des Wirkleistungsproduktes möglichst optimalen Einsatzplan zu gelangen, sollten die Akteure möglichst viel Flexibilität aufbringen. Da die Erbringung des Wirkleistungsproduktes im Erfüllungszeitraum entsprechend vergütet wird, ist die freiwillige Bereitstellung von Flexibilität hier leicht über Anreizsysteme zu realisieren. Dies soll an dieser Stelle nicht vertieft werden, da die monetären Rahmenbedingungen dem übergeordneten System und weniger dem Einsatzplanungsverfahren zuzuordnen sind. Beispielsweise wird in [\[BSA11\]](#) ein auf spieltheoretischen Verfahren basierender Verteilungsschlüssel für monetäre Erlöse vorgeschlagen, welcher neben einer direkten beitragsabhängigen Verteilung auch die Kooperativität sowie Flexibilität von Akteuren auf der Basis von historischen Erfahrungswerten in die Verteilung einfließen lässt. Ein solcher Verteilungsschlüssel kann, wie in der referenzierten Quelle beschrieben, bereits im Prozess der Verbundbildung als Entscheidungskriterium verwendet werden.

4.5 FAIRNESS

Die im vorigen Abschnitt durch die Kombination der Eigenschaften *Lebendigkeit* und *unabhängige Aktionsgeschwindigkeit* geforderte „endliche Asynchronität“ kann auch durch das Konzept der *Fairness* aus dem Gebiet der verteilten Algorithmen ausgedrückt werden. Hierbei sind je nach Kontext unterschiedliche Definitionen gebräuchlich, wobei bis auf seltene Ausnahmen grundsätzlich über das Verhalten eines Algorithmus „im Unendlichen“ argumentiert wird. So schreibt bspw. Francez in [Fra86, S. 3]:

The most basic context [...] is that of a *repetitive choice among alternatives*. In this context, fairness means that having to repeatedly choose among a set of alternatives, no alternative will be postponed forever.

Auf dieser Basis kann der Fairness-Begriff in verschiedene Klassen unterteilt werden, die jeweils eigene Nebenbedingungen an die obige Aussage knüpfen. Im Wesentlichen werden dabei drei Klassen betrachtet [AO83; Fra86; HRY91; Tel00, u. a.]:

Definition 4.5.1 (Bedingungslose Fairness)

Wenn in einer unendlich langen Ausführung jedes Ereignis unendlich oft auftritt, wird dies als *bedingungslose Fairness* bezeichnet.

Ein Beispiel hierfür ist die Ausführung von nebenläufigen, voneinander unabhängigen Prozessen auf einer Einprozessormaschine. Bedingungslose Fairness liegt dann vor, wenn jeder Prozess in unendlich vielen Zeitschritten unendlich oft Prozessorzeit zugewiesen bekommt. Wichtig hierbei ist, dass keine weiteren Anforderungen an die Wartezeit zwischen den Aktivierungen einzelner Prozesse, oder an die Länge der zugewiesenen Prozessorzeit gestellt werden.

Definition 4.5.2 (Schwache Fairness)

Seien Ereignisse an Bedingungen geknüpft, und sei in einer unendlich langen Ausführung irgendwann die Bedingung für das Auftreten eines Ereignisses erfüllt. *Schwache Fairness* liegt dann vor, wenn unter der Voraussetzung, dass die Bedingung solange erfüllt bleibt bis das Ereignis eingetreten ist, das Ereignis in endlich vielen Schritten auch tatsächlich eintritt.

Ein Beispiel hierfür ist eine Prozess-Warteschlange: Schwache Fairness garantiert, dass jeder Prozess, der sich in die Warteschlange einreicht und auf Aktivierung wartet, nach endlich langer Zeit auch tatsächlich aktiviert wird.

Definition 4.5.3 (Starke Fairness)

Seien Ereignisse an Bedingungen geknüpft. Wenn die Bedingung für ein Ereignis in einer unendlich langen Ausführung unendlich oft (aber nicht notwendigerweise in konsekutiven Zeitschritten) erfüllt ist, und das Ereignis nach endlich vielen Schritten tatsächlich eintritt, dann wird dies als *starke Fairness* bezeichnet.

Ein Beispiel hierfür sind kritische Bereiche in nebenläufigen Programmen: Starke Fairness garantiert, dass wenn die Bedingung zum Betreten eines kritischen Bereiches unendlich oft erfüllt ist, jeder gewillte Prozess diesen irgendwann betreten kann.

Weitere Fairness-Klassen weiten die an sie geknüpften Nebenbedingungen auf quantitative Merkmale aus. Beispielsweise führt Francez die Klasse *Equifairness* ein.

Definition 4.5.4 (Equifairness)

Sei B eine Menge von Ereignissen, deren Auftreten alle an die gleiche Bedingung gebunden sind. Gilt nun für eine unendlich lange Ausführung, dass es unendlich oft vorkommt, dass alle Ereignisse aus B im bisherigen Verlauf gleich oft eingetreten sind, dann liegt *Equifairness* vor.

Die oben geforderte Lebendigkeitseigenschaft von Agenten entspricht hier am ehesten der schwachen Fairness: Die Bedingung für die Aktivität eines Agenten ist das Eintreffen einer Nachricht. Die schwache Fairness schreibt vor, dass sobald diese Bedingung erfüllt ist, also sobald eine Nachricht an einen Agenten gesendet wurde, der Agent nach endlich vielen Schritten auch tatsächlich aktiv wird und die Nachricht bearbeitet.

Im Verlauf dieser Arbeit wurde bisher nur informell argumentiert, dass das Verfahren zu einem eindeutigen Konsens konvergiert, und damit terminieren kann. Im folgenden Kapitel wird daher näher auf die formalen Konvergenzeigenschaften der Heuristik eingegangen. Dabei wird auch auf die hier vorgestellten Fairness-Begriffe nochmals Bezug genommen.

5

KONVERGENZ DES VERFAHRENS

In [Abschnitt 3.6.3](#) wurde bereits informell auf die Konvergenz des vorgestellten selbstorganisierenden Einsatzplanungsverfahrens eingegangen. Im Folgenden soll nun eine formale Betrachtung der Konvergenz erfolgen. Hierzu wird das Verhalten eines Verbundteilnehmers in Anlehnung an die Eingabe-/Ausgabe Automaten (engl. *I/O-Automata*) aus [[Lyn96](#)] in Pseudocode beschrieben. Auf dieser Basis wird dann mittels der Methode der Konvergenzstufen (engl. *Convergence Stairs*) nach [[Dol00](#)] gezeigt, dass das Verfahren gegen eine finale Lösung konvergiert. Dabei bedeutet Konvergenz, dass das Verfahren terminiert, wenn sich alle Verbundteilnehmer auf eine gemeinsame Lösung geeinigt haben. Als final wird eine Lösung bezeichnet, wenn sie von keinem Verbundteilnehmer weiter verbessert werden kann. Um die einzelnen Schritte zu erläutern, wird in diesem Abschnitt wiederholt auf die in [Kapitel 3](#) eingeführten Definitionen und Beschreibungen Bezug genommen.

Aufgrund der vergleichsweise schwierig zu erfassenden Natur asynchroner Systeme mit potenziell fehlerbehafteten Kommunikationsnetzen wird im nachfolgenden [Abschnitt 5.1](#) zunächst eine vereinfachte Variante des Verfahrens betrachtet, in welcher von synchronem Verhalten und einem perfekten Kommunikationsnetz ausgegangen wird. Die Synchronität des Verhaltens wird durch einen globalen Planer modelliert, welcher den Verbundteilnehmern in wohldefinierter Reihenfolge die Ausführung ihres Verhaltens erlaubt. Dabei erfolgt die Zuweisung der Ausführungszeiten explizit im Rundlauf-Verfahren, sodass alle Verbundteilnehmer abwechselnd und mit der gleichen Häufigkeit aktiv werden. Dies ist eine Verschärfung der im vorigen Kapitel beschriebenen Equifairness ([Definition 4.5.4](#)) und wird im Folgenden als *absolute Equifairness* bezeichnet. Das perfekte Kommunikationsnetz zeichnet sich dabei im Vergleich zu [Definition 3.3.1](#) durch die Eigenschaft aus, dass Nachrichten unmittelbar, d. h. ohne Verzögerung zugestellt werden. Unter diesen Voraussetzungen ist es möglich, das bisher verwendete *Message Passing Model* (d. h. die Verbundteilnehmer

interagieren mittels Nachrichtenaustausch) durch das *Shared Memory Model* zu ersetzen, in welchem Verbundteilnehmer die öffentlichen Attribute ihrer Nachbarn direkt einsehen können [AW04]. Es wird demnach vom Nachrichtenverkehr abstrahiert, um die Beweisführung zu vereinfachen. In [Abschnitt 5.2](#) wird der Beweis dann auf den ursprünglichen Fall der asynchronen Ausführung und verzögerungsbehafteter Kommunikation erweitert, und damit die Konvergenz auch bei schwächerer Fairnessanforderung nachgewiesen. In den Argumentationen zu beiden Ausführungsmodellen wird dabei die *freie Gesinnungswahl* der Verbundteilnehmer (siehe [Designschritt 4](#) in [Abschnitt 3.6](#)) der Übersichtlichkeit halber außer Acht gelassen, sodass alle Elemente des Suchraumes von dem jeweiligen Agenten als akzeptabel anerkannt werden. In [Abschnitt 5.3](#) wird dieser Aspekt ergänzt. Aussagen zur Konvergenzgeschwindigkeit und zur finalen Lösungsgüte werden abschließend in [Abschnitt 5.4](#) und [Abschnitt 5.5](#) gemacht.

5.1 SHARED MEMORY MODEL

Die Beweismethode der Konvergenzstufen beinhaltet die Erfüllung aufeinander aufbauender Prädikate $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_k$ mit $k > 1$, wobei jedes Prädikat $\mathcal{A}_{i+1}$ mit $1 \leq i < k$ eine Verfeinerung des Prädikates $\mathcal{A}_i$ ist. Verfeinerung ist dabei gleichzusetzen mit $\mathcal{A}_{i+1} \Rightarrow \mathcal{A}_i$ (siehe [Dol00, S. 34]). Das letzte Prädikat $\mathcal{A}_k$ beschreibt die insgesamt zu beweisende Aussage. Die Beweisführung geht nun in Stufen vor: Initial wird gezeigt, dass irgendwann das erste Prädikat $\mathcal{A}_1$ gilt. Dann wird gezeigt, dass unter der Voraussetzung der Gültigkeit von $\mathcal{A}_1$ immer ein Zustand erreicht wird, in welchem $\mathcal{A}_2$ gilt. Dies wird für alle vorhandenen Stufen $1 < i \leq k$ durchgeführt, bis schließlich das Erreichen von $\mathcal{A}_k$ dargelegt wurde.

Im vorliegenden Fall werden drei Stufen betrachtet. Die erste Stufe zeigt die Vervollständigung des Wissens der Verbundteilnehmer, sodass nach einer endlichen Zeitspanne die wahrgenommenen Systemzustände sowie Lösungskandidaten aller Verbundteilnehmer *vollständig* sind (vgl. [Definition 3.6.5](#) und [3.6.6](#)). Darauf folgend beschreibt die zweite Stufe die Ermittlung einer finalen Lösung bei einem beliebigen Verbundteilnehmer. Die dritte Stufe beinhaltet schließlich die Ausbreitung dieser finalen Lösung über den gesamten Verbund und damit die Terminierung.

5.1.1 Erste Konvergenzstufe

Das Prädikat $\mathcal{A}_1$ der ersten Konvergenzstufe ist wie folgt gegeben.

Theorem 1. *Nach einer endlichen Zeitspanne nach dem Start des Systems besitzen alle Verbundteilnehmer vollständige wahrgenommene Systemzustände sowie vollständige Lösungskandidaten, sodass gilt:*

$$\forall a_i \in A : |\Omega_i| = |A| \wedge |\gamma_i| = |A| \quad (5.1)$$

Beweis. Gemäß [Definition 3.6.5](#) ergänzt ein Verbundteilnehmer seinen wahrgenommenen Systemzustand kontinuierlich mit den Informationen seiner Nachbarn. Hierbei wird der wahrgenommene Systemzustand sowohl vergrößert, als auch aktualisiert. Da das als Kommunikationstopologie verwendete Overlay-Netz schwarmbildend ist (und damit zusammenhängend, vgl. [Definition 3.6.1](#)), besteht zwischen zwei Verbundteilnehmern im Graphen des Overlay-Netzes höchstens die Distanz $|A| - 1$. Unter Voraussetzung der oben eingeführten *absoluten Equifairness* erhält ein beliebiger Verbundteilnehmer somit im *worst-case* mittels der Informationssynthese (siehe [Abschnitt 3.6.3](#)) in höchstens $|A| \cdot (|A| - 1)$ Schritten Informationen über alle anderen Verbundteilnehmer, sodass spätestens nach dieser Zeitspanne obige Voraussetzung erfüllt ist. Die gleiche Argumentation gilt für die Vervollständigung der Lösungskandidaten nach [Definition 3.6.6](#). $\square$

5.1.2 Zweite Konvergenzstufe

Auf der zweiten Konvergenzstufe soll die Existenz eines finalen Lösungskandidaten betrachtet werden.

Definition 5.1.1 (Finaler Lösungskandidat)

Eine Lösung ist final, wenn sie von keinem Verbundteilnehmer durch eine veränderte eigene Fahrplanauswahl weiter verbessert werden kann. Eine solche Lösung sei im Folgenden mit γ^* bezeichnet.

Würde man also eine Repräsentation des Lösungsraumes konstruieren, in welcher sich benachbarte Lösungen durch genau eine abweichende Fahrplanauswahl eines Verbundteilnehmers unterscheiden, so würde die Menge der finalen Lösungen genau

der Menge der lokalen Optima in diesem Lösungsraum entsprechen. Somit kann es für ein gegebenes Einsatzplanungsproblem beliebig viele finale Lösungen geben. Im Folgenden soll nun gezeigt werden, dass in einer beliebigen Ausführung des Einsatzplanungsverfahrens nach endlich vielen Schritten eine dieser finalen Lösungen gefunden wird. Das von jedem Verbundteilnehmer ausgeführte Verhalten wird dazu als Pseudocode beschrieben. Unter Voraussetzung der Gültigkeit des Prädikates $\mathcal{A}_1$ kann die Vollständigkeit des Wissens eines Verbundteilnehmers dabei als erfüllt angesehen werden, sodass sich das in [Algorithmus 1](#) abgebildete Verhalten ergibt. In den ersten drei Abschnitten ist hier in Anlehnung an [[Lyn96](#)] die Signatur des Automaten dargestellt, beginnend mit den eigenen Attributen des Verbundteilnehmers mit Lese- und Schreibzugriff. Es folgen die nur-lesbaren Attribute anderer Verbundteilnehmer. Diese sind gemäß [Abschnitt 3.6.2](#) auf die direkte Nachbarschaft des Verbundteilnehmers beschränkt. Der Übersichtlichkeit halber beschreibt der dritte Abschnitt einige Hilfsfunktionen, welche im Folgenden verwendet werden. Auf die Definition dieser Funktionen wird hier nicht weiter eingegangen, da sie sich unmittelbar aus den Definitionen [3.6.3](#), [3.1.11](#), [3.6.6](#) und [3.6.4](#) ergeben.

Im vierten Abschnitt ist das Verhalten des Verbundteilnehmers dargestellt. Dieses spiegelt nach den oben eingeführten Vereinfachungen das in [Abschnitt 3.6](#) beschriebene Verhalten wider: Der Verbundteilnehmer observiert kontinuierlich die ihn umgebenden Verbundteilnehmer, hier durch die umfassende Schleife realisiert. Bei jedem Durchlauf aktualisiert der Verbundteilnehmer in den Zeilen [2](#) – [8](#) zunächst seinen wahrgenommenen Systemzustand (d. h. die Menge der aktuell ausgewählten Fahrpläne anderer Verbundteilnehmer, diese ist aufgrund Konvergenzstufe $\mathcal{A}_1$ bereits vollständig) gemäß [Definition 3.6.5](#) sowie in den Zeilen [9](#) – [13](#) seinen Lösungskandidaten gemäß [Definition 3.6.6](#). Anschließend prüft er in den Zeilen [14](#) – [16](#), ob er den aktuellen wahrgenommenen Systemzustand soweit verbessern kann, dass sich eine höhere Güte als die des derzeitigen (aktualisierten) Lösungskandidaten ergeben würde. Ist dies der Fall, wird der Lösungskandidat ersetzt. Die letzte Anweisung des Verhaltens in Zeile [17](#) beschreibt die Zuweisung des eigenen Fahrplans gemäß dem dann geltenden Lösungskandidaten.

Für das Prädikat $\mathcal{A}_2$ der zweiten Konvergenzstufe soll nun gezeigt werden, dass mit der Ausführung von [Algorithmus 1](#) nach endlich vielen Schritten eine finale Lösung γ^* von einem Verbundteilnehmer gefunden wird, sowie dass diese nicht

Algorithmus 1 Automat zur Fahrplanauswahl eines Verbundteilnehmers a_i

Eigene Attribute:

γ_i	▷ Lösungskandidat von a_i
θ_i	▷ Fahrplan von a_i
Ω_i	▷ Wahrgenommener Systemzustand von a_i

Fremde Attribute:

$\forall a_j \in N_i: \gamma_j$	▷ Lösungskandidaten der Nachbarn
$\forall a_j \in N_i: \Omega_j$	▷ Wahrgenommene Systemzustände der Nachbarn

Verwendete Funktionen:

$\text{TIME}(\theta)$	▷ Ermittelt Konstruktionszeitpunkt von θ
$f(\gamma)$	▷ Ermittelt Güte von γ
$\text{ID}(\gamma)$	▷ Ermittelt Ersteller von γ
$\text{GET}(\gamma, i)$	▷ Ermittelt Fahrplan von a_i in γ

Eigenes Verhalten:

```

1: loop

2:   for  $\Omega_j \in N_i$  do
3:     for  $\theta_k \in \Omega_i, \theta_k' \in \Omega_j$  do
4:       if  $\text{TIME}(\theta_k') > \text{TIME}(\theta_k)$  then
5:          $\Omega_i \leftarrow (\Omega_i \setminus \{\theta_k\}) \cup \{\theta_k'\}$ 
6:       end if
7:     end for
8:   end for

9:   if  $\exists a_j \in N_i: f(\gamma_j) > f(\gamma_i)$  then
10:     $\gamma_i \leftarrow \gamma_j$ 
11:   else if  $\exists a_j \in N_i: f(\gamma_j) = f(\gamma_i)$  and  $\text{ID}(\gamma_j) > \text{ID}(\gamma_i)$  then
12:     $\gamma_i \leftarrow \gamma_j$ 
13:   end if

14:   if  $\exists \theta \in S_i: f(\Omega \cup \{\theta\}) > f(\gamma_i)$  then
15:     $\gamma_i \leftarrow \Omega \cup \{\theta\}$ 
16:   end if

17:    $\theta_i \leftarrow \text{GET}(\gamma_i, i)$ 

18: end loop

```

wieder verloren geht und somit die Terminierung des Verfahrens einleiten kann.

Hierzu muss zunächst der Begriff der Ausführungsfolge in dem betrachteten verteilten System beschrieben werden. Der eingangs aufgrund der vorgenommenen Vereinfachung auf ein synchrones System eingeführte globale Planer erlaubt den einzelnen verteilten Prozessen (hier Verbundteilnehmer) nacheinander jeweils die Ausführung ihrer Verhaltensschleife aus [Algorithmus 1](#). Eine solche Schleifenausführung eines Verbundteilnehmers a_i beinhaltet stets genau einen Durchlauf der Schleife und sei mit e_i bezeichnet. Eine Ausführungsfolge c ist dann eine Folge von solchen Schleifenausführungen und beschreibt damit den Zustand des Systems zu einem bestimmten Zeitpunkt der Ausführung:

Definition 5.1.2 (Ausführungsfolge)

Sei $e_u, e_v, e_w, \dots$ eine Menge von unmittelbar aufeinander folgenden Schleifendurchläufen. Dann wird

$$c = (e_u, e_v, e_w, \dots) \quad (5.2)$$

als *Ausführungsfolge* bezeichnet. Über die konkrete Reihenfolge der aktiven Verbundteilnehmer $(u, v, w, \dots)$ in einer solchen Folge wird keine Annahme getroffen, solange sie die eingangs eingeführte *absolute Equifairness* bzgl. der Zuweisung der Ausführungszeiten erfüllt.

Das System terminiert genau dann, wenn kein Verbundteilnehmer in seiner Ausführung eine Zuweisung vornimmt.

Definition 5.1.3 (Zuweisungsfreiheit)

Sei e_i ein Schleifendurchlauf eines Verbundteilnehmers a_i . Der Schleifendurchlauf wird als *zuweisungsfrei* bezeichnet, wenn a_i in e_i keinem seiner eigenen Attribute einen neuen Wert zuweist.

Definition 5.1.4 (Terminierung)

Sei c eine Ausführungsfolge. Besitzt c ein Suffix c_s mit

$$\begin{aligned} |c_s| &= |A| \quad \text{und} \\ \forall e_i, e_j \in c_s : i &\neq j \quad \text{und} \\ \forall e_i \in c_s : e_i &\text{ ist zuweisungsfrei,} \end{aligned}$$

dann ist das Verfahren mit der Ausführung von c *terminiert*, und c wird als *terminierende Ausführungsfolge* bezeichnet.

Sei nun für eine beliebige Ausführungsfolge c mit

$$h(c) = \sum_{a_i \in A} f(\gamma_i) \quad (5.3)$$

die Summe der Gütewerte der aktuellen Lösungskandidaten aller Verbundteilnehmer nach Ausführung der in c enthaltenen Schleifenausführungen gegeben.

Lemma 1. *Das Verfahren besitzt die „anytime“ Eigenschaft, sodass der Funktionswert von h mit jeder möglichen Schleifenausführung e_i nicht kleiner wird:*

$$c' = c + e_i \Rightarrow h(c') \geq h(c) \quad (5.4)$$

Beweis. Es wird das in [Algorithmus 1](#) dargestellte Verhalten eines Verbundteilnehmers a_i betrachtet. Zu Beginn wird in Zeile 5 der wahrgenommene Systemzustand des Verbundteilnehmers mit Informationen aus den wahrgenommenen Systemzuständen seiner Nachbarn aktualisiert. Dies hat keine Auswirkungen auf den Funktionswert von h . In Zeile 10 wird der eigene Lösungskandidat γ_i mit einem fremden Lösungskandidaten γ_j ersetzt, sofern γ_j eine höhere Güte erzielt. Durch diese Aktion wird der Funktionswert von h also inkrementiert. In Zeile 12 wird ebenfalls der eigene Lösungskandidat γ_i mit einem fremden Lösungskandidaten γ_j ersetzt. Dies geschieht allerdings nur unter der Bedingung, dass ein fremder Lösungskandidat mit gleicher Güte, aber höherer Ersteller-ID gefunden wurde, und dient der Konsensbildung bei Lösungen gleicher Güte. Hierdurch bleibt der Funktionswert von h also unverändert. Auch in Zeile 15 wird der eigene Lösungskandidat γ_i ersetzt, diesmal jedoch mit einem *neuen* Lösungskandidaten, sofern ein solcher mit höherer Güte gebildet werden kann. Dies hat wiederum zur Folge, dass der Funktionswert von h inkrementiert wird, da der neue Lösungskandidat eine höhere Güte besitzt als der ursprüngliche. In Zeile 17 wird schließlich der eigene Fahrplan ersetzt, was jedoch keine Auswirkungen auf den Funktionswert von h hat. Insgesamt wird also mit jeder möglichen Ausführung des beschriebenen Verhaltens der Funktionswert von h entweder unverändert beibehalten oder inkrementiert, was die Gültigkeit von [Gleichung 5.4](#) bestätigt. $\square$

Lemma 2. *Das Verfahren ist verklemmungsfrei in dem Sinne, dass der Funktionswert von h höchstens endlich viele Schleifenausführungen gleichbleibend ist.*

Beweis. Der Funktionswert von h wird immer dann inkrementiert, wenn ein Lösungskandidat durch einen Lösungskandidaten mit besserer Güte ersetzt wird. Bei einer Schleifenausführung e_i eines beliebigen Verbundteilnehmers a_i ist der Funktionswert von h also nur in genau den folgenden Fällen gleichbleibend:

1. Der eigene Lösungskandidat γ_i wird unverändert beibehalten. Dies ist der Fall wenn es in der Nachbarschaft von a_i keinen zu bevorzugenden fremden Lösungskandidaten γ_j gibt und a_i nicht in der Lage ist, durch eine modifizierte Fahrplanauswahl einen noch besseren Lösungskandidaten zu bilden.
2. In der Nachbarschaft von a_i gibt es einen Lösungskandidaten mit gleicher Güte, aber größerer Ersteller-ID.

Der *worst-case* für beide Fälle tritt ein, wenn alle Verbundteilnehmer jeweils unterschiedliche Lösungskandidaten mit gleicher Güte besitzen. Dann können aufgrund der *absolut equifairen* Zuteilung der Ausführungszeiten höchstens $|A| - 1$ Verbundteilnehmer an der Reihe sein, für die Fall 1 gilt, denn es muss mindestens einen Verbundteilnehmer geben, der einen Nachbarn mit größerer ID besitzt. Daraus folgend muss es in jeweils $|A|$ Schleifenausführungen von verschiedenen Verbundteilnehmern aber auch immer mindestens eine geben, in welcher Fall 2 gilt, wodurch sich ein Lösungskandidat weiter über den Verbund ausbreitet. Dies geht solange, bis alle Verbundteilnehmer den gleichen Lösungskandidaten besitzen und das Verfahren terminiert, oder ein Verbundteilnehmer eine bessere Lösung findet. Somit ist gewährleistet, dass der Funktionswert von h nach endlichen vielen Schritten inkrementiert wird, solange das System nicht terminiert ist. $\square$

Das Prädikat $\mathcal{A}_2$ der zweiten Konvergenzstufe ist nun durch das folgende Theorem gegeben.

Theorem 2. *Nach endlich vielen Schleifenausführungen in [Algorithmus 1](#) findet ein Verbundteilnehmer eine Lösung γ^* , welche nachfolgend von keinem Verbundteilnehmer weiter verbessert werden kann und somit final ist.*

Beweis. Die Lösungskandidaten im beschriebenen Verfahren werden aus Kombinationen von Fahrplänen aus den Suchräumen S_i aller Verbundteilnehmer $a_i \in A$ gebildet. Der Lösungsraum des Optimierungsproblems ist daher durch das Kreuzprodukt aller Suchräume gegeben:

$$L = \prod_{a_i \in A} S_i \quad (5.5)$$

Da die Suchräume endlich sind (vgl. [Definition 3.1.10](#)), ist auch L endlich groß, und $|L|$ ist die Anzahl aller möglichen Lösungskandidaten. Mit [Lemma 1](#) und [Lemma 2](#) ist garantiert, dass das System solange immer bessere Lösungen erzeugt, bis das Terminierungskriterium aus [Definition 5.1.4](#) erfüllt ist. Wenn dieses erfüllt ist, impliziert das aber insbesondere, dass kein Verbundteilnehmer in der Lage ist, seinen derzeitigen Lösungskandidaten durch Ersetzung oder Neubildung weiter zu verbessern. Irgendwann muss also ein Verbundteilnehmer einen Lösungskandidaten γ^* finden, der nicht weiter verbessert werden kann. Im *worst-case* ist dies dann der Fall, nachdem bereits alle anderen $|L| - 1$ Kombinationsmöglichkeiten von Fahrplänen gebildet wurden. Da ein Lösungskandidat nur mit Lösungskandidaten höherer Güte überschrieben werden kann, bleibt γ^* bis zur Terminierung existent. $\square$

5.1.3 Dritte Konvergenzstufe

Auf der dritten Konvergenzstufe kann durch das vorhergehende Prädikat $\mathcal{A}_2$ vorausgesetzt werden, dass mindestens ein Verbundteilnehmer im Besitz eines finalen Lösungskandidaten γ^* ist.

Definition 5.1.5 (Besitz eines Lösungskandidaten)

Sei a_i ein Verbundteilnehmer und γ ein Lösungskandidat. Der Verbundteilnehmer a_i ist im *Besitz des Lösungskandidaten* γ , wenn gilt:

$$\gamma_i = \gamma$$

Der Besitz eines Lösungskandidaten bezieht sich somit auf die aktuelle Belegung des eigenen Attributes eines Verbundteilnehmers.

Sei zudem für eine beliebige Ausführungsfolge c mit $l(c)$ die Anzahl der Verbundteilnehmer gegeben, welche nach Ausführung von c im Besitz von γ^* sind. Dann

ist

$$VF(c) = |A| - l(c) \quad (5.6)$$

eine Potenzialfunktion (in [Dol00] als *Variant Function* bezeichnet) mit dem Wertebereich $[0, |A|]$. Das Prädikat $\mathcal{A}_3$ der dritten Konvergenzstufe ist nun durch das folgende Theorem gegeben.

Theorem 3. *Nachdem ein Verbundteilnehmer eine finale Lösung γ^* gefunden hat, terminiert das System in endlich vielen Schritten. Zum Zeitpunkt der Terminierung sind alle Verbundteilnehmer im Besitz von γ^* .*

Beweis. Sei c_t eine beliebige terminierende Ausführungsfolge mit

$$c_t = (\underbrace{e_a, e_b, \dots}_{c_p}, e_i, \underbrace{e_u, e_v, \dots}_{c_s}) \quad (5.7)$$

wobei c_p ein beliebiges Präfix ist und e_i einen Schleifendurchlauf bezeichnet, in welchem ein Verbundteilnehmer a_i den Lösungskandidaten γ^* durch Zuweisung an sein eigenes Attribut γ_i in Besitz nimmt. Mit $c_p' = c_p + e_i$ gilt dann $VF(c_p') < VF(c_p)$. Es ist jetzt zu zeigen, dass für jedes beliebige Suffix c_s die Potenzialfunktion aus Gleichung 5.6 monoton fallend ist, nach endlich vielen Schritten den Wert Null annimmt, und dass das System anschließend terminiert. Da e_i nicht zuweisungsfrei ist und das Verfahren gemäß Definition 5.1.4 erst dann terminiert, wenn auf einen nicht zuweisungsfreien Schleifendurchlauf noch mindestens $|A|$ zuweisungsfreie Schleifendurchläufe folgen, in denen jeder Verbundteilnehmer mindestens einmal enthalten ist, muss nach e_i also für jeden Verbundteilnehmer noch mindestens eine Schleifenausführung in dem terminierenden Suffix c_s enthalten sein. Dann können zwei Fälle unterschieden werden:

1. $VF(c_p') = 0$: Alle Verbundteilnehmer sind nach Ausführung von c_p' im Besitz des finalen Lösungskandidaten γ^* . Nach e_i folgen also noch $|A|$ weitere Schleifendurchläufe in c_s , welche alle zuweisungsfrei sind.
2. $VF(c_p') > 0$: Es muss mindestens eine nicht-zuweisungsfreie Schleifenausführung e_j in den ersten $|A|$ Elementen von c_s geben, für die gilt: a_j ist nicht im Besitz von γ^* und hat aber einen direkten Nachbarn, der im Besitz von γ^* ist. Aufgrund von Algorithmus 1 muss a_j diesen Lösungskandidaten ebenfalls

in Besitz nehmen. Sei $c_t' = c_p' + c_s'$ mit $c_s' = (\dots, e_j) \subseteq c_s$ die resultierende Ausführungsfolge bis inklusive e_j . Dann ist $VF(c_t') < VF(c_t)$, und der Wert der Potenzialfunktion wurde in höchstens $|A|$ Schleifendurchläufen nach e_i dekrementiert.

Da e_j nicht zuweisungsfrei ist, müssen durch die Terminierungsbedingung nach Ausführung von c_t' noch mindestens $|A|$ weitere Schleifendurchläufe folgen, und die vorliegende Fallunterscheidung muss für den verbleibenden Suffix erneut evaluiert werden.

Diese rekursive Argumentation zeigt also, dass solange $VF(c) > 0$ für eine beliebige Ausführungsfolge c gilt, in jeweils maximal $|A|$ auf c folgenden Schleifendurchläufen der Funktionswert von VF um eins dekrementiert wird. Sobald $VF(c) = 0$, folgen noch $|A|$ zuweisungsfreie Schleifendurchläufe und das Verfahren terminiert. Das Verfahren benötigt damit ab dem Zeitpunkt, zu dem ein finaler Lösungskandidat gefunden wird, im *worst-case* maximal $|A| \cdot (|A| + 1)$ weitere Schleifendurchläufe bis zur Terminierung. $\square$

5.2 MESSAGE PASSING MODEL

Für das vereinfachte Modell mit synchroner Ausführung sowie perfekter Kommunikation wurde im vorigen Abschnitt gezeigt, dass das System gegen eine finale Lösung konvergiert und dann terminiert. Im Folgenden sollen die dabei durchgeführten Argumentationsschritte auf ihre Gültigkeit im ursprünglichen *Message Passing Model*, also im Fall der asynchronen Ausführung mit verzögerungsbehaftetem Kommunikationsnetz, überprüft werden. Hierzu wird das Modell zunächst auf ein asynchrones Modell und Kommunikation mit fester Nachrichtenverzögerung erweitert. Später wird die Argumentation dann auf Kommunikationsnetze mit variabler Verzögerung ausgeweitet.

5.2.1 Asynchrones Modell mit fester Nachrichtenverzögerung

Wenn statt einem synchronen Ausführungsmodell mit *absolut equifairer* Zuweisung der Ausführungszeiten von einem asynchronen Modell ausgegangen wird, muss für jeden Verbundteilnehmer separat betrachtet werden, mit welcher Häufigkeit er aktiv

wird. Es ist somit nicht mehr gewährleistet, dass in $|A|$ Schleifenausführungen jeder Verbundteilnehmer genau einmal an der Reihe ist. Stattdessen muss zu Argumentationszwecken ein globaler Zeitbegriff in das System eingeführt werden. Hierzu ist es zweckmäßig, zunächst von einem Kommunikationsnetz auszugehen, in welchem Nachrichten eine fixe Zeitspanne $t_m > 0$ benötigen, um zugestellt zu werden. Dann kann die Realzeit mit der Auflösung t_m diskretisiert werden, um so den Begriff des Zeitschrittes zu definieren.

Definition 5.2.1 (Zeitschritt)

Eine *Zeitschritt* s umfasst im betrachteten System die kleinste Zeitspanne der Realzeit, die eine beliebige Nachricht benötigt, um in dem gegebenen Kommunikationsnetz zugestellt zu werden. Ein Schleifendurchlauf eines Verbundteilnehmers wird grundsätzlich im selben Zeitschritt abgeschlossen, in dem er begonnen wurde.

Es wird somit davon ausgegangen, dass die Berechnungen der Verbundteilnehmer im Vergleich zur Nachrichtenübertragung deutlich schneller ablaufen. Diese Annahme ist hier zulässig, da der entgegengesetzte Fall mittels variabler Nachrichtenverzögerungen abgebildet werden kann und im nachfolgenden [Abschnitt 5.2.2](#) behandelt wird. Die während der Schleifenausführungen verursachten Aktualisierungen der Attribute der Verbundteilnehmer werden per Nachricht den jeweiligen Nachbarn mitgeteilt. Da diese Nachrichten per obiger Definition erst im folgenden Zeitschritt zugestellt werden folgt daraus, dass alle Schleifendurchläufe als *parallel* betrachtet werden müssen, die im selben Zeitschritt ausgeführt werden.

Korollar 1. *Die im selben Zeitschritt stattfindenden parallelen Schleifenausführungen können sich nicht gegenseitig beeinflussen. Dies ergibt sich aus der Tatsache, dass Beeinflussungen ausschließlich mittels ausgetauschter Nachrichten erfolgen können, und eine Nachricht aber genau einen Zeitschritt zur Zustellung benötigt.*

Daraus ergibt sich die erweiterte Definition einer Ausführungsfolge:

Definition 5.2.2 (Parallele Ausführungsfolge)

Sei $ts = (s_i, s_j, s_k, \dots)$ eine Menge von unmittelbar aufeinander folgenden Zeitschritten. Dann finden zu jedem Zeitschritt $s_i \in ts$ potenziell mehrere parallele Schleifendurchläufe $E_i = \{e_u, e_v, \dots\}$ statt, wobei $\forall u, v \in E_i : u \neq v$. Da jede abge-

schickte Nachricht gemäß [Definition 5.2.1](#) im folgenden Zeitschritt zugestellt wird und dabei einen Schleifendurchlauf bei dem jeweiligen Empfänger verursacht, gilt $\forall i: |E_i| \geq 1$. Unter diesen Voraussetzungen wird

$$c = (E_i, E_j, E_k, \dots) \quad (5.8)$$

als (*parallele*) *Ausführungsfolge* bezeichnet.

Weiterhin wird die Annahme getroffen, dass zwischen je zwei Schleifendurchläufen eines beliebigen Verbundteilnehmers eine endliche Anzahl von Zeitschritten liegt, solange das System nicht terminiert ist. Die durch die im vorigen Abschnitt verwendete explizite Zuweisung von Ausführungszeiten im Rundlauf-Verfahren erzwungene *absolute Equifairness* wird somit also auf *schwache Fairness* gemäß [Definition 4.5.2](#) abgeschwächt und entspricht damit dem in [Abschnitt 4.5](#) geforderten Konzept der „endlichen Asynchronität“. An dieser Stelle sei bemerkt, dass eine solche Fairness im *Message Passing Model* mit autonomen Akteuren nicht erzwungen, sondern lediglich als Voraussetzung formuliert werden kann (vgl. [Abschnitt 4.4.1](#)).

Definition 5.2.3 (Periode)

Die größte Anzahl von Zeitschritten ρ , die zwischen zwei beliebigen Schleifendurchläufen eines beliebigen Verbundteilnehmers liegt, wird als *Periode* des Systems bezeichnet.

Daraus folgt insbesondere, dass innerhalb einer Periode (d. h. innerhalb von ρ Zeitschritten) jeder Verbundteilnehmer mindestens einmal aktiv wird.

Insgesamt ergeben sich folgende Änderungen zum *Shared Memory Model*:

1. Statt über $|A|$ Schleifendurchläufe zu argumentieren, in denen jeder Verbundteilnehmer jeweils *genau* einmal an der Reihe ist, muss nun die Zeitspanne von $\rho \cdot |A|$ Zeitschritten betrachtet werden, in denen jeder Verbundteilnehmer *mindestens* einmal an der Reihe ist.
2. Alle Schleifendurchläufe, die im selben Zeitschritt ausgeführt werden, sind als parallel zu betrachten.

Punkt 1 wirkt sich auf die Argumentation in [Abschnitt 5.1](#) wie folgt aus: Im Beweis zu [Theorem 1](#) wird der *worst-case* von $|A| \cdot (|A| - 1)$ Schritten (d. h. Schleifendurch-

laufen) zur Erfüllung des Theorems ermittelt. Dies muss durch $\rho \cdot |A| \cdot (\rho \cdot |A| - 1)$ Zeitschritte ersetzt werden. Da ρ eine endlich große Zahl ist, behält die Argumentation ansonsten ihre Gültigkeit. In den Beweisen zu [Theorem 2](#) und [Theorem 3](#) wird ähnlich vorgegangen. Hier muss an jeder Stelle, an der über die Anzahl von Schleifendurchläufe argumentiert wird, der Wert $|A|$ durch $\rho \cdot |A|$ ersetzt werden. Zudem muss [Definition 5.1.4](#) durch folgende Definition ersetzt werden:

Definition 5.2.4 (Periodenabhängige Terminierung)

Sei c eine Ausführungsfolge, und sei ρ die Periode des Systems. Besitzt c ein Suffix c_s mit

$$|c_s| = \rho \cdot |A| \quad \text{und} \\ \forall e_i \in c_s : e_i \text{ ist zuweisungsfrei,}$$

dann ist das Verfahren mit der Ausführung von c *terminiert*, und c wird als *terminierende Ausführungsfolge* bezeichnet.

Dies genügt, um die Gültigkeit der Argumentationen unter Berücksichtigung von Punkt 1 der obigen Auflistung zu gewährleisten. Für Punkt 2 muss der Begriff der Ausführungsfolge aus [Definition 5.1.2](#) auf parallele Schleifenausführungen gemäß obiger [Definition 5.2.2](#) erweitert werden. Da sich die im selben Zeitschritt stattfindenden Schleifenausführungen nach [Korollar 1](#) nicht gegenseitig beeinflussen können, müssen die auf Ausführungsfolgen basierenden Argumentationen in [Abschnitt 5.1](#) in ihrer Struktur nicht verändert werden und behalten prinzipiell ihre Gültigkeit. Lediglich formal muss an den entsprechenden Stellen über Mengen von Schleifendurchläufen statt über einzelne Elemente argumentiert werden.

5.2.2 Asynchrones Modell mit variabler Verzögerung

Das asynchrone Modell mit variabler Verzögerung zeichnet sich im Vergleich zu dem im vorigen Abschnitt betrachteten Modell dadurch aus, dass Nachrichten mit einer variablen Verzögerung zugestellt werden, sodass verschiedene Nachrichten unterschiedliche Verzögerungszeiten besitzen können. Eine Voraussetzung hierbei ist, dass diese Verzögerungen endlich sind, d. h. dass keine Nachrichten auf dem Kommunikationsweg verloren gehen. Das Auftreten von variablen Verzögerungen

wirkt sich auf die vorangegangene Argumentation insofern aus, als dass die Definition der Periode wie folgt modifiziert werden muss:

Definition 5.2.5 (Verzögerungsabhängige Periode)

Sei $d_{\max}$ die maximale Anzahl an Zeitschritten, die bei einem gegebenen Kommunikationsnetz als Verzögerung für das Zustellen einer Nachricht auftreten kann, und sei $r_{\max}$ die größte Anzahl von Zeitschritten, die zwischen zwei beliebigen Schleifendurchläufen eines beliebigen Verbundteilnehmers liegt. Dann wird $\rho = d_{\max} \cdot r_{\max}$ als *Periode* des Systems bezeichnet.

Diese Modifikation genügt, um die Gültigkeit des Konvergenzbeweises auch bei einem verzögerungsbehafteten Kommunikationsnetz zu gewährleisten. Die zuvor eingeführte *schwache Fairness* bzgl. der Aktionszeitpunkte der Verbundteilnehmer (in der obigen Definition durch $r_{\max}$ repräsentiert) muss dabei weiterhin als Voraussetzung angenommen werden.

5.3 KONVERGENZ BEI FREIER GESINNUNGSWAHL

In den vorigen Abschnitten wurde die Konvergenzbetrachtung ohne Bezug zu der in [Abschnitt 3.6](#) eingeführten freien Gesinnungswahl der Verbundteilnehmer durchgeführt. Dies soll nun ergänzt werden.

Für die erste Konvergenzstufe ergeben sich keine Änderungen, da jeder Verbundteilnehmer zu Beginn des Verfahrens verpflichtet ist, unabhängig von seiner Gesinnung eine initiale Fahrplanauswahl zu treffen. Die Gültigkeit von [Theorem 1](#) ist also weiterhin gegeben. Für die zweite Konvergenzstufe wird darauf aufbauend wiederum das Verhalten eines Verbundteilnehmers als Pseudocode beschrieben, dieser ist in [Algorithmus 2](#) abgebildet. Im Vergleich zu [Algorithmus 1](#) wurde lediglich Zeile 14 angepasst. Hier wurde die Klausel $\text{ACCEPT}(\theta)$ hinzugefügt, sodass eine neue Fahrplanauswahl nur dann gebildet wird, wenn sie den aktuellen Lösungskandidaten verbessern würde und zugleich eine für den Verbundteilnehmer in Bezug auf seine lokalen Präferenzen akzeptable Lösung darstellt. Diese einfache Modifikation zeigt deutlich, dass die freie Gesinnungswahl ausschließlich das Bilden von *neuen* Lösungskandidaten beeinflussen kann. Dies wiederum wirkt sich lediglich auf die potenziell erreichbare globale Güte eines finalen Lösungskandidaten γ^* aus, da ggf. nicht mehr

Algorithmus 2 Automat zur Fahrplanauswahl eines Verbundteilnehmers a_i bei freier Gesinnungswahl

Eigene Attribute:

γ_i	▷ Lösungskandidat von a_i
θ_i	▷ Fahrplan von a_i
Ω_i	▷ Wahrgenommener Systemzustand von a_i

Fremde Attribute:

$\forall a_j \in N_i: \gamma_j$	▷ Lösungskandidaten der Nachbarn
$\forall a_j \in N_i: \Omega_j$	▷ Wahrgenommene Systemzustände der Nachbarn

Verwendete Funktionen:

$\text{TIME}(\theta)$	▷ Ermittelt Konstruktionszeitpunkt von θ
$f(\gamma)$	▷ Ermittelt Güte von γ
$\text{ID}(\gamma)$	▷ Ermittelt Ersteller von γ
$\text{GET}(\gamma, i)$	▷ Ermittelt Fahrplan von a_i in γ
$\text{ACCEPT}(\theta)$	▷ TRUE, wenn Fahrplan θ akzeptabel

Eigenes Verhalten:

```

1: loop

2:   for  $\Omega_j \in N_i$  do
3:     for  $\theta_k \in \Omega_i, \theta_k' \in \Omega_j$  do
4:       if  $\text{TIME}(\theta_k') > \text{TIME}(\theta_k)$  then
5:          $\Omega_i \leftarrow (\Omega_i \setminus \{\theta_k\}) \cup \{\theta_k'\}$ 
6:       end if
7:     end for
8:   end for

9:   if  $\exists a_j \in N_i: f(\gamma_j) > f(\gamma_i)$  then
10:     $\gamma_i \leftarrow \gamma_j$ 
11:   else if  $\exists a_j \in N_i: f(\gamma_j) = f(\gamma_i)$  and  $\text{ID}(\gamma_j) > \text{ID}(\gamma_i)$  then
12:     $\gamma_i \leftarrow \gamma_j$ 
13:   end if

14:   if  $\exists \theta \in S_i: f(\Omega \cup \{\theta\}) > f(\gamma_i) \wedge \text{ACCEPT}(\theta)$  then
15:     $\gamma_i \leftarrow \Omega \cup \{\theta\}$ 
16:   end if

17:    $\theta_i \leftarrow \text{GET}(\gamma_i, i)$ 

18: end loop

```

alle $|L|$ möglichen Kombinationen von Fahrplänen (siehe [Gleichung 5.5](#)) erreicht werden können. Die Argumentation bezüglich [Theorem 2](#) wird somit nicht berührt, und es kann weiterhin davon ausgegangen werden, dass ein finaler Lösungskandidat γ^* nach endlich vielen Zeitschritten gefunden wird.

Die dritte Konvergenzstufe wird von der freien Gesinnungswahl ebenso wenig berührt, da hier über die Ausbreitung des finalen Lösungskandidaten argumentiert wird, welche unabhängig vom potenziellen neuen Fahrplanauswahlen geschieht (da diese per Definition alle vom finalen Lösungskandidaten überschrieben werden).

Zusammenfassend beeinträchtigt die freie Gesinnungswahl die Konvergenz des Verfahrens also nicht. Es ist jedoch möglich, dass die damit verbundene Einschränkung der Menge von möglichen Fahrplankombinationen zu einer insgesamt geringeren globalen Güte des finalen Lösungskandidaten, und damit zu einer geringeren Erreichung des Wirkleistungsproduktes führt. Dies wird im Rahmen der Evaluation des Verfahrens in [Abschnitt 7.3.4](#) genauer betrachtet.

5.4 KONVERGENZGESCHWINDIGKEIT

Im Verlauf dieses Kapitels wurden bereits an den einigen Stellen Aussagen über die *worst-case* Komplexität der einzelnen Schritte gemacht. Dies soll nun noch einmal zusammengefasst und um Aussagen zum *best-case* ergänzt werden. Dabei wird wiederum zunächst von dem vereinfachten *Shared Memory Model* ausgegangen, und die Argumentation anschließend auf das *Message Passing Model* erweitert.

Sowohl die Vervollständigung der Lösungskandidaten (Prädikat $\mathcal{A}_1$ der ersten Konvergenzstufe) als auch die Ausbreitung eines finalen Lösungskandidaten vor der Terminierung (Prädikat $\mathcal{A}_3$ der dritte Konvergenzstufe) sind unmittelbar von der Topologie des verwendeten Overlay-Netzes abhängig. Sei $m = |A|$ die Größe des betrachteten Verbundes und $\delta_{\max}(O)$ die maximale Distanz zweier beliebiger Verbundteilnehmer im verwendeten Overlay-Netz O . Im *best-case* ist $\delta_{\max}(O) = 1$ (g.d.w. O vollvernetzt ist) und sowohl $\mathcal{A}_1$ als auch $\mathcal{A}_3$ werden in jeweils einem Schritt erreicht. Im *worst-case* ist $\delta_{\max}(O) = m - 1$ (g.d.w. O eine Liste ist) und $\mathcal{A}_1$ und $\mathcal{A}_3$ können erst nach $m \cdot (m - 1)$ Schritten erreicht werden (vgl. [Abschnitt 5.1.1](#)).

Das Finden des finalen Lösungskandidaten γ^* (und damit die Erfüllung des Prädikates $\mathcal{A}_2$ der zweiten Konvergenzstufe) ist der aufwändigste Teil des Verfahrens

und ist von einer Mehrzahl von Parametern abhängig (z. B. Topologie des Overlay-Netzes, Verbundzusammensetzung, initiale Fahrplanauswahl, Wahl des Sprechers). Im *best-case* ist bereits die initiale Lösung des Systems (d. h. die Kombination der initial ausgewählten Fahrpläne der Verbundteilnehmer) der finale Lösungskandidat. In diesem – sehr unwahrscheinlichen – Fall besitzen bereits mit der Erfüllung von $\mathcal{A}_1$ alle Agenten den finalen Lösungskandidaten und das Verfahren terminiert sofort. Im *worst-case* dagegen wird mit der schlechtest möglichen Lösung begonnen und dann sukzessive der gesamte Kombinationsraum der Fahrpläne durchsucht, und der finale Lösungskandidat erst ganz zum Schluss gebildet. Da der Kombinationsraum L (= Lösungsraum des Optimierungsproblems) gemäß [Gleichung 5.5](#) aus dem Kreuzprodukt aller lokalen Suchräume der Verbundteilnehmer besteht, variiert dessen Größe je nach Verbundzusammensetzung. Kann für die maximal auftretende Größe eines beliebigen Anlagensuchraums S eine Konstante n bestimmt werden, dann existiert eine obere Schranke für die Größe des Kombinationsraumes, aus der sich unmittelbar der *worst-case* Aufwand für $\mathcal{A}_2$ ergibt:

$$\forall a_i \in A : |S_i| \leq n \Rightarrow |L| \leq n^m \quad (5.9)$$

Während also insgesamt der *best-case* im *Shared Memory Model* in $\mathcal{O}(1)$ liegt, muss für den *worst-case* Aufwand hier $\mathcal{O}(n^m)$ angenommen werden.

In der Erweiterung auf das *Message Passing Model* mit fester Nachrichtenverzögerung wird die zuvor angenommene *absolute Equifairness* bzgl. der Zuweisung von Ausführungszeiten aufgelockert, und durch *schwache Fairness* ersetzt. Für die Betrachtung der Konvergenz muss dann gemäß [Abschnitt 5.2.1](#) zusätzlich die Periode ρ des Systems betrachtet werden. Für den Aufwand im *best-case* ergibt sich demnach

$$\mathcal{O}(\rho),$$

und für den *worst-case*

$$\mathcal{O}(n^{\rho \cdot m}).$$

Für die Erweiterung auf variable Nachrichtenverzögerungen verändert sich gemäß [Abschnitt 5.2.2](#) lediglich die Definition der Periode ρ , die sich nun aus den oberen Schranken für die mögliche Dauer der Zustellung einer Nachricht und der möglichen

zeitlichen Distanz zweier aufeinanderfolgender Schleifendurchläufe eines beliebigen Verbundteilnehmers zusammensetzt.

Da ρ in beiden Fällen eine endlich große Zahl ist und daher als konstanter Faktor in die Komplexitätsbetrachtung eingeht, ergibt sich insgesamt, dass die Zahl der benötigten Zeitschritte zur Terminierung des Verfahrens unabhängig vom betrachteten Ausführungsmodell im *best-case* konstant ist. Im *worst-case* wächst die zur Terminierung benötigte Zeit mit n^m exponentiell in Abhängigkeit der Verbundgröße. Bei COHDA handelt es sich um eine Heuristik, daher kann keine Garantie gegeben werden, dass der *worst-case* nie eintreten wird. Vermutlich ist die Wahrscheinlichkeit dafür jedoch in der Realität verschwindend gering. Aufgrund der prinzipiell unbekannten Laufzeitparameter des Verfahrens (Topologie des Overlay-Netzes, Verbundzusammensetzung, initiale Fahrplanauswahl, Wahl des Sprechers) kann an dieser Stelle keine generelle Aussage über den Aufwand im *average-case* getroffen werden. Daher wird das Verfahren in dieser Arbeit simulativ evaluiert, um zumindest für ausgewählte Szenarien statistische Aussagen über die Effizienz der Heuristik treffen zu können.

5.5 FINALE LÖSUNGSGÜTE

Abschließend sei zu diesem Kapitel bemerkt, dass *jedes lokale Optimum des betrachteten Optimierungsproblems einen möglichen finalen Lösungskandidaten γ^* darstellt*. Es kann somit keine untere Schranke für die Güte der von dem Verfahren produzierten finalen Lösung gegeben werden. Analog zur Effizienz (bzgl. Konvergenzgeschwindigkeit etc.) muss somit auch die Performanz des Verfahrens simulativ ermittelt werden.

6 | UMSETZUNG

Als Vorbereitung für eine simulationsbasierte Evaluation sowie als Hilfestellung im Entwicklungsprozess des Verfahrens wurde die in [Kapitel 4](#) vorgestellte Heuristik COHDA prototypisch in einem Softwaresystem implementiert. Im Folgenden werden einige Details dieser Umsetzung beschrieben, sofern sie für die in [Kapitel 7](#) durchgeführte Simulationsstudie von Relevanz sind. Abschließend wird in diesem Kapitel eine mögliche Umsetzung des Systems für reale Anlagen diskutiert.

6.1 SIMULATIONSUMGEBUNG

Da bei der Entwicklung des Lösungsansatzes zur selbstorganisierten Einsatzplanung in [Kapitel 3](#) ein Fokus auf die realitätsnahe Modellierung des Kommunikationsnetzes im Hinblick auf die unmittelbare Realisierbarkeit des Verfahrens in der Anwendung gelegt wurde, muss auch die für die Evaluation verwendete Simulationsumgebung diesen Aspekt betrachten. Es wird daher eine Simulationsumgebung verwendet, welche in der Lage ist, ein paketbasiertes Kommunikationsnetz für verteilte asynchrone Prozesse zur Verfügung zu stellen und die in [Definition 3.3.1](#) aufgestellten Anforderungen an das Kommunikationsnetz zu erfüllen. Die in der Programmiersprache Python¹ im Rahmen von [[Lün12](#)] entwickelte Simulationsumgebung ist aufgrund der Ähnlichkeit der verfolgten Fragestellung gut für die Wiederverwendung in der vorliegenden Arbeit geeignet. Diese Simulationsumgebung stellt dabei im Wesentlichen das Kommunikationsnetz sowie einen Simulationskern als virtuellem Zeitgeber für die Taktung des Systems zur Verfügung. Die Agenten des zu realisierenden Multiagentensystems werden nun als individuelle Prozesse implementiert und erhalten von der Simulationsumgebung einen eindeutigen Bezeichner, welcher fortan als Zieladresse für die Kommunikation der Agenten untereinander dient. Die Prozesse

¹<http://www.python.org>, letzter Zugriff 6. Januar 2014

können wahlweise verteilt auf verschiedenen Maschinen platziert werden, oder aber auf der selben Maschine ausgeführt werden. In beiden Fällen koordiniert der Simulationskern sowohl den Nachrichtentransfer über das Kommunikationsnetz, als auch die Zuweisung von Prozessorzeit an die Prozesse. Die Zeitgebung wird dabei auf eine Weise durchgeführt, dass für die Prozesse eine verteilte, asynchrone Umgebung simuliert wird.

6.2 KOMMUNIKATIONSMODELL

Wie in obigem Abschnitt beschrieben, wird jeder Agent des Multiagentensystems als eigenständiger Prozess implementiert, welcher über das vom Simulationskern kontrollierte Kommunikationsnetz mit anderen Agenten in Kontakt treten kann. Nachrichten werden dabei wahlweise blockierend (verbindungsorientiert) oder nicht-blockierend (verbindungslos) zugestellt. In beiden Fällen ist es möglich, dass der Sender über die Zustellung einer Nachricht informiert wird. Für blockierende Transmissionen wird hierzu der Rückgabewert der Sendefunktion verwendet. Bei nicht-blockierender Konfiguration wird stattdessen eine Rückruffunktion (engl. *callback*) eingesetzt.

Im vorliegenden Fall wird aufgrund der in [Definition 3.3.1](#) aufgestellten Anforderungen eine nicht-blockierende Transmission ohne Zustellbestätigung gewählt. Dies ermöglicht es Agenten, unmittelbar nach dem Versenden einer Nachricht mit anderen Aktionen fortzufahren, ohne auf die ggf. durch das unzuverlässige Kommunikationsnetz verzögerte Nachrichtenzustellung warten zu müssen.

6.3 PROZESSMODELL

Um die in [Abschnitt 4.4.2](#) beschriebene Eigenschaft der *unabhängigen Aktionsgeschwindigkeiten* zu realisieren, wird für jeden Verbundteilnehmer a_i eine über die Zeit variierende zufällige Reaktionszeit r_i eingeführt.

Definition 6.3.1 (Reaktionszeit)

Die *Reaktionszeit* eines Verbundteilnehmers a_i ist eine Zufallsgröße $r_i \in [0, \infty)$, welche den zeitlichen Abstand zwischen dem Eintreffen einer Nachricht und der Reaktion des Agenten auf diese Nachricht bestimmt.

Hiermit ist es möglich, den Nachrichtenempfang von den Verhaltensphasen *Wahrnehmung*, *Entscheidung* und *Aktion* eines Verbundteilnehmers zeitlich zu entkoppeln. Die resultierende Modellierung des Agentenprozesses ist in [Abbildung 6.1](#) dargestellt. Auf der linken Seite der Abbildung ist der Teilprozess *mailbox* zu sehen, welcher auf den Empfang von Nachrichten ausgelegt ist. Eingehende Nachrichten werden dabei im Zustand *store*(μ_j) in einem Pufferspeicher abgelegt. Mit jedem Speichervorgang wird eine interne bool'sche Variable *dirty* auf *true* gesetzt, um zu kennzeichnen, dass neue Informationen eingetroffen sind. Der auf der rechten Seite dargestellte Teilprozess *optimizer* startet im Zustand *sleep*(r_i). In unregelmäßigen, durch die Zufallsgröße r_i bestimmten Abständen prüft der Prozess nun mittels der Variable *dirty*, ob neue Informationen eingetroffen sind. Ist dies der Fall, wird die Variable auf *false* zurückgesetzt und anschließend die Verhaltensphasen *Wahrnehmung*, *Entscheidung* und *Aktion* durchgeführt. Hierbei ist zu beachten, dass im Teil *Wahrnehmung* nun möglicherweise nicht nur eine einzelne Nachricht in das Arbeitsgedächtnis des Verbundteilnehmers integriert werden muss, sondern ggf. mehrere Nachrichten, die sowohl vom selben, als auch von unterschiedlichen Absendern stammen können. Da Fahrplanauswahlen von Verbundteilnehmern jeweils mit individuellen Zählvariablen versehen sind (vgl. [Definition 3.6.3](#)), ist die Reihenfolge der Abarbeitung der gespeicherten Nachrichten hier unerheblich. Im Anschluss an diese Integrierung neuer Informationen in das Arbeitsgedächtnis wird der Pufferspeicher für empfangene Nachrichten geleert, und wie in [Abschnitt 4.2](#) beschrieben mit den Phasen *Entscheidung* und *Aktion* fortgefahren.

Diese Entkopplung des Nachrichtenempfanges von der Informationsverarbeitung eines Verbundteilnehmers bewirkt, dass die Systemteilnehmer nicht unmittelbar auf neue Informationen reagieren, sondern ggf. mehrere eingehende Nachrichten „sammeln“, bevor sie diese auswerten. Damit wird dem Gesamtsystem eine gewisse Trägheit verliehen, was in einem reduzierten Kommunikationsaufwand pro Zeiteinheit resultiert. Inwieweit sich das auf die erreichbare Lösungsgüte oder Konvergenzgeschwindigkeit des Verfahrens auswirkt, wird im Rahmen der Evaluation in [Abschnitt 7.3.3](#) untersucht.

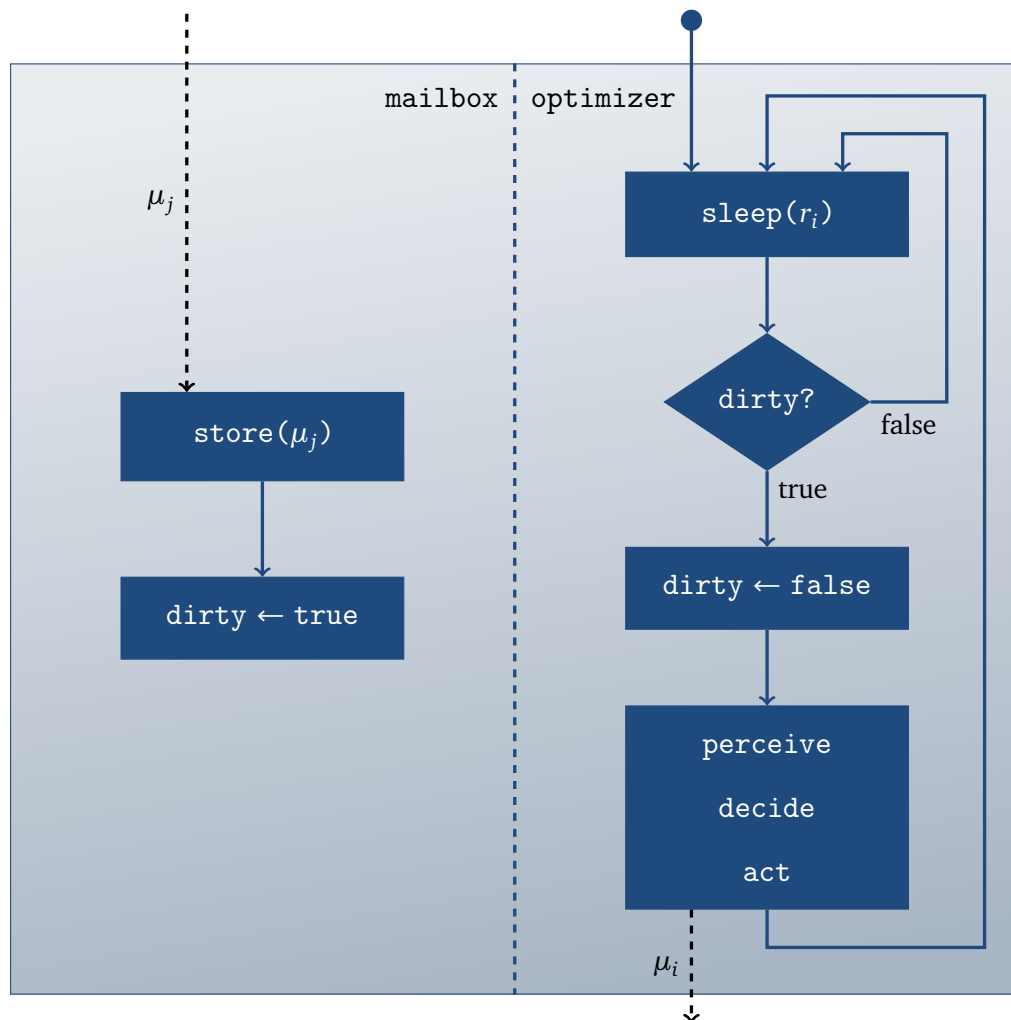


Abbildung 6.1: Nebenläufiges Prozessmodell eines Agenten im betrachteten System.

6.4 SUCHRAUM UND SUCHRAUMMODELL

Bei der Formalisierung des prädiktiven Einsatzplanungsproblem in [Abschnitt 3.1](#) wurde mit [Definition 3.1.10](#) für einen Verbundteilnehmer a_i ein Suchraum S_i definiert, welcher die von der assoziierten Energieanlage realisierbaren Fahrpläne enthält. Dabei sollte es a_i möglich sein, Anfragen in Form von hypothetischen Wirkleistungsverläufen π an das zugehörige Suchraummodell σ_i zu stellen, und als Antwort darauf möglichst ähnliche, aber realisierbare Fahrpläne θ_i zu erhalten. Dies wird im Rahmen der Heuristik COHDA eingesetzt, damit ein Verbundteilnehmer in der Verhaltensphase *Entscheidung* eine optimale Fahrplanauswahl hinsichtlich seiner wahrgenommenen Situation erstellen kann. Im Folgenden werden nun zwei mögliche Realisierungen von Suchräumen und Suchraummodellen vorgestellt.

6.4.1 Enumeration mit sequenzieller Suche

Eine einfache Möglichkeit besteht darin, als Suchraum eine endliche Menge von realisierbaren Fahrplänen zu enumerieren, und diese in einer Liste zu speichern. Die Erzeugung von realisierbaren Fahrplänen kann etwa durch Bestimmung des Anlagenzustandes auf Basis aktueller Messwerte und anschließender Simulation eines integrierten Softwaremodells der Energieanlage geschehen, in welchem verschiedene mögliche Verhaltensweisen der Anlage im betrachteten Planungshorizont unter Einhaltung der anlagentechnischen Restriktionen erzeugt werden. Für diese Vorgehensweise ist es erforderlich, dass die Energieanlage über die technischen Fähigkeiten verfügt, eine solche Simulation durchzuführen. Ein Gegenstand aktueller Forschung hierzu ist die Methode der modellbasierten prädiktiven Regelung (engl. *Model Predictive Control*). Diese besitzt in der verfahrenstechnischen Industrie bereits eine hohe Akzeptanz [[Rau03](#), Kap. 5] und wird seit einigen Jahren für die Anwendung in Steuerungsverfahren in der Energiedomäne untersucht [[Neg07](#); [McN12](#)]. Konkrete Betrachtungen von Mikro-Blockheizkraftwerken sowie Wärmepumpen in diesem Kontext liefern etwa [[Hou10](#)] und [[Wim04](#)]. Auf einem solchermaßen gebildeten

Suchraum S_i^{enum} kann dann das Suchraummodell σ_i^{enum} wie folgt definiert werden:

$$\begin{aligned}\sigma_i^{\text{enum}} : \mathbb{R}^{|T|} &\rightarrow \mathbb{R}^{|T|}, \\ \pi &\mapsto \arg \min_{\theta_i \in S_i} (\|\pi - \theta_i\|_1)\end{aligned}$$

Hier wurde exemplarisch die Summennorm als Distanzmetrik verwendet, aber auch alternative Metriken sind möglich, siehe [Abschnitt 3.1.1](#). Die Funktion $\arg \min$ kann nun bspw. in Form einer sequenziellen Suche realisiert werden.

Diese Kombination aus einer Listendarstellung des Suchraumes und einer auf dieser Liste operierenden Suchfunktion ist – sofern die technischen Voraussetzungen erfüllt sind – leicht realisierbar. Komplexere Kombinationen aus Datenstruktur zur Speicherung der enumerierten Elemente und darauf operierender Suchfunktion, wie etwa Suchbäume, können nur in speziellen Fällen angewendet werden, da sie eine zur Distanzmetrik passende Ordnungsrelation auf den zu speichernden Elementen voraussetzen. Bei Verwendung der Summennorm als Distanzmetrik existiert keine sinnvolle Ordnungsrelation für Elemente mit $|T| > 1$ Dimensionen (wie es im betrachteten Anwendungsfall üblicherweise der Fall ist), daher stellt die dargestellte Kombination aus Liste und sequenzieller Suche hier eine zulässige Wahl dar. Wird hingegen eine Distanzmetrik verwendet, welche eine sinnvolle Ordnungsrelation erlaubt, so sollte eine dazu passende Datenstruktur samt Suchfunktion (bspw. Suchbäume) verwendet werden. Dies wird jedoch hier nicht weiter betrachtet, da im Folgenden eine Modellierung vorgestellt wird, welche nicht auf Enumeration beruht. Die Motivation dazu entspringt der Frage, wie viele realisierbare Fahrpläne für einen Suchraum erzeugt werden müssen, um den Handlungsspielraum der betreffenden Anlage ausreichend abzubilden. Prinzipiell gilt dabei: Je mehr unterschiedliche Fahrpläne jede Anlage anbieten kann, desto größer ist die Chance auf eine optimale Lösung des prädiktiven Einsatzplanungsproblem. Zugleich steigt jedoch sowohl der Speicher- als auch der Berechnungsaufwand erheblich, da für eine große Menge von realisierbaren Fahrplänen diese zunächst für jede Anlage berechnet (d. h. simuliert) werden müssten, und zudem für jede Anlage in Form einer Enumeration explizit gespeichert werden müssten. Weiterhin steigt der Berechnungsaufwand für die im Suchraummodell enthaltene Suchfunktion je nach verwendeter Distanzmetrik mehr oder weniger stark mit der Größe des Suchraumes an. Im folgenden

Abschnitt wird daher eine alternative Umsetzung auf der Basis eines maschinellen Lernverfahrens vorgestellt, welche diese Nachteile vermeidet und in der Lage ist, einen nahezu vollständigen Handlungsspielraum bei gleichbleibendem Speicher- und Berechnungsaufwand abzubilden.

6.4.2 Maschinelles Lernverfahren

Um den Handlungsspielraum einer Energieanlage abzuschätzen, existieren prinzipiell mehrere Herangehensweisen: Liegt ein mathematisches Modell der Anlage vor, kann ausgehend von dem aktuellen Anlagenzustand eine Betrachtung der aus dem Modell ableitbaren Nebenbedingungen und Freiheitsgrade erfolgen, um daraus analytisch den derzeitigen Handlungsspielraum der Anlage zu bestimmen (da das Modell der Anlage bereits eine Vereinfachung der Realität ist, handelt es sich trotz der analytischen Herangehensweise um eine Abschätzung). Problematisch hierbei ist jedoch, dass dieses Vorgehen nur schwer automatisierbar ist, und für jeden Anlagentyp individuell angepasst werden muss. Dennoch stellt dies derzeit im internationalen Forschungskontext eine verbreitete Methode dar, um Freiheitsgrade von flexiblen Energieanlagen zur Laufzeit zu bestimmen: Oftmals wird dies für einfache Anlagentypen wie thermische Geräte (bspw. Kühltürme) oder elektrische Speicher angewendet (vgl. dazu die in [Abschnitt 1.3](#) referenzierte Literatur). Auch für komplexere Anlagentypen wird dieses Verfahren gelegentlich verwendet (siehe z. B. [\[Trö10\]](#)).

Eine alternative Methode besteht darin, ein maschinelles Lernverfahren anzuwenden, um den Handlungsspielraum einer Energieanlage ausgehend von einigen wenigen Informationen über das Verhalten der Anlage zu extrapolieren. Das im Folgenden beschriebene Verfahren wurde von Jörg Bremer entwickelt und ist in [\[BRS10; BRS11a; BRS11b\]](#) ausführlich dokumentiert. Es soll an dieser Stelle nur kurz umrissen werden, für Details siehe die angegebene Literatur. Ebenso wie die oben vorgestellte Methode des enumerierten Suchraumes basiert dieses Verfahren auf der Bestimmung einer Menge von realisierbaren Fahrplänen mittels eines Anlagensimulationsmodells. Statt der direkten Verwendung als Suchraum wird diese Menge hier aber als *Muster* für den tatsächlichen Handlungsspielraum der Anlage interpretiert. Ausgehend von diesem Muster kann nun der Handlungsspielraum abgeschätzt werden. Um dies zu erreichen, wird eine Kern-Transformation der Daten

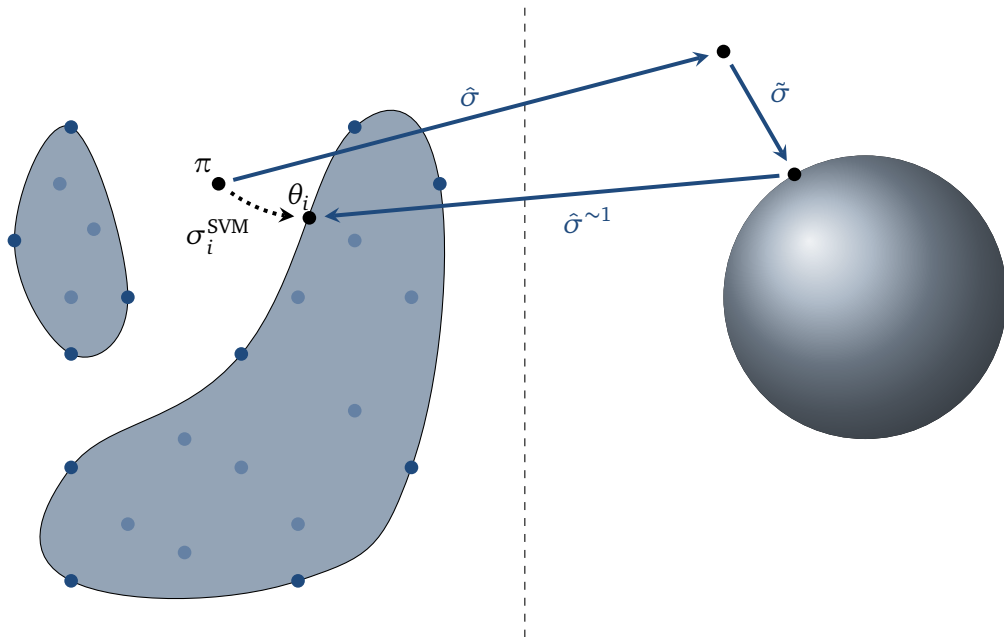


Abbildung 6.2: Suchraummodell – Dekoder Ansatz nach [BS13]. Autor der Darstellung: Jörg Bremer (mit Genehmigung verwendet).

in einen höherdimensionalen Raum durchgeführt, in welchem anschließend eine Hyperkugel bestimmt wird, die alle Elemente des Musters vollständig umschließt. Die Rücktransformation in den ursprünglichen Raum liefert dann die Grenzfläche zwischen realisierbaren und nicht realisierbaren Fahrplänen. Diese kann mittels Stützvektoren beschrieben werden und stellt damit einen Klassifikator in Form einer *Support Vector Machine* dar. Der tatsächliche Handlungsspielraum der Anlage wurde also auf Basis einiger exemplarischer Handlungsoptionen (in Form von simulativ ermittelten realisierbaren Fahrplänen) abgeschätzt und durch eine Menge von Stützvektoren beschrieben. Diese Stützvektoren definieren somit den Suchraum S_i^{SVM} der Anlage.

Nun muss auf diesem Suchraum noch eine geeignete Suchfunktion definiert werden, um hypothetische Fahrpläne auf möglichst ähnliche, realisierbare Fahrpläne abbilden zu können. Der *Dekoder* Ansatz aus [BS13] liefert eine solche Suchfunktion und bedient sich wiederum der Kern-Transformation (vgl. [Abbildung 6.2](#)): Sei π

ein hypothetischer Fahrplan, für den ein möglichst ähnlicher, realisierbarer Fahrplan gefunden werden soll. Liegt π bereits innerhalb der durch die Stützvektoren beschriebenen Grenzfläche, sodass $\pi \in S_i^{\text{SVM}}$, dann wird π ohne Modifikation unmittelbar als Antwort zurückgeliefert. Liegt π jedoch außerhalb des Handlungsspielraumes, so wird die Anfrage mittels einer Funktion $\hat{\sigma}$ in den höherdimensionalen Raum transformiert. Dort liegt π geometrisch betrachtet außerhalb der Hyperkugel, welche den Handlungsspielraum umschließt. Da das Zentrum sowie der Radius der Hyperkugel bekannt sind, kann π mittels einer einfachen Vektoroperation $\tilde{\sigma}$ auf die Hülle der Kugel projiziert werden. Wird dieser projizierte Punkt nun mittels $\hat{\sigma}^{-1}$ in den ursprünglichen Raum zurück transformiert, ergibt sich ein Punkt θ_i unmittelbar auf der Grenzfläche des Handlungsspielraumes. Effektiv wurde somit der nicht realisierbare Wirkleistungsverlauf π auf einen möglichst ähnlichen, realisierbaren Fahrplan θ_i abgebildet. Dieser *Dekoder* Ansatz liefert somit eine auf dem Suchraum S_i^{SVM} operierende Suchfunktion, und definiert damit das Suchraummodell σ_i^{SVM} als Komposition der soeben beschriebenen Transformationen:

$$\begin{aligned} \sigma_i^{\text{SVM}} : \mathbb{R}^{|T|} &\rightarrow \mathbb{R}^{|T|}, \\ \pi &\mapsto (\hat{\sigma}^{-1} \circ \tilde{\sigma} \circ \hat{\sigma})(\pi) \end{aligned}$$

Der wesentliche Vorteil dieses Verfahrens gegenüber dem einfachen Enumerieren eines Suchraumes liegt in der deutlich höheren Anzahl realisierbarer Fahrpläne, welche in dem Suchraum S_i^{SVM} enthalten sind. Der Suchraum wird durch eine geringe Menge an Stützvektoren vollständig beschrieben, sodass der Speicheraufwand nicht unmittelbar von der Menge der durch S_i^{SVM} beschriebenen Fahrpläne abhängig ist. Gleiches gilt für den Berechnungsaufwand: Es muss lediglich ausgehend von dem verwendeten Muster die Menge der Stützvektoren bestimmt werden, was wiederum unabhängig von der Menge der darstellbaren Fahrpläne ist. Auch das dazugehörige Suchraummodell σ_i^{SVM} zeigt diese Vorteile im Vergleich zu dem auf Listen operierenden Suchraummodell σ_i^{enum} aus dem vorangegangenen Abschnitt. Denn der *Dekoder* Ansatz ermöglicht es, durch geschickte geometrische Transformationen, einen hypothetischen Wirkleistungsverlauf direkt auf einen realisierbaren Fahrplan abzubilden, ohne dabei den gesamten Suchraum durchlaufen zu müssen. Weitere Details zu diesem Vergleich sind in [HBS13] sowie in Abschnitt 7.6.3 zu finden.

6.5 UMSETZUNG FÜR REALE ANLAGEN

In der vorliegenden Arbeit wird postuliert, dass die an der Einsatzplanung beteiligten Energieanlagen jeweils durch einen lokal implementierten Softwareagenten repräsentiert werden, welcher stellvertretend die Koordination mit den anderen Akteuren vornimmt. In den vorigen Abschnitten wurde dazu eine Softwarearchitektur vorgestellt, die eine Evaluation des entworfenen Einsatzplanungsverfahrens unter diesen Voraussetzungen simulativ ermöglicht. Hierbei stellt sich jedoch die Frage der Umsetzbarkeit des vorgeschlagenen Konzeptes für reale Anlagen, und soll daher im Folgenden kurz diskutiert werden.

In [Abschnitt 3.2](#) wurde ein Agent als eine autonome, selbstinteressierte, zielgerichtet handelnde Softwarekomponente beschrieben, welche steuerungstechnisch direkt an die von ihr repräsentierte Energieanlage gekoppelt ist und über ein gegebenes Kommunikationsnetz ([Abschnitt 3.3](#)) mit gleichartigen Agenten interagieren kann. In obigem [Abschnitt 6.4](#) wurden die geforderten berechnungstechnischen Fähigkeiten eines solchen Agenten genauer spezifiziert. Zur Realisierung dieser Fähigkeiten wurde die modellbasierte prädiktive Regelung als geeignet identifiziert. Die momentan im Handel befindlichen Energieanlagen für den Privat- und Gewerbebetrieb weisen derzeit allerdings noch keine agentenbasierten (im Sinne der obigen Beschreibung, d. h. mit anderen Agenten autonom interagierende) Steuerungseinheiten auf. Die technischen Möglichkeiten hierfür sind jedoch gegeben – so sind etwa die Mikro-Blockheizkraftwerke des Herstellers Vaillant² bereits mit leistungsstarken Systemregler-Modulen ausgestattet, welche einerseits aufgrund unmittelbar an die Anlage angebundener Sensoren die nötigen Informationen zum Aufbau eines Suchraummodells besitzen, und mögliche Fahrpläne mittels direkt zugreifbarer Regelstrecken der Anlagen nach Bedarf umsetzen können. Zudem weisen diese Module eine kommunikationstechnische Anbindung an lokale Netzwerke auf, womit die technischen Anforderungen an eine agentenbasierte Steuerung hypothetisch erfüllt sind und durch die Verwendung einer entsprechenden Betriebssoftware realisiert werden könnten.

Für einen solchen Betrieb ist aber auch die Infrastruktur auf übergeordneter Ebene zu betrachten. Das in dieser Arbeit vorgeschlagene System wurde mit Hinblick auf technische Realisierbarkeit gestaltet (bspw. durch die asynchrone Modellierung des

²<http://www.vaillant.de/Produkte/Kraft-Waerme-Kopplung>, letzter Zugriff 6. Januar 2014

Agentenverhaltens und die damit verbundene Unabhängigkeit von echtzeitfähiger Kommunikation). Die Umsetzung eines derartigen Systems für reale Handelsplattformen erfordert aber zudem eine Spezifikation, welche sich einfach in bestehende Strukturen integrieren lässt, herstellerübergreifend implementierbar ist und dabei Aspekte wie Datenschutz und Betriebssicherheit berücksichtigt. Eine Möglichkeit, dies zu realisieren, ist mit der von der OPC Foundation³ entwickelten *OPC Unified Architecture* (OPC UA) gegeben. Dabei handelt es sich um eine Reihe von Standards, die ihren Ursprung in der Automatisierungstechnik besitzt und abstrakt betrachtet die strukturierte Erstellung von Informationsmodellen und den Austausch entsprechend formatierter Daten über lokale Netzwerkgrenzen hinweg erlaubt. Die Architektur baut dabei auf ein generisches Objektmodell auf, das den Einsatz in beliebigen Anwendungsdomänen erlaubt. Für den vorliegenden Anwendungsfall wurden die Datenmodelle des CIM Standards⁴ (*Common Information Model*) und der Norm IEC 61850⁵ als zentrale Komponenten identifiziert und in die OPC UA integriert [RS13]. Durch die Bereitstellung von Referenzimplementierungen wird der Einsatz dieser Architektur in der Praxis vereinfacht. Die Umsetzung eines auf dieser Architektur basierenden Systems für agentenbasierte Steuerungen wird derzeit im Rahmen der modularen Simulationsplattform mosaik⁶ untersucht [Sch+13].

³<http://www.opcfoundation.org>, letzter Zugriff 6. Januar 2014

⁴<http://www.dmtf.org/standards/cim>, letzter Zugriff 6. Januar 2014

⁵<http://www.iec.ch/smartgrid/standards>, letzter Zugriff 6. Januar 2014

⁶<https://mosaik.offis.de>, letzter Zugriff 6. Januar 2014

7

EVALUATION

Zu Beginn dieser Arbeit wurde das Ziel formuliert, einen Ansatz zur verteilten Einsatzplanung zu entwickeln. Dieser sollte den Kommunikationsaufwand der beteiligten Akteure soweit wie möglich minimieren, dabei jedoch die Koordination zwischen diesen Akteuren hinsichtlich der Erfüllung eines gemeinsamen Wirkleistungsproduktes maximieren. In [Kapitel 3](#) und [4](#) wurde dazu ein Lösungsansatz präsentiert, der auf einem Selbstorganisationsmechanismus beruht und es ermöglicht, eine rein verteilte Konsensbildung durchzuführen. Die garantierte Konvergenz des Verfahrens zu einem gemeinsamen Einsatzplan wurde in [Kapitel 5](#) bereits formal hergeleitet. Mit [Lemma 1](#) wurde dabei zudem die *anytime* Eigenschaft des Verfahrens gezeigt. Bisher konnte jedoch keine Aussage bzgl. der Performanz oder der Effizienz des Verfahrens getroffen werden.

Performanz bezeichnet die Fähigkeit eines Verfahrens, seine Aufgabe zu erfüllen.

Im vorliegenden Fall soll ein Optimierungsproblem gelöst werden, daher wird als Performanzindikator die Güte der von der Heuristik produzierten Lösung herangezogen.

Effizienz beschreibt die Beanspruchung von Ressourcen eines Verfahrens. Je mehr Ressourcen für die Erfüllung seiner Aufgabe benötigt werden, desto ineffizienter ist das Verfahren. Da es sich im vorliegenden Fall um ein verteiltes System handelt, werden die Ressourcentypen *Laufzeit*, *Nachrichtenaufkommen* sowie *Berechnungsaufwand* für die Effizienzbewertung betrachtet.

Im Folgenden werden die Ergebnisse einer Simulationsstudie zur Beleuchtung dieser Qualitätsmerkmale vorgestellt. Die Durchführung der Simulationen stützt sich auf die in [Kapitel 6](#) beschriebene prototypische Realisierung des Einsatzplanungsverfahrens. Um die Qualitätsmerkmale des Verfahrens möglichst unabhängig von domänenspezifischen Einflüssen bewerten zu können, wird die Evaluation zunächst

Parameter	Beschreibung	Standardbelegung
O	Overlay-Topologie	<i>Small-World</i> , $\phi = 0.5$
$d_{\max}$	Maximale Nachrichtenverzögerung	1
$r_{\max}$	Maximale Reaktionszeit	1
ψ_{stat}	Statische Gesinnungswahl	0
ψ_{dyn}	Dynamische Gesinnungswahl	0
m	Größe des Verbundes A	10
n_i	Größe eines Suchraums S_i	$\forall a_i \in A : n_i = 5$
q	Länge des Planungshorizontes T	5
σ	Suchraummodell	σ^{enum}

Tabelle 7.1: Parameter mit Standardwerten für synthetische Probleminstanzen.

anhand von synthetischen Probleminstanzen durchgeführt. Es werden verschiedene Parameter untersucht, welche einen Einfluss auf das Verhalten der Heuristik haben. Diese sind im Einzelnen in [Tabelle 7.1](#) aufgeführt. Die Tabelle beinhaltet auch die Standardbelegung der Parameter, welche die Ausgangslage der Evaluation darstellt und für die Untersuchungen jeweils als gegeben angenommen werden kann, sofern in den einzelnen Abschnitten nicht anders angegeben. Als Overlay-Netz wird standardmäßig wie in [Abschnitt 4.1](#) eine *Small-World* Topologie nach der Konstruktionsvorschrift des verbesserten Modells von Watts und Strogatz aufgrund der dort formulierten Vorteile dieser Topologie verwendet. Der Parameter ϕ bestimmt die Dichte der Topologie über die Anzahl an in den Ring einzufügenden Querverbindungen [[Str01](#), S. 273] und liefert mit dem Wert 0.5 im Durchschnitt eine Querverbindung pro Verbundteilnehmer. Nachrichtenverzögerungen sowie verzögerte Reaktionen der Verbundteilnehmer sind in der Standardeinstellung nicht vorhanden (der Minimalwert von $d_{\max} = r_{\max} = 1$ ergibt sich aus [Definition 5.2.1](#) in [Abschnitt 5.2](#)). Die in [Abschnitt 3.6.4](#) eingeführte freie Gesinnungswahl wird als Wahrscheinlichkeit des Abweisens eines Fahrplans modelliert und beträgt im Standardfall $\psi_{\text{stat}} = \psi_{\text{dyn}} = 0$ (zur Erläuterung siehe [Abschnitt 7.3.4](#)), sodass jeder vorhandene Fahrplan als mögliche Lösung von einem Verbundteilnehmer akzeptiert wird (d. h. die Gesinnungswahl hat bei dieser Konfiguration keinen Effekt). Die Standardwerte für m , n_i und q sind durch die Konstruktionsvorschrift der verwendeten synthetischen Probleminstanz

$P(m = 10, n = 5, q = 5)$ vorgegeben (siehe [Abschnitt 7.1](#)). Als Suchraummodell wird zunächst das einfache enumerierte Modell aus [Abschnitt 6.4.1](#) verwendet. Für die Distanzmetrik kommt wiederum die Summennorm zum Einsatz.

Im Laufe der Evaluation werden diese Parameter einzeln im Detail untersucht und ihre Auswirkungen auf Effizienz und Performanz des Verfahrens ausgewertet. Die Experimente werden dabei nach *Robustheit* und *Skalierbarkeit* kategorisiert: Während die Robustheit die Auswirkungen der Laufzeitparameter Overlay-Topologie, Nachrichtenverzögerung, Reaktionszeit und Gesinnungswahl auf gleichbleibende Problemgrößen untersucht, fokussiert sich die Skalierbarkeit auf das Verhalten des Verfahrens bei wachsenden Problemgrößen hinsichtlich Verbundgröße, Suchraumgröße und Planungshorizont. In allen diesen Simulationsreihen wird jede Konfiguration (d. h. jede Parameterbelegung) 100 Mal mit jeweils verschiedenem Seed-Wert für den Pseudozufallszahlengenerator¹ simuliert. Im Rahmen der so durchgeführten Evaluation der synthetischen Probleminstanzen ([Abschnitte 7.1 bis 7.4](#)) werden zunächst die qualitativen Zusammenhänge der einzelnen Parameter auf die obigen Qualitätsmerkmale betrachtet. Für tiefergehende Erkenntnisse bzgl. der funktionalen Zusammenhänge wird im Anschluss in [Abschnitt 7.5](#) eine Regressionsanalyse durchgeführt. In einem weiteren Teil werden daran anschließend dann domänenspezifische Szenarien untersucht, um die Einsatzfähigkeit des Verfahrens für den in dieser Arbeit fokussierten Anwendungsfall der prädiktiven Einsatzplanung im Smart Grid zu diskutieren ([Abschnitt 7.6](#)). In diesem Kontext wird zudem der Einfluss verschiedener Suchraummodelle auf die Heuristik betrachtet. Abschließend liefert [Abschnitt 7.7](#) einen Vergleich von COHDA mit dem *Stigspace* Ansatz, welcher bereits in [Abschnitt 1.3.2](#) beschrieben wurde.

7.1 SYNTHETISCHE PROBLEMINSTANZEN

In [Abschnitt 3.1.1](#) wurde eine Formulierung des betrachteten kombinatorischen Problems PEP als ganzzahliges lineares Optimierungsproblem (MC-COP) vorgestellt. Wie dort bereits erläutert, liegt dieses Problem in der Problemklasse „schwach NP-vollständig“. Eine *Probleminstanz* ist eine konkrete Instanz des Problems mit defi-

¹Mersenne Twister nach [MN98], siehe <http://docs.python.org/2.7/library/random.html> (letzter Zugriff: 6. Januar 2014)

nierten Eingabeparametern. Eine solche Instanz wird im Folgenden als synthetisch bezeichnet, wenn die Eingabeparameter vollständig artifiziell konstruiert wurden und sich nicht aus einem Anwendungsfall ableiten. Das Ziel bei der Verwendung von synthetischen Probleminstanzen ist es, das Verfahren unter vollständig bekannten Bedingungen zu evaluieren, um beobachtete Effekte eindeutig interpretieren zu können. Eine Voraussetzung dabei ist eine gewissenhafte Konstruktion geeigneter Parameterbelegungen, für die inhärente Eigenschaften (wie etwa die Verteilung optimaler Lösungen im Lösungsraum) im Vorherein bekannt sind.

Wie in [Abschnitt 3.1.1](#) dargelegt, stellt das MC-COP eine Verallgemeinerung des mehrdimensionalen Multiple-Choice Subset-Sum Problem (MC-SSP) dar, eine Variante des Untersummenproblems, welches zwar ein seit einigen Dekaden bekanntes Problem darstellt, aber bisher nicht intensiv untersucht wurde. Das MC-SSP weist allerdings wiederum sehr starke Ähnlichkeit mit der entsprechenden Variante des Rucksackproblems (mehrdimensionales MC-KP, vgl. [[Pis99](#); [HMS04](#); [Mos+06](#)]) auf. Bei diesem Problem besitzen die zu kombinierenden Elemente des Suchraumes zwei Attribute: Gewicht und Profit. Das Optimierungsziel ist es dabei, den aggregierten Profit der ausgewählten Elemente zu maximieren, dabei darf die Summe der dazu gehörigen Gewichte eine vordefinierte Schranke jedoch nicht überschreiten. In diesem Kontext wurden Probleminstanzen als besonders schwer lösbar identifiziert, welche sowohl eine starke Korrelation zwischen den Profiten der einzelnen Elemente des Suchraumes, als auch zwischen der Gewichts- und Profitverteilung in den Elementen insgesamt aufweisen [[Pis95](#); [HLS10](#)]. Aus diesen beiden Korrelationen folgt nun insbesondere, dass auch die Gewichte der Elemente stark miteinander korreliert sind. Da die (mehrdimensionalen) Gewichte mit den Fahrplänen des PEP gleichzusetzen sind (siehe [Abschnitt 3.1.1](#)), stellen solche Probleminstanzen nach Eliminierung der Profitwerte auch für die vorliegende Problemstellung geeignete, als schwer einzustufende Instanzen dar. Zu bemerken ist, dass hierbei zunächst die durch die freie Gesinnungswahl in [Abschnitt 3.6.4](#) eingeführte Mehrziel-Erweiterung außer Acht gelassen wird. Dies ist zulässig, da diese Erweiterung lediglich zu einer Verringerung der Problemgröße führt, an der Struktur des gegebenen Problems jedoch nichts ändert (vgl. [Abschnitt 5.3](#)). In [Abschnitt 7.3.4](#) werden die betrachteten Probleminstanzen dann um die Gesinnungswahl ergänzt, um deren Effekt bzgl. Performanz und Effizienz des Verfahrens zu untersuchen.

7.1.1 Konstruktion

Eine Konstruktionsvorschrift derartiger Probleminstanzen mit korrelierten Profiten für die mehrdimensionale Variante des MC-KP liefert [HLS10], welche gemäß obiger Überlegung für die vorliegende Problemstellung adaptiert wird. Konkret handelt es sich dabei um die Gruppe G-C(L)-D(S). Die Vorgehensweise zur Konstruktion ist wie folgt: Es existieren m Klassen (= Verbundteilnehmer) mit jeweils n möglichen Gewichtsvektoren (= realisierbare Fahrpläne), welche jeweils q Dimensionen besitzen (= Planungshorizont mit q Zeitintervallen). Die Werte m, n, q sind prinzipiell frei wählbar, aus Gründen der Vergleichbarkeit werden sie aber wie in der referenzierten Literatur angegeben verwendet: $P(m = 10, n = 5, q = 5)$. Die (temporären) Profite für die einzelnen Gewichtsvektoren werden nun klassenübergreifend mittels einer linearen Funktion erzeugt. Die Berechnungsvorschrift für den Profit p_{ij} des Gewichtsvektors j in Klasse i ist

$$p_{ij} = \frac{j-1}{n-1} (p_i^{\max} - p_i^{\min}) + p_i^{\min} \quad (7.1)$$

mit den Intervallgrenzen

$$p_i^{\min} = b \cdot (i - 1) \quad (7.2)$$

$$p_i^{\max} = b \cdot i \quad (7.3)$$

und b als Konstante mit $b = 10$ gemäß [HLS10], sodass sich insgesamt über alle Klassen hinweg eine monoton steigende Folge von Profitwerten ergibt. Hieraus werden nun stark korrelierte Gewichtsvektoren abgeleitet. Die Gewichtskomponente für Dimension k eines Gewichtsvektors j in der Klasse i ist gegeben als

$$w_{ijk} = p_{ij} + \frac{p_i^{\max}}{\delta} \quad (7.4)$$

mit

$$\delta = k + 5 \quad (7.5)$$

Damit ergibt sich für jeden Gewichtsvektor eine für die Dimensionen $k \in [1, q]$ monoton steigende Folge von Gewichtskomponenten, sowie eine starke Korrelation zu der über die Klassen hinweg steigenden Folge von Profitwerten. Die Profitwerte

können nun außer Acht gelassen werden (sie dienten lediglich der Erzeugung der Korrelation für die Gewichte).

Das Ziel des Optimierungsproblem es, eine Kombination von Gewichtsvektoren zu finden, deren Aggregation $\mathbf{s}$ (= Lösungsvektor) einen vorgegebenen Zielvektor möglichst exakt annähert. Die Erzeugung dieses Zielvektors ist ebenfalls in der Konstruktionsvorschrift der Problemgruppe festgehalten. Hierzu wird der Wertebereich der Gewichte in jeder Dimension in $H = 100$ Intervalle unterteilt, und der Parameter $h \in [1, H]$ als Selektor für eines dieser Intervalle eingeführt. Dieser für eine spezifische Probleminstance festzulegende Wert bestimmt im Folgenden die Lage des Zielvektors im Lösungsraum relativ zum minimal respektive maximal erreichbaren Lösungsvektor und ermöglicht es, in Abhängigkeit der Struktur des Lösungsraumes einer gegebenen Probleminstance mehr oder minder schwer zu erfüllende Zielvektoren zu erzeugen. Die Berechnungsvorschrift für die k -te Komponente des Zielvektors lautet nach dieser Konstruktionsvorschrift:

$$z_k^h = \frac{h}{H+1} \left(\sum_{i=1}^m w_{ik}^{\max} - \sum_{i=1}^m w_{ik}^{\min} \right) + \sum_{i=1}^m w_{ik}^{\min} \quad (7.6)$$

wobei $w_{ik}^{\max}$ und $w_{ik}^{\min}$ die kleinste bzw. größte Gewichtskomponente der Klasse i in der Dimension k angeben. Diese Berechnung muss für das vorab gewählte h nun für jede Dimensionskomponente vorgenommen werden, sodass sich insgesamt ein Zielvektor $\mathbf{z}^h$ über alle q Dimensionen ergibt. Ein hoher Wert für h resultiert dann bspw. in einem Zielvektor nahe des maximal möglichen Lösungsvektors, d. h. die optimale Lösung der Probleminstance besteht dann aus einer Aggregation großer Gewichtsvektoren. Welcher Wert von h für eine Problemgruppe besonders schwer zu lösen ist, muss jeweils im Rahmen der Evaluationsplanung individuell bestimmt werden. Im vorliegenden Fall ist die Problemgruppe auf $P(10, 5, 5)$ fixiert. Mittels $H = 100$ und $h \in [1, H]$ ergeben sich daher 100 mögliche Zielvektoren, und damit 100 konkrete Probleminstance. Zur Veranschaulichung der Struktur dieser Instance sind die Gewichtsvektoren und Zielvektoren in [Abbildung 7.1](#) grafisch dargestellt. Der obere Graph enthält für jede der zehn Klassen (jeweils mit einer individuellen Farbe gekennzeichnet) die jeweils fünf beinhalteten Gewichtsvektoren mit je fünf Dimensionen. Der untere Graph zeigt die entsprechenden Zielvektoren für alle möglichen Werte des Parameters $h \in [1, H]$. Die Gewichts- und Zielvektoren sind dieser

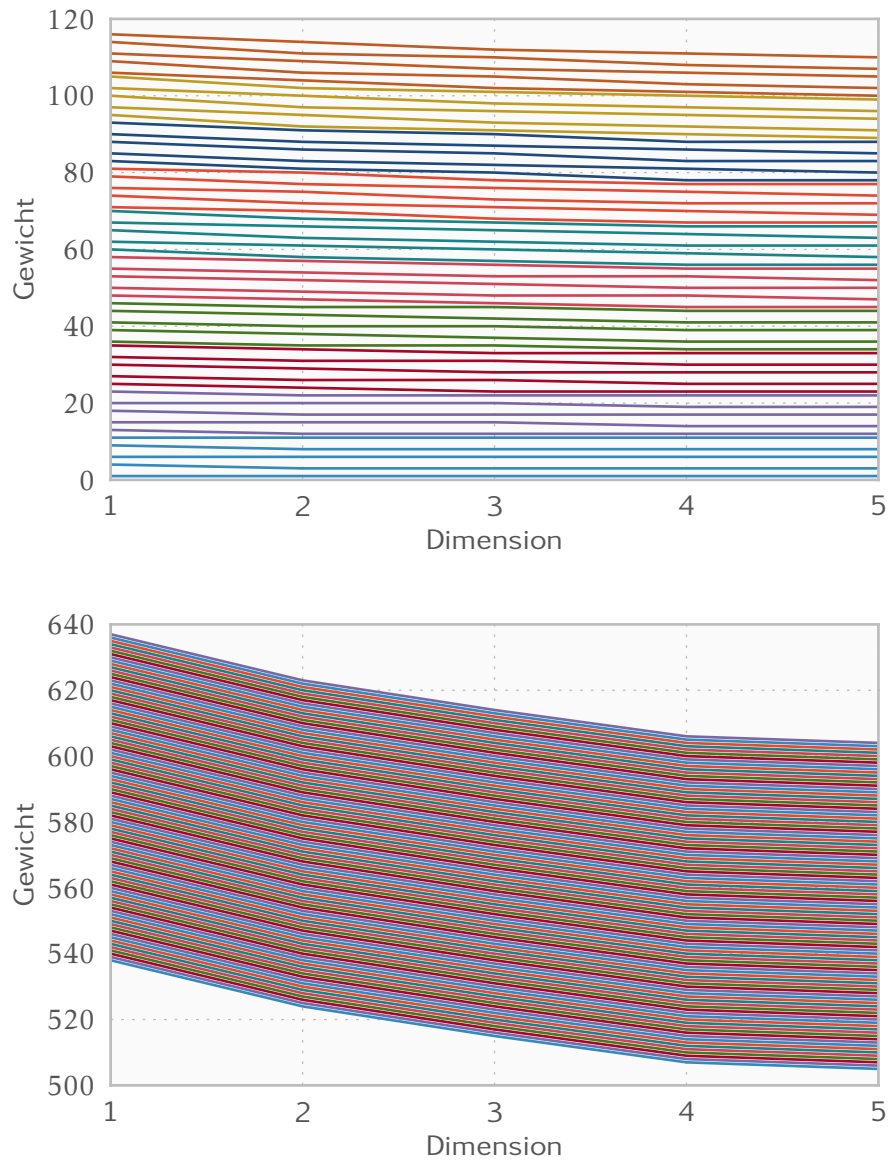


Abbildung 7.1: Gewichtsvektoren der Problemgruppe G-C(L)-D(S) für $P(10, 5, 5)$ (oben) sowie dazugehörige Zielvektoren für $h \in [1, H]$ mit $H = 100$ (unten).

Arbeit zur Referenz nochmals als Wertereihen in [Anhang A](#) beigelegt. Im Folgenden wird nun bestimmt, wie sich diese Problemgruppe für verschiedene Werte von h verhält.

7.1.2 Lösungsraum

Mit $h \in [1, 100]$ ergibt sich für die Problemgruppe $P(10, 5, 5)$ eine Menge von 100 konkreten Probleminstanzen. Da h die Lage des Zielvektors relativ zu den konstruierten Gewichten einer Probleminstanz festlegt, ist zu erwarten, dass nicht jede Instanz gleichermaßen lösbar ist. Aufgrund der kombinatorischen Natur der Problemstellung existiert aber trotz der geringen Größe der Parameter für jede Instanz eine Menge von $n^m = 5^{10} = 9\,765\,625$ verschiedenen Lösungsvektoren (= mögliche Kombinationen von Gewichtsvektoren). Um die Evaluation des in dieser Arbeit entwickelten Einsatzplanungsverfahrens nicht mit allen 100 Instanzen separat durchführen zu müssen, soll vorab ein geeigneter Wert h , und damit eine spezifische Probleminstanz zur weiteren Betrachtung ausgewählt werden. Es stellt sich daher die Frage nach der Schwierigkeit der Probleminstanzen für bestimmte Werte h .

Definition 7.1.1 (Schwierigkeit einer Probleminstanz)

Unter der *Schwierigkeit einer Probleminstanz* wird in der vorliegenden Arbeit ein Maß für die Lösbarkeit einer Probleminstanz verstanden. Dieses bezieht sich nicht notwendigerweise auf Optimalität: Da für ein gegebenes h jedem Lösungsvektor $\mathbf{s}$ im jeweiligen Lösungsraum eine Güte $v^h = f(z^h, \mathbf{s})$ hinsichtlich des Zielvektors z^h zugeordnet werden kann, ist es möglich die Häufigkeitsdichte aller auftretenden Lösungsgüten im Lösungsraum aufzustellen. f ist dabei die zu maximierende Gütefunktion nach [Definition 3.1.11](#) unter Verwendung der Summennorm als Distanzmetrik (vgl. dazu auch [Abschnitt 3.1.1](#)).

Diese Häufigkeitsdichte beschreibt dann die Auftretenswahrscheinlichkeit von Lösungsgüten für den gesamten Lösungsraum. Weist die Häufigkeitsdichte eine hohe Wahrscheinlichkeit für optimale oder nahe-optimale Lösungen auf, so wird die Schwierigkeit der betrachteten Probleminstanz als gering eingestuft. Umgekehrt ist die Schwierigkeit hoch, wenn nur sehr wenige Lösungen nahe des Optimums existieren.

Zur Bestimmung einer Problemistanz mit hoher Schwierigkeit werden nun die Häufigkeitsdichten der Lösungsüten aller Problemistanzen mit $h \in [1, 100]$ berechnet. Mittels der durch Anwendung der Exhaustionsmethode (*brute-force search*) gewonnenen besten und schlechtesten Gütewerte v_{best}^h und v_{worst}^h wird hierzu zunächst der normalisierte Fehler für alle Lösungsvektoren der jeweiligen Instanz mit

$$e^h(\mathbf{s}) = 1 - \frac{f(z^h, \mathbf{s}) - v_{worst}^h}{v_{best}^h - v_{worst}^h} \quad (7.7)$$

bestimmt. [Abbildung 7.2](#) zeigt ein dreidimensionales Histogramm für die Verteilung dieser Fehlerwerte im Lösungsraum für alle Instanzen $h \in [1, 100]$. Zum leichteren Verständnis ist das Histogramm aus zwei verschiedenen Perspektiven abgebildet (diese sind gegeneinander um 120° bzgl. Achse x_3 gedreht). Auf Achse x_1 sind die verschiedenen Instanzen abgetragen. Für jede Instanz h beschreibt die Ebene $x_2 - x_3$ nun die Häufigkeitsdichte f^h der durch die möglichen Lösungsvektoren resultierenden Fehlerwerte. Das heißt jeder Datenpunkt im dargestellten Koordinatensystem beschreibt innerhalb einer Instanz h die relative Häufigkeit $f_e^h(\mathbf{s}) \in [0, 1]$ des Auftretens eines Lösungsvektors $\mathbf{s}$ mit einem resultierenden Fehler e . Demnach häufen sich offenbar für sehr kleine und sehr große Werte h die Fehlerwerte im mittleren Bereich, d. h. sowohl optimale ($e^h = 0$) als auch pessimale ($e^h = 1$) Lösungsvektoren sind bei diesen Instanzen nur mit äußerst geringer Häufigkeit im Lösungsraum vertreten:

$$\begin{aligned} f_0^0(\mathbf{s}) &= 1.024 \cdot 10^{-7} \\ f_1^0(\mathbf{s}) &= 1.024 \cdot 10^{-7} \\ f_0^{100}(\mathbf{s}) &= 1.126 \cdot 10^{-6} \\ f_1^{100}(\mathbf{s}) &= 1.126 \cdot 10^{-6} \end{aligned}$$

Für mittlere Werte h hingegen verschiebt sich die Häufigkeitsdichte in Richtung sehr kleiner Fehlerwerte, d. h. für diese Instanzen existieren vergleichsweise viele optimale und nahe-optimale Lösungsvektoren:

$$\begin{aligned} f_0^{50}(\mathbf{s}) &= 2.413 \cdot 10^{-2} \\ f_{0.01}^{50}(\mathbf{s}) &= 6.052 \cdot 10^{-2} \end{aligned}$$

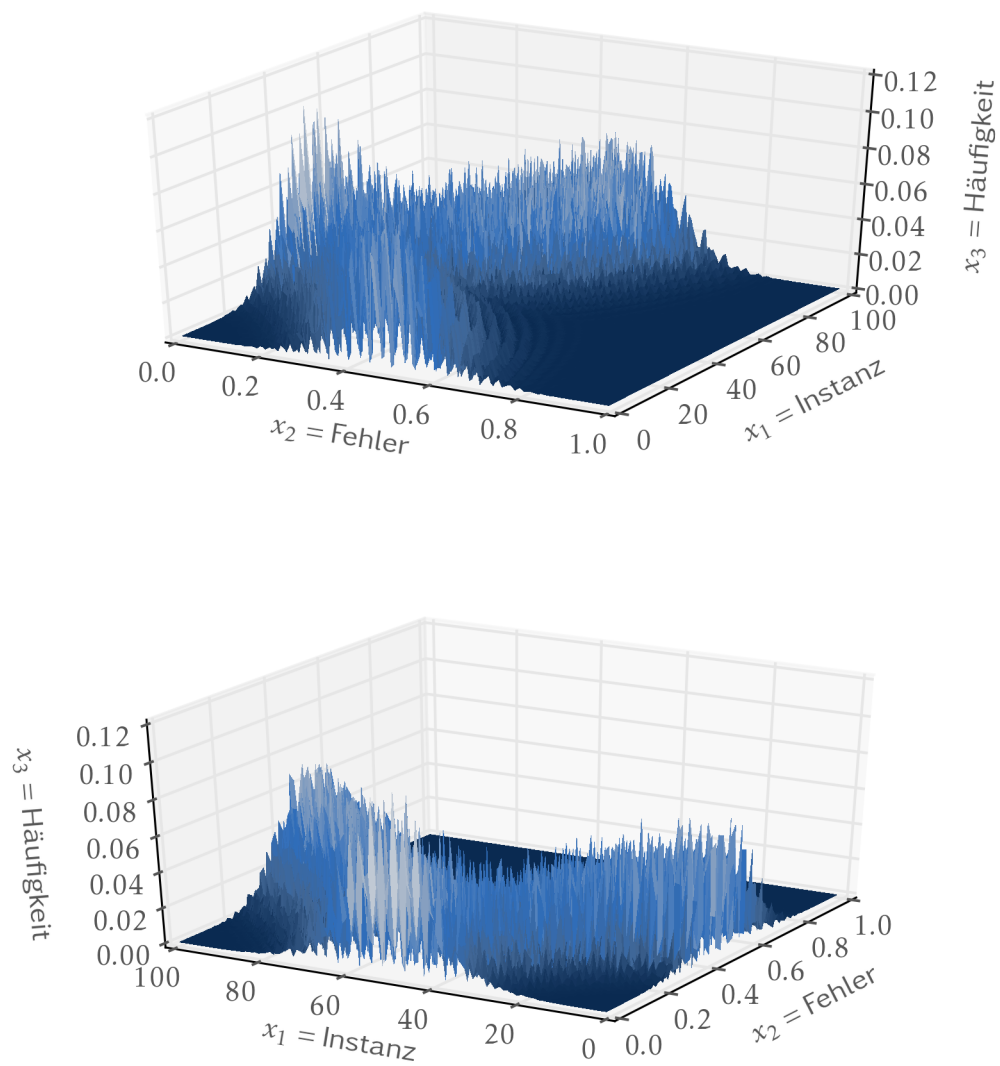


Abbildung 7.2: Häufigkeitsdichte der Fehlerwerte im Lösungsraum für alle Instanzen $h \in [1, 100]$, Darstellung aus zwei Perspektiven.

Diese Verteilung würde vermuten lassen, dass Instanzen mit sehr großen und sehr kleinen Werten h besonders schwer lösbar sind. Um diese Hypothese zu stützen, kann zusätzlich zur Häufigkeitsdichte die von einer Heuristik im Mittel produzierte Lösungsgüte je Instanz betrachtet werden. Um an dieser Stelle keine voreingenommene Aussagen zu erhalten, welche später in der eigentlichen Evaluation konsequenterweise nur bestätigt werden können, wird hierzu neben der in dieser Arbeit entwickelten COHDA Heuristik eine einfache *Random Walk* Heuristik herangezogen. Damit bleibt grundlegende Evaluationsplanung unabhängig von dem zu evaluierenden Verfahren. Dabei ist allerdings anzumerken, dass die *Random Walk* Heuristik nicht als Alternative zu COHDA verstanden werden sollte, da sie ein zentrales Verfahren mit vollständigem Wissen darstellt. Sie wird hier lediglich zur Untersuchung der vorgestellten Probleminstanzen angeführt. Ein Vergleich der COHDA Heuristik mit alternativen Verfahren wird später in [Abschnitt 7.7](#) gegeben.

Die *Random Walk* Heuristik ist in [Algorithmus 3](#) als Pseudocode dargestellt. Der Ausgangspunkt ist eine initiale Lösung, die durch zufällige punktuelle Modifikationen iterativ verbessert werden soll. Die initiale Lösung wurde im vorliegenden Fall mit dem *worst case* belegt, also mit derjenigen Lösung, die für die gegebenen Probleminstanz den größten Fehlerwert besitzt. Diese konnte wie oben aufgrund der geringen Problemgröße durch Anwendung der Exhaustionsmethode ermittelt werden. Nun wird für jede Klasse sequenziell das entsprechende Element der aktuellen Lösung mit einem zufälligen Element ersetzt, und die resultierende Lösung evaluiert. Liefert sie einen geringeren Fehler als die bisherige Lösung, wird mit der modifizierten Lösung weiter gerechnet, ansonsten wird sie wieder verworfen. Dies wird $l = 100$ Mal wiederholt, sodass insgesamt pro Instanz sequenziell 1000 zufällige Modifikationen der initialen Lösung stattfinden. Diese Heuristik wird nun für jede Instanz 100 Mal ausgeführt. Die sich ergebende Verteilung der Fehlerwerte ist in [Abbildung 7.3](#) dargestellt. Entgegen der obigen Hypothese können für sehr große und sehr kleine Werte h optimale Lösungen gefunden werden, während der verbleibende Fehler in Richtung mittlerer Werte für h tendenziell ansteigt. Ein Erklärungsansatz hierfür ist, dass für mittlere Werte von h viele nahe-optimale Lösungen im Lösungsraum existieren (vgl. [Abbildung 7.2](#)), welche lokale Minima für Fehlerfunktion [Gleichung 7.7](#) darstellen, sodass die Heuristik überwiegend in diesen Minima terminiert. Der grundlegende Trend des Verlaufes der Fehlerwerte wird von alternierenden Abweichungen nach oben und unten überlagert. Offenbar beinhaltet die Struktur der Probleminstanzen

Algorithmus 3 Random Walk Heuristik

Parameter:

$m := 10$ ▷ Anzahl Klassen (gegeben durch Probleminstance)
 $l := 100$ ▷ Anzahl Wiederholungen

Funktionen:

$\text{INIT}()$ ▷ Erzeuge initiale Lösung als array[1..m]
 $\text{RND}(i)$ ▷ Liefert zufälliges Element aus Klasse i
 $\text{COPY}(s)$ ▷ Erzeuge Kopie von s
 $e^h(s)$ ▷ Berechnet Fehler für Lösung s gemäß [Gleichung 7.7](#)

```

1:  $s \leftarrow \text{INIT}()$ 
2: for  $j \leftarrow 1, l$  do
3:   for  $i \leftarrow 1, m$  do
4:      $c \leftarrow \text{RND}(i)$  ▷ Wähle zufälliges Element für Klasse  $i$ 
5:      $s' \leftarrow \text{COPY}(s)$ 
6:      $s'[i] \leftarrow c$ 
7:     if  $e^h(s') < e^h(s)$  then ▷ Vergleiche Lösungen
8:        $s \leftarrow s'$  ▷ Falls besser, ersetze Lösung
9:     end if
10:  end for
11: end for
12: return  $s$  ▷ Ergebnis: beste gefundene Lösung

```

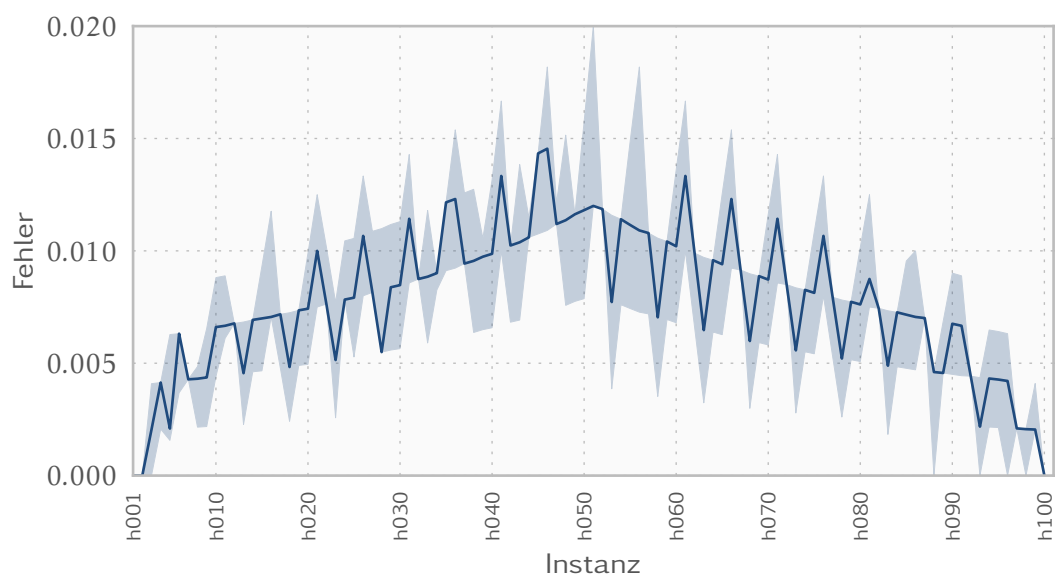


Abbildung 7.3: Verbleibender Fehler für alle Instanzen $h \in [1, 100]$ bei Verwendung einer *Random Walk* Heuristik. Darstellung als Median $\pm$ oberes/unteres Quartil für 100 Ausführungen je Instanz h .

für bestimmte Vielfache von h jeweils leichtere und schwierigere Variationen des Grundproblems.

Analog zur Anwendung der *Random Walk* Heuristik zeigt [Abbildung 7.4](#) (oben) die Verteilung der Fehlerwerte bei Anwendung der COHDA Heuristik. Dabei wurde als Overlay-Netz wie in [Abschnitt 4.1](#) eine *Small-World* Topologie nach der Konstruktionsvorschrift des verbesserten Modells von Watts und Strogatz [[Str01](#)] verwendet. Bei COHDA wirkt sich der Einfluss des Pseudozufallszahlengenerators lediglich auf die Struktur dieses Overlay-Netzes aus, und ist damit deutlich geringer als die im Wesentlichen auf Randomisierung basierende *Random Walk* Heuristik. Es wurden wiederum für jede Probleminstanz 100 Läufe durchgeführt. Auch hier ist der Trend zu größeren Fehlerwerten bei mittleren Werten für h , sowie eine Überlagerung von regelmäßigen Abweichungen dieses Trends zu sehen. Auffällig ist, dass die Abweichungen in negativer Richtung hier bis zu optimalen bzw. nahe-optimalen Fehlerwerten reichen. Eine mögliche Folgerung hieraus ist, dass das zielgerichtete Vorgehen der COHDA Heuristik im Vergleich zu zufallsbasierten Verfahren wie der *Random Walk* Heuristik von bestimmten Problemstrukturen profitiert und diese ausnutzen kann. Der Effekt wird an dieser Stelle jedoch nicht weiter untersucht, da die verwendeten Probleminstanzen ein Hilfsmittel darstellen und nicht im Fokus der Arbeit stehen. [Abbildung 7.4](#) (unten) zeigt die Simulationsdauern der einzelnen Durchläufe, d. h. die benötigte Anzahl an Simulationsschritten, bis sich alle Agenten auf einen gemeinsamen Lösungskandidaten geeinigt haben (vgl. [Abschnitt 4.3](#)). Der sichtbare Trend zu einer schnelleren Konsensbildung bei größeren Werten für h lässt sich durch folgende Argumentation erklären: Da die Verbundteilnehmer zu Beginn der Heuristik auf der Basis von unvollständigem Wissen agieren und noch nicht wissen, wie groß der Verbund ist bzw. welche Beiträge andere Verbundteilnehmer zur Lösung beitragen können, wählen sie von Beginn an tendenziell große Gewichtsvektoren aus ihrem Lösungsraum aus. Und da große Werte für h einen Zielvektor mit relativ großen Gewichten implizieren (siehe [Abbildung A.2](#) in [Anhang A](#)), verfolgen sie damit automatisch eine zielführende Strategie. Bei kleinen Werten für h hingegen, d. h. Zielvektoren mit kleinen Gewichten, sind die anfangs ausgewählten Gewichtsvektoren tendenziell zu groß. Sobald zusätzliches Wissen bzgl. anderer Verbundteilnehmer bei den Agenten eintrifft, werden die Gewichtsvektoren angepasst und in ihrer Größe nach unten korrigiert. Dieser erhöhte Koordinationsbedarf führt dann zu einer längeren Laufzeit des Verfahrens.

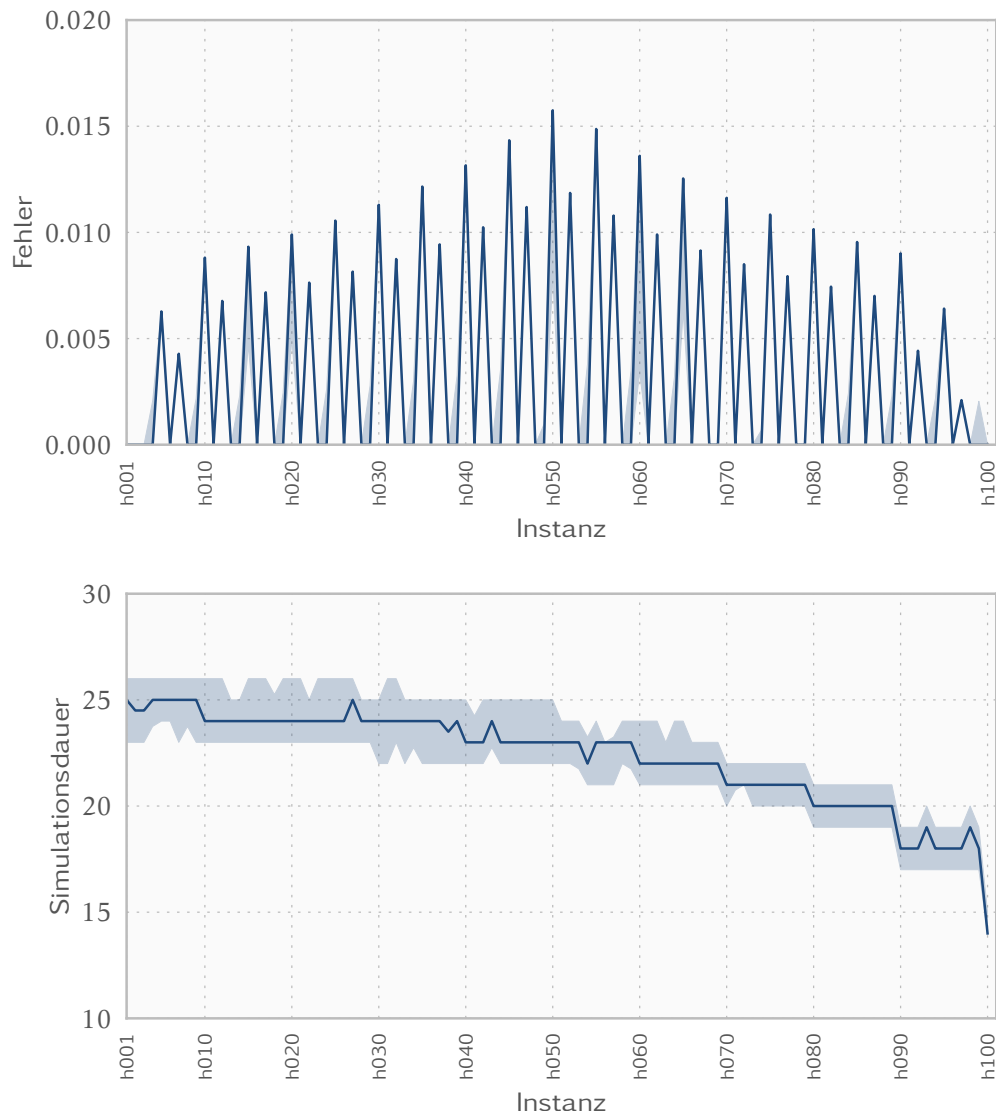


Abbildung 7.4: Verbleibender Fehler (oben) sowie Simulationsdauer (unten) für alle Instanzen $h \in [1, 100]$ bei Verwendung der COHDA Heuristik. Darstellung als Median $\pm$ oberes/unteres Quartil für 100 Ausführungen je Instanz h .

Als Ergebnis dieser Voruntersuchung wird festgehalten, dass die nachfolgende Evaluation der COHDA Heuristik anhand einer Probleminstanz mit einem kleinen Wert für h durchgeführt wird, da diese aufgrund des erhöhten Koordinationsaufwandes nicht-triviale Konfigurationen für das Verfahren darstellen. Konkret wird Probleminstanz $P(10, 5, 5)_{h=5}$ ausgewählt, da sie die Instanz mit kleinstem h darstellt, welche von der Heuristik statistisch betrachtet nicht bis zur Optimalität gelöst wird (der Median des Fehlers beträgt $\tilde{e} = 6.276 \cdot 10^{-3}$, mit einem Interquartilsabstand von $IQR = 4.742 \cdot 10^{-3}$).

7.2 LAUFZEITVERHALTEN

Auf Basis der im vorherigen Abschnitt als geeignet identifizierten Probleminstanz $P(m = 10, n = 5, q = 5)_{h=5}$ der Problemgruppe G-C(L)-D(S) soll die COHDA Heuristik im weiteren Verlauf im Detail evaluiert werden. Hierzu werden die in [Tabelle 7.1](#) vorgestellten Parameter nacheinander auf ihren Einfluss bzgl. Effizienz und Performanz des Verfahrens geprüft. Einleitend wird in diesem Abschnitt jedoch zunächst das grundlegende Laufzeitverhalten der Heuristik betrachtet. Für den initialen Lösungsvektor, d. h. die Menge der zu Beginn von den Agenten ausgewählten Gewichtsvektoren, wird der *worst-case* der Probleminstanz gewählt. Auf diese Weise wird die Konvergenz des Fehlerwertes über die Zeit von der schlechtesten Lösung ($e = 1$) in Richtung der optimalen Lösung ($e = 0$) deutlich. Da es sich bei COHDA um ein verteiltes System handelt, existiert allerdings keine Entität, die Wissen über die vom Gesamtsystem aktuell erreichte Lösungsgüte besitzt. Für die nachfolgenden Auswertungen wurde daher eine Beobachterkomponente in den Simulationskern integriert, welche lesenden Zugriff auf alle Attribute der Verbundteilnehmer sowie die ausgetauschten Nachrichten des Kommunikationsnetzes besitzt. [Abbildung 7.5](#) zeigt hierzu den Verlauf einer Simulation für die ausgewählte Probleminstanz. Das dieser Abbildung zugrundeliegende, von der Beobachterkomponente angefertigte Protokoll der Simulation ist exemplarisch in [Anhang A](#) nachzulesen. Auf der horizontalen Achse ist der zeitliche Verlauf der Simulation dargestellt. Ein Simulationsschritt entspricht dabei dem in [Definition 5.2.1](#) ([Abschnitt 5.2](#)) beschriebenen Zeitschritt. Der zu minimierende Fehler e gemäß [Gleichung 7.7](#) wird durch die linke vertikale

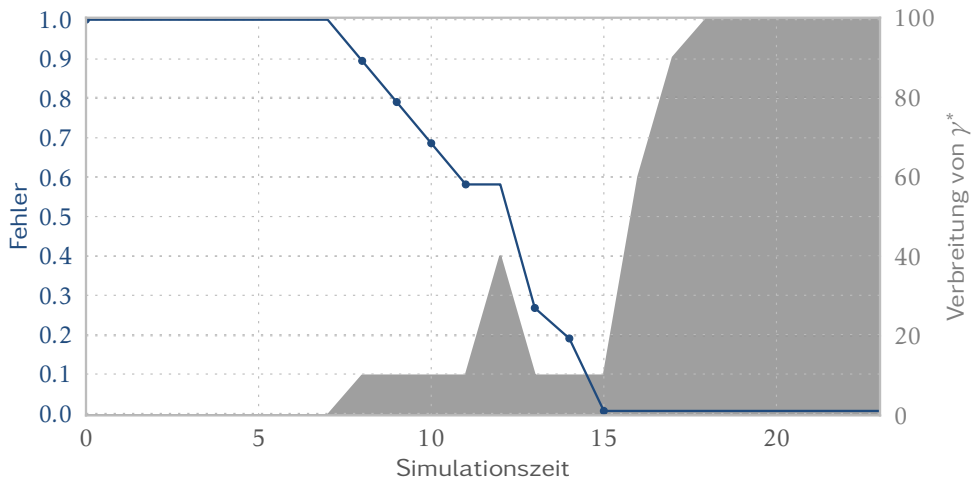


Abbildung 7.5: Laufzeitverhalten einer Simulation für $P(10, 5, 5)_{h=5}$.

Achse abgebildet. Die dazugehörige Datenreihe (durchgezogene Linie) wurde gebildet, indem zu jedem Simulationsschritt unter allen Verbundteilnehmern derjenige Lösungskandidat mit geringstem Fehler ausfindig gemacht wurde (in der Abbildung als γ^* bezeichnet). Die Datenreihe zeigt somit den Fehlerwert des zu einem Zeitpunkt jeweils „besten“ Lösungskandidaten im System. Die darauf liegenden Punkte markieren dabei die Zeitpunkte, zu denen ein neuer bester Lösungskandidat, d. h. mit noch geringerem Fehler als der zuvor identifizierte, von einem Verbundteilnehmer erzeugt wird. Zu Beginn der Simulation besitzen die Verbundteilnehmer nur sehr eingeschränktes Wissen, und können daher keine vollständigen Lösungskandidaten (siehe [Definition 3.6.6](#)) für das Optimierungsproblem erzeugen. Erst in Zeitschritt 8 besitzt (mindestens) ein Agent genug Wissen zur Bildung eines Lösungskandidaten. Von diesen, potenziell mehreren, Lösungskandidaten besitzt der beste einen Fehler von $e = 8.954 \cdot 10^{-1}$. In den nachfolgenden Zeitschritten 9, 10 und 11 werden durch Änderungen der Auswahl von Gewichtsvektoren sukzessive bessere Lösungskandidaten von Verbundteilnehmern im System erzeugt. In Zeitschritt 12 kann kein besserer Lösungskandidat gefunden werden – der bis dahin erreichte minimale Fehler von $e = 5.816 \cdot 10^{-1}$ bleibt zunächst konstant bestehen, bis in Zeitschritt 13, 14 und 15 erneut Verbesserungen auftreten. Letzterer stellt mit $e = 6.276 \cdot 10^{-3}$ die finale

Lösung des Verfahrens dar.

Parallel dazu ist in der Abbildung mit der schattierten Fläche die Verbreitung $d(\gamma^*)$ des jeweils besten Lösungskandidaten im System zu jedem Zeitpunkt dargestellt:

$$d(\gamma^*) = \frac{|\Gamma|}{|A|} \quad \text{mit } \Gamma = \{a_i \mid \gamma_i = \gamma^*\} \quad (7.8)$$

Hieran ist abzulesen, dass der in Zeitschritt 8 erzeugte Lösungskandidat zunächst nur einem einzelnen Verbundteilnehmer bekannt ist, sodass $d(\gamma^*) = 0.1$. Dies gilt ebenfalls für die nachfolgend in den Zeitschritten 9, 10 und 11 erzeugten Lösungskandidaten. Da in Zeitschritt 12 kein besserer Lösungskandidat gefunden wird und der bestehende Lösungskandidat γ^* aufgrund der Informationssynthese an die Nachbarn des erzeugenden Verbundteilnehmers kommuniziert wird, steigt die Verteilung dieses Lösungskandidaten in Zeitschritt 12 auf $d(\gamma^*) = 0.4$. Mit dem Aufkommen von wiederum besseren Lösungskandidaten in Zeitschritt 13, 14 und 15 sinkt die Verteilung wieder auf $d(\gamma^*) = 0.1$ ab. Die Terminierung der Heuristik ist nun anhand der monoton steigenden Verteilung ab Zeitschritt 15 zu sehen. Da der hier erzeugte Lösungskandidat von keinem Verbundteilnehmer weiter verbessert werden kann, breitet er sich im Verbund aus, bis er allen Agenten bekannt ist: $d(\gamma^*) = 1.0$ in Zeitschritt 18. Die Simulation läuft von diesem Zeitpunkt an noch solange, bis alle Nachrichten zugestellt wurden und kein Agent mehr neue Nachrichten erzeugt (vgl. [Abschnitt 4.3](#)). Dies ist hier in Zeitschritt 23 der Fall.

Das Ergebnis dieser Simulation ist in [Abbildung 7.6](#) visualisiert. Das obere Diagramm zeigt sowohl den Zielvektor der Problem Instanz (gestrichelte Linie), als auch den Lösungsvektor, der sich durch den finalen Lösungskandidaten γ^* nach Terminierung der Heuristik ergibt (durchgezogene Linie). Letzterer besitzt, wie oben bereits beschrieben, einen verbleibenden Fehler von $e = 6.276 \cdot 10^{-3}$ und ist daher nicht komplett deckungsgleich mit dem Zielvektor. Im vorliegenden Fall ist der Lösungsvektor in jeder Dimension etwas zu groß. Das untere Diagramm zeigt in einer gröber aufgelösten vertikalen Skale die aggregierten Gewichtsvektoren (jeder in einer eigenen Farbe), aus denen sich der Lösungsvektor zusammensetzt.

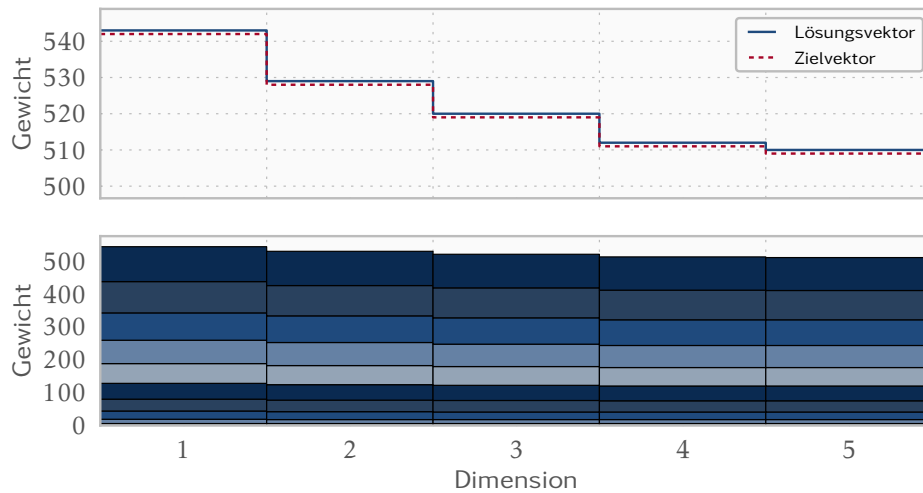


Abbildung 7.6: Resultat einer Simulation für $P(10, 5, 5)_{h=5}$.

7.3 ROBUSTHEIT

Nach der in [Abschnitt 7.2](#) erfolgten Beschreibung des grundlegenden Laufzeitverhaltens der Heuristik werden im Folgenden nun die Auswirkungen der Einflussfaktoren Topologie, Nachrichtenverzögerung, Reaktionszeit sowie Gesinnungswahl hinsichtlich der Robustheit des Verfahrens betrachtet. Dazu werden jeweils Qualitätsmerkmale für Performanz und Effizienz untersucht.

7.3.1 Topologie

Die für das Overlay-Netz verwendete Topologie wird von dem Verfahren nicht vorgeschrieben, sondern stellt einen prinzipiell frei wählbaren Parameter dar. Die Heuristik kann somit auch dann eingesetzt werden, wenn auf ein bestehendes Kommunikationsnetz mit fester Topologie aufgebaut werden soll. [Abbildung 7.7](#) zeigt die Ergebnisse einer Simulationsreihe mit verschiedenen Topologie-Typen. Hierbei kamen eine einfache Liste, ein Ring, ein rechtwinkliges Gitter, eine *Small-World* sowie ein vollvermaschtes Netz zum Einsatz. Für die Performanz des Verfahrens wurde der nach Terminierung der Heuristik verbleibende Fehler evaluiert. Es ist deutlich zu sehen, dass die Performanz nicht von der vorliegenden Topologie beeinflusst wird. Lediglich

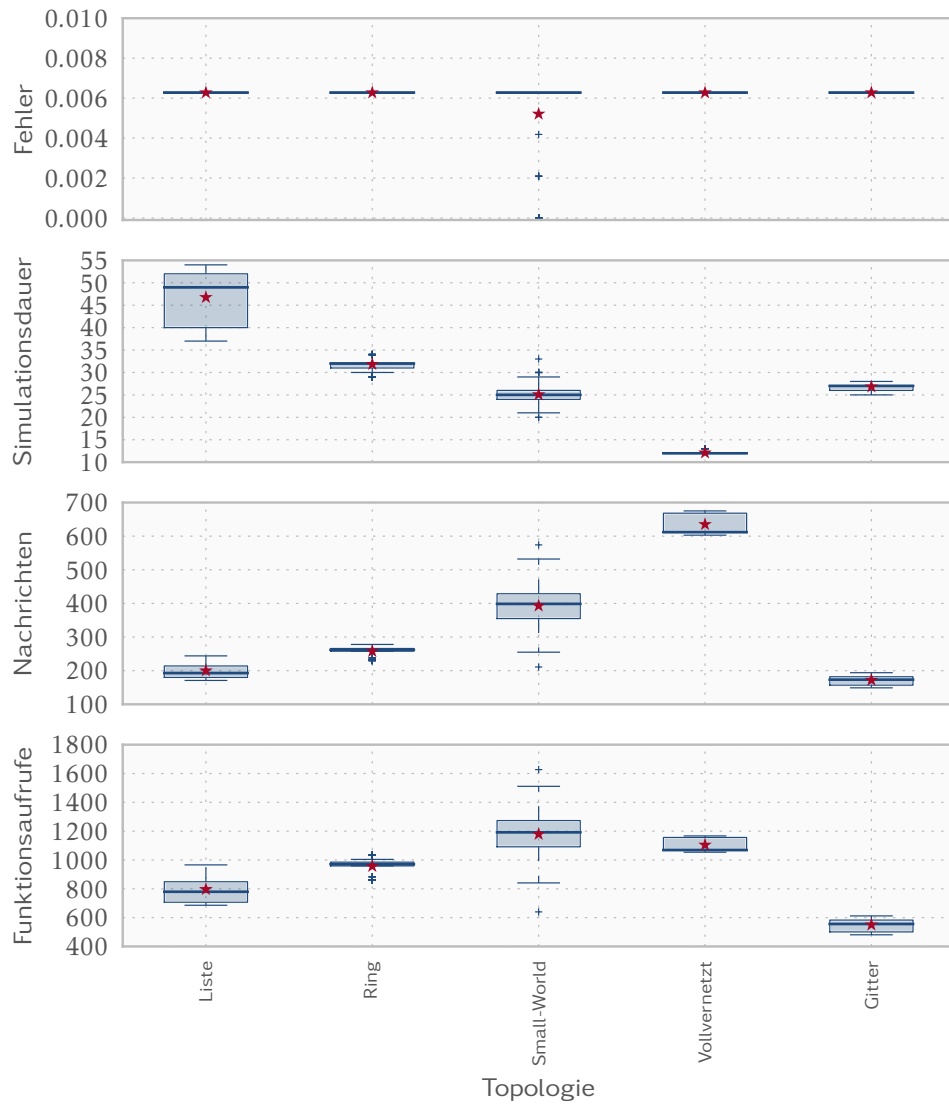


Abbildung 7.7: Verbleibender Fehler, Simulationsdauer, Kommunikationsaufwand und Rechenaufwand bei Verwendung verschiedener Topologien für das Overlay-Netz. Darstellung jeweils als Boxplot sowie Mittelwert (Sternsymbol) für 100 Ausführungen je Topologie.

einige wenige Ausreißer bei der *Small-World*-Topologie weichen von dem ansonsten konstanten Wert $\tilde{e} = 6.276 \cdot 10^{-3}$ ab. Dies ist auf die randomisiert gebildete Struktur der Topologie zurückzuführen, da sich hier bei den einzelnen Durchläufen jeweils ein leicht abweichendes Overlay-Netz ergibt, wodurch es in einigen Fällen möglich wird, das lokale Minimum $\tilde{e} = 6.276 \cdot 10^{-3}$ im Lösungsraum zu vermeiden.

Für die Effizienz des Verfahrens wurden die Simulationsdauer, die Anzahl der versendeten Nachrichten sowie die Anzahl der Auswertungen der Zielfunktion betrachtet. Im Vergleich von Simulationsdauer und Nachrichtenaufkommen ist festzustellen, dass sich mit höherem Vernetzungsgrad der Topologie prinzipiell eine schnellere Terminierung, aber zugleich ein höherer Kommunikationsbedarf ergibt. In der Reihenfolge Liste, Ring, *Small-World*, Vollvermaschung nimmt die Simulationsdauer ab, während das Nachrichtenaufkommen steigt. Eine Ausnahme hiervon bildet die Gitter-Topologie. Diese weist trotz mittlerer Simulationsdauer nur sehr wenig Nachrichten auf. Dieser positive Effekt ist jedoch auf kleine Problemgrößen beschränkt – bei größeren Instanzen zeigt sich wiederum der entgegengesetzte Verlauf von Simulationsdauer und Nachrichtenaufkommen wie bei den anderen Topologien. [Abbildung A.4](#) in [Anhang A](#) stellt hierzu einen Vergleich der *Small-World* mit der Gitter-Topologie für verschiedene Verbundgrößen m dar. Die Anzahl der Zielfunktionsauswertungen als Maß für den Berechnungsaufwand des Verfahrens verhält sich analog dazu und steigt ebenfalls mit dem Vernetzungsgrad, da jede versendete Nachricht eine Reaktion in den jeweiligen Nachbarn auslöst. Dies gilt jedoch nur bis zu einem gewissen Grad. Mit sehr hoher Dichte der Topologie sinkt die Zahl der Zielfunktionsauswertungen wieder, was sich auf die reduzierte Simulationsdauer zurückführen lässt: Wird diese Kennzahl bzgl. der Simulationsdauer normiert, ergibt sich der gleiche Verlauf wie bei dem Nachrichtenaufkommen.

Die identifizierten Zusammenhänge bzgl. des Vernetzungsgrades der Topologie sind in [Abbildung 7.8](#) nochmals deutlich nachvollziehbar. Hier wurde die *Small-World* Topologie mit unterschiedlichen Werten für den Dichteparameter ϕ evaluiert. Die gegensätzlichen Verläufe von Simulationsdauer und Nachrichtenaufkommen sind klar erkennbar, ebenso der oben beschriebene Verlauf der Zielfunktionsaufrufe bei zunehmendem Vernetzungsgrad. Ergänzend zu dieser qualitativen Einschätzung wird später in [Abschnitt 7.5](#) eine Regressionsanalyse durchgeführt, um den funktionalen Zusammenhang zwischen Vernetzungsgrad und den Effizienzparametern zu schätzen.

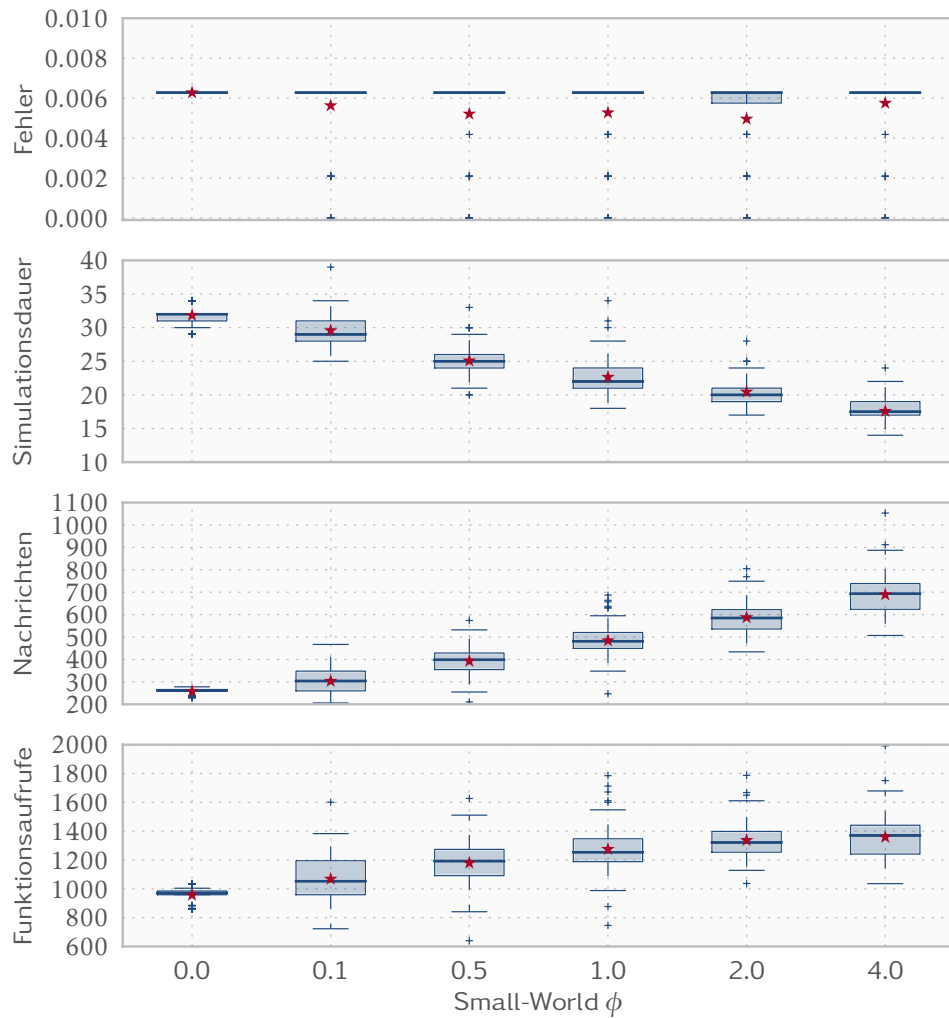


Abbildung 7.8: Verbleibender Fehler, Simulationsdauer, Kommunikationsaufwand und Rechenaufwand bei Verwendung verschiedener Werte für den *Small-World* Dichteparameter ϕ . Darstellung jeweils als Boxplot sowie Mittelwert (Sternsymbol) für 100 Ausführungen je Wert.

Das wesentliche Ergebnis der Evaluation des Topologieparameters ist die Unabhängigkeit der Performanz des Verfahrens von der verwendeten Topologie. Für die Effizienz ist festzuhalten, dass sich abhängig vom Vernetzungsgrad die Laufzeit und das Nachrichtenaufkommen gegensätzlich entwickeln. Die Topologie kann somit als Tuning-Parameter der Heuristik betrachtet werden, mit welchem eine individuelle Präferenz für Laufzeit oder Nachrichtenaufkommen realisiert werden kann. Dabei ist zu beachten, dass der Berechnungsaufwand, durch die Anzahl von Zielfunktionsauswertungen ausgedrückt, bei mittleren bis hohen Vernetzungsgraden maximal wird.

7.3.2 Nachrichtenverzögerung

Die Definition des Kommunikationsnetzes in [Abschnitt 3.3](#) beinhaltet verzögerungsbehaftete Übertragungskanäle, um die Robustheit des Verfahrens bei Verwendung realer Kommunikationsnetze sicherzustellen. In [Kapitel 5](#) wurde bereits gezeigt, dass die Konvergenz des Verfahrens auch bei Nachrichtenverzögerungen garantiert ist. Um nun zusätzlich die Performanz und Effizienz des Verfahrens unter diesen Bedingungen zu bewerten, wurden Simulationen mit verschiedenen Nachrichtenverzögerungen durchgeführt. [Abbildung 7.9](#) zeigt die Ergebnisse dieses Experimentes. Die angegebene Nachrichtenverzögerung ist dabei als Maximalwert zu verstehen: Ein Wert $d_{\max} = 4$ bedeutet beispielsweise, dass jede versendete Nachricht μ mit einer zufälligen Verzögerung $d_{\mu} \in [1, 4]$ belegt wurde. Anhand der konstanten Obergrenze von $\tilde{\epsilon} = 6.276 \cdot 10^{-3}$ des nach Terminierung der Heuristik verbleibenden Fehlers ist zu erkennen, dass das Verfahren in seiner Performanz robust gegen Nachrichtenverzögerungen ist. Lediglich durch die verwendete *Small-World* Topologie bedingt ergeben sich einige Ausreißer in Richtung optimaler Lösungen (vgl. dazu die Evaluation der Topologietypen in [Abschnitt 7.3.1](#)). Die Simulationsdauer wächst konsequenterweise mit steigender Nachrichtenverzögerung. Das Nachrichtenaufkommen sowie die Anzahl der Zielfunktionsaufrufe scheinen bereits sehr früh kein nennenswertes Wachstum mehr aufzuweisen und sind offenbar kaum von der Nachrichtenverzögerung abhängig. Konkretere Aussagen hierzu wird die Regressionsanalyse in [Abschnitt 7.5](#) liefern.

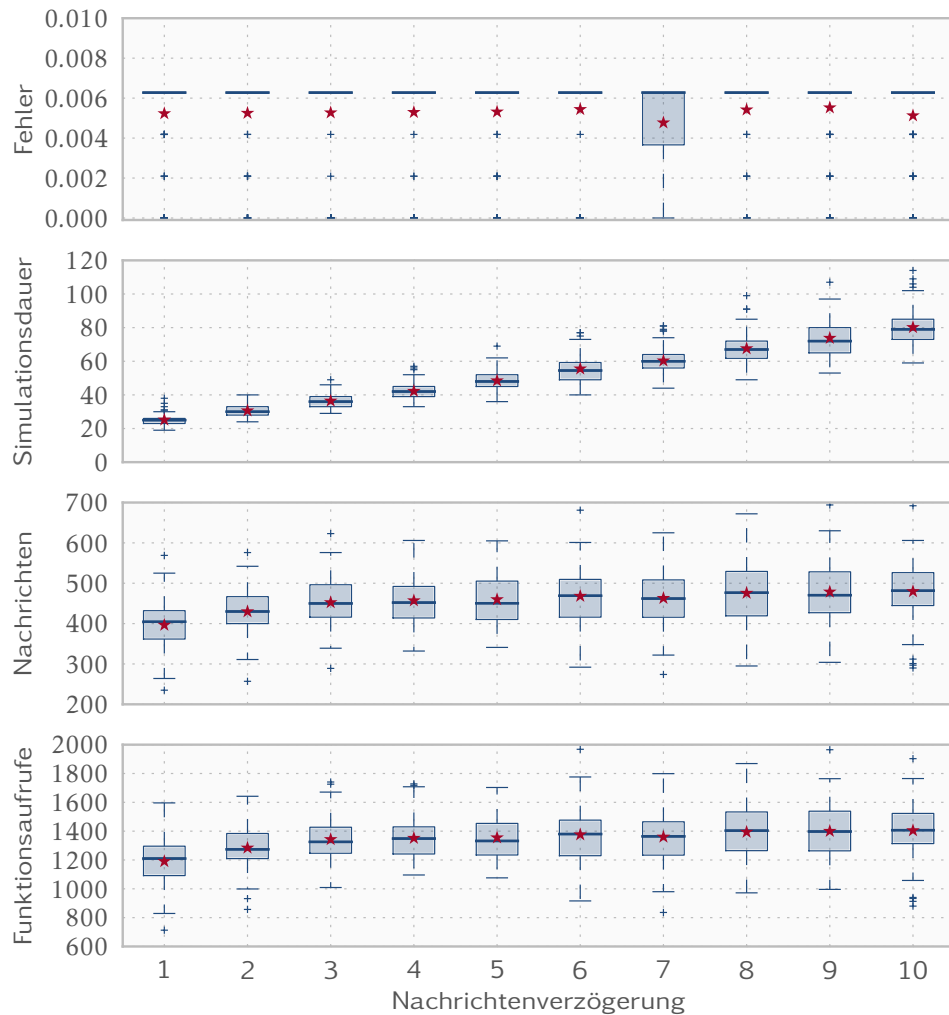


Abbildung 7.9: Verbleibender Fehler, Simulationsdauer, Kommunikationsaufwand und Rechenaufwand bei Simulation verschiedener maximaler Nachrichtenverzögerungen für das Kommunikationsnetz. Darstellung jeweils als Boxplot sowie Mittelwert (Sternsymbol) für 100 Ausführungen je maximaler Nachrichtenverzögerung.

7.3.3 Reaktionszeit

In [Abschnitt 6.3](#) wurde eine agentenspezifische Reaktionszeit r_i eingeführt. Analog zur Nachrichtenverzögerung im vorhergehenden Abschnitt sollen nun die Auswirkungen der Reaktionszeit auf Effizienz und Performanz des Verfahrens untersucht werden. Hierzu wurden Simulationen mit verschiedenen Werten für die Reaktionszeiten der Verbundteilnehmer durchgeführt. In [Abbildung 7.10](#) sind die Ergebnisse dieser Simulationsreihe zu sehen. Die auf der horizontalen Achse dargestellte Reaktionszeit repräsentiert wiederum einen Maximalwert $r_{\max}$ und ist wie folgt zu interpretieren: Betritt ein Agent den Zustand `sleep( $r_i$ )` (vgl. [Abbildung 6.1](#)), so wird eine zufällige Anzahl von Simulationsschritten $r_i \in [1, r_{\max}]$ bestimmt, die der Agent in diesem Zustand verweilt. Während dieser Zeitspanne kann er zwar Nachrichten empfangen und abspeichern, diese jedoch nicht weiter bearbeiten. Erst nach Ablauf der Wartezeit wird der Agent aktiv und führt die Schritte *Wahrnehmung*, *Entscheidung* und *Aktion* aus. Anschließend betritt er wieder den Zustand `sleep( $r_i$ )` und verfährt wie beschrieben.

Die Simulationsergebnisse zeigen, dass das Verfahren hinsichtlich Performanz robust gegen randomisierte Reaktionszeiten ist. Die bereits in den vorherigen Abschnitten identifizierte Obergrenze des Fehlers von $\tilde{\epsilon} = 6.276 \cdot 10^{-3}$ für die betrachtete Problemistanz wird auch hier nicht überschritten. Die Effizienzindikatoren zeichnen dabei ein interessantes Bild: Während die Simulationsdauer mit steigenden Reaktionszeiten wächst, ist für das Nachrichtenaufkommen als auch den Berechnungsaufwand ein sinkender Verlauf zu erkennen. Trotz längerer Simulationsdauer (und gleichbleibender Topologie des Overlay-Netzes) werden also bei größeren Reaktionszeiten über den gesamten Simulationsverlauf insgesamt weniger Nachrichten verschickt und weniger Zielfunktionsauswertungen durchgeführt. Damit kann das Verfahren auch bzgl. seiner Effizienz als robust gegen randomisierte Reaktionszeiten betrachtet werden. Zudem wird dadurch das bereits in [\[And+12\]](#) gefundene Ergebnis des positiven Einflusses von Abweichungen in den Verhaltensweisen von homogenen Agenten (engl. *inter-agent variation*) auf den Selbstorganisationsmechanismus des Systems nochmals bestätigt. Für den konkreten funktionalen Zusammenhang zwischen der Reaktionszeit und den Effizienzmerkmalen sei wiederum auf [Abschnitt 7.5](#) verwiesen.

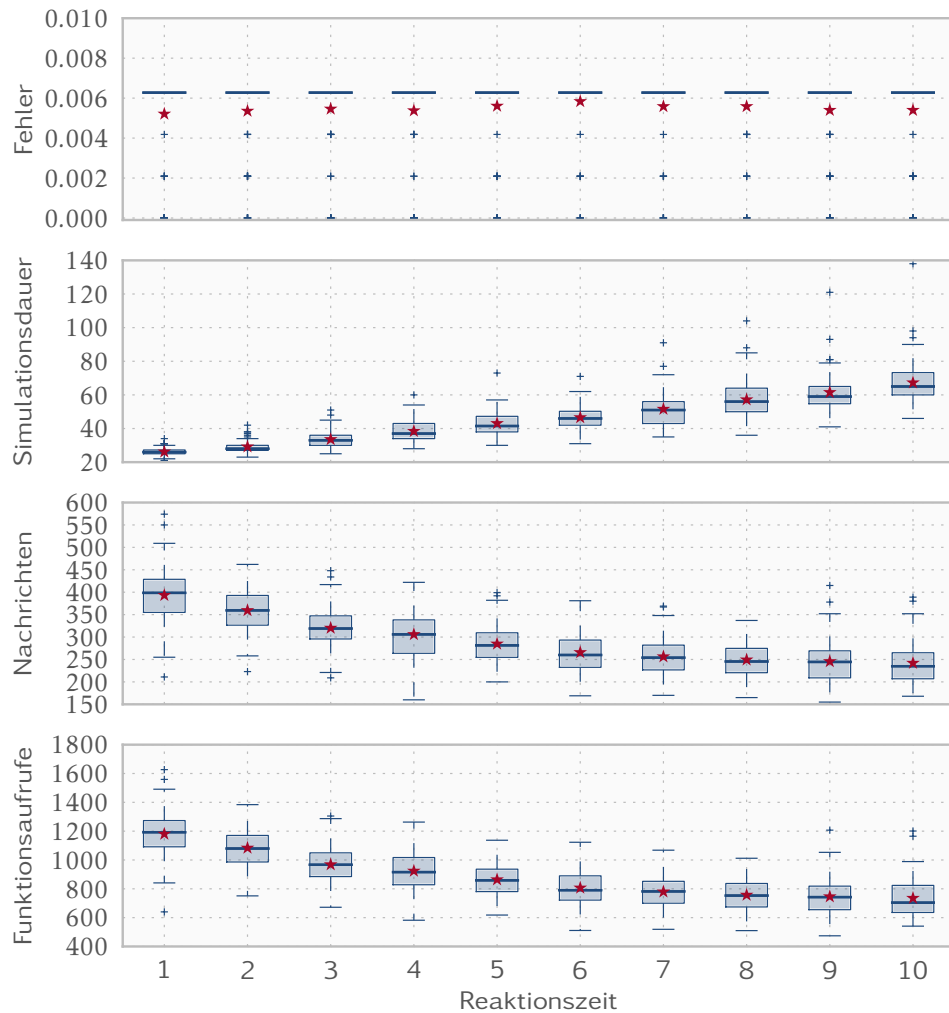


Abbildung 7.10: Verbleibender Fehler, Simulationsdauer, Kommunikationsaufwand und Rechenaufwand bei Simulation verschiedener maximaler Reaktionszeiten der Verbundteilnehmer. Darstellung jeweils als Boxplot sowie Mittelwert (Sternsymbol) für 100 Ausführungen je maximaler Reaktionszeit.

7.3.4 Freie Gesinnungswahl

In [Abschnitt 3.6.4](#) wurde die freie Gesinnungswahl für Verbundteilnehmer modelliert. Diese erlaubt es einem Verbundteilnehmer in seinem Entscheidungsprozess, die Fahrpläne in seinem lokalen Suchraum hinsichtlich privater Kriterien zu bewerten und darauf basierend beliebige Fahrpläne von der Einsatzplanung auszuschließen. Hat er einen Fahrplan aber einmal als gültig und akzeptabel bewertet und diesen als realisierbare Lösung an seine Nachbarn kommuniziert, so darf er diesen Fahrplan im Laufe des Prozesses nicht nachträglich von der Einsatzplanung ausschließen. In [Abschnitt 5.3](#) wurde bereits nachgewiesen, dass die Konvergenz des Verfahrens auch bei freier Gesinnungswahl garantiert ist. Im Folgenden soll nun der Einfluss der Gesinnungswahl auf die Performanz und Effizienz des Verfahrens untersucht werden. Dazu werden die Parameter ψ_{stat} und ψ_{dyn} eingeführt, über welche verschiedene Ausprägungen der Gesinnungswahl simuliert werden können. Beide Parameter können den Wertebereich $[0, 1)$ annehmen und modellieren prinzipiell die maximale Wahrscheinlichkeit, mit welcher ein Verbundteilnehmer im Laufe der Heuristik einen Fahrplan aus seinem Lösungsraum als ungeeignet markiert und von der Lösungsfindung ausschließt. Die Parameter unterscheiden sich dabei in dem Zeitpunkt ihrer Anwendung: ψ_{stat} kommt ausschließlich während der Konstruktion der Probleminstanz P zum Einsatz. Dabei wird für jedes Element (= Fahrplan) jeder Klasse (= Verbundteilnehmer, vgl. [Abschnitt 7.1.1](#)) mit der Wahrscheinlichkeit ψ_{stat} entschieden, dass das Element aus der Klasse gelöscht wird. Damit steht das Element für den gesamten Lösungsfindungsprozess nicht zur Verfügung. Dies kann bspw. dann auftreten, wenn ein Verbundteilnehmer zu Beginn der Einsatzplanung Teile seines prinzipiell möglichen Handlungsspielraums etwa aufgrund von Nutzerpräferenzen von der weiteren Betrachtung ausschließt. ψ_{dyn} hingegen wird erst zur Laufzeit ausgewertet und soll exemplarisch dynamische Entscheidungsfunktionen modellieren, in welchen bspw. aus strategischen Gründen (wie etwa die persönliche Gewinnmaximierung) zu bestimmten Zeitpunkten mehr oder weniger Elemente des theoretisch möglichen Handlungsspielraumes zugelassen werden. Da die vorliegende Arbeit von konkreten individuellen Zielsetzungen der Verbundteilnehmer abstrahiert, werden diese im Rahmen der Evaluation als stochastische Entscheidungsfunktionen modelliert. In jeder stattfindenden Entscheidungsphase eines Verbundteilnehmers wird daher für ein aus dem Lösungsraum ausgewähltes Element mittels der Wahrschein-

lichkeit ψ_{dyn} entschieden, dass das Element verworfen werden soll. Dabei bleibt es jedoch im Lösungsraum enthalten, sodass es vorkommen kann, dass ein vormals verworfenes Element in einer späteren Entscheidungsfindung von dem Verbundteilnehmer doch zugelassen wird. Für die vorliegende Evaluation werden beide Parameter getrennt ausgewertet, und dazu jeder Parameter über seinen Wertebereich in Schritten von 0.1 variiert. [Abbildung 7.11](#) zeigt die Simulationsergebnisse für ψ_{stat} . Die bereits in [Abschnitt 5.3](#) postulierte Beeinträchtigung der finalen Lösungsgüte aufgrund der durch ψ_{stat} induzierten Reduktion des Suchraumes ist deutlich am Verlauf des verbleibenden Fehlers zu erkennen. Je höher die Wahrscheinlichkeit, dass ein Element bereits zum Zeitpunkt der Szenariobildung von dem späteren Suchprozess ausgeschlossen wird, desto kleiner wird die Möglichkeit, optimale und nahe-optimale Lösungen zu produzieren. Durch die damit verbundene reduzierte Komplexität des gestellten Problems sinken gleichfalls die Anzahl der benötigten Simulationsschritte, Nachrichten sowie Zielfunktionsaufrufe.

Im Vergleich dazu zeigt [Abbildung 7.12](#) die Simulationsergebnisse für ψ_{dyn} . Mit Ausnahme von sehr großen Werten für ψ_{dyn} (d. h. es werden in den einzelnen Entscheidungsprozessen sehr viele Fahrpläne verworfen) scheint die Performanz des Verfahrens nur sehr wenig von der dynamischen Gesinnungswahl beeinträchtigt zu sein. Dies ist mit einer erhöhten Laufzeit sowie einem größeren Nachrichtenaufkommen verbunden. ψ_{dyn} stellt damit bei moderaten Ausprägungen lediglich einen Störfaktor für das System dar. Da die jeweils verworfenen Fahrpläne in einem späteren Entscheidungsprozess eines Verbundteilnehmers erneut zur Verfügung stehen und potenziell gewählt werden können, ist das System in der Lage, diesen Umstand mit erhöhtem Koordinationsaufwand zu kompensieren. Wenn ψ_{dyn} allerdings zu groß wird, und damit ein Fahrplan auch bei Wiedervorlage mit hoher Wahrscheinlichkeit wieder verworfen wird, terminiert die Heuristik mangels Handlungsoptionen mit einem relativ großen verbleibenden Fehler. Interessanterweise sinkt die Anzahl der Zielfunktionsaufrufe in diesen Fällen. Dies ist offenbar auf die geringere Anzahl an produzierten Lösungskandidaten im System zurückzuführen, wodurch die einzelnen Verbundteilnehmer seltener zwischen einem bestehenden und einem empfangenen Lösungskandidaten entscheiden müssen. Ein weiterer hervorzuhebender Effekt wird in [Abbildung 7.13](#) deutlich. Hier ist der verbleibende Fehler aus der vorherigen Grafik auf einer kleineren Skala dargestellt, um die Unterschiede bei den Ausprägungen $\psi_{\text{dyn}} < 0.8$ zu visualisieren. Während das Fehlen der freien Gesin-

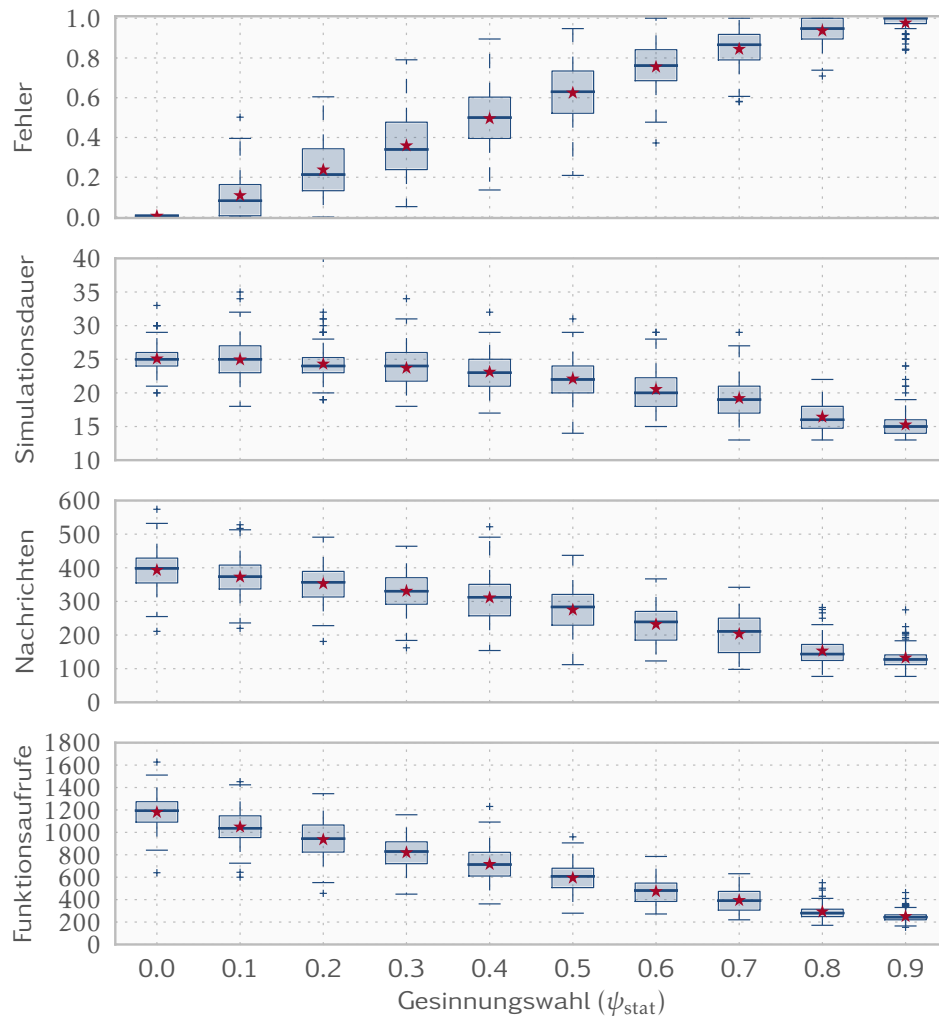


Abbildung 7.11: Verbleibender Fehler, Simulationsdauer, Kommunikationsaufwand und Rechenaufwand bei Simulation verschiedener Ausprägungen des statischen Gesinnungswahlparameters. Darstellung jeweils als Boxplot sowie Mittelwert (Sternsymbol) für 100 Ausführungen je Wert.

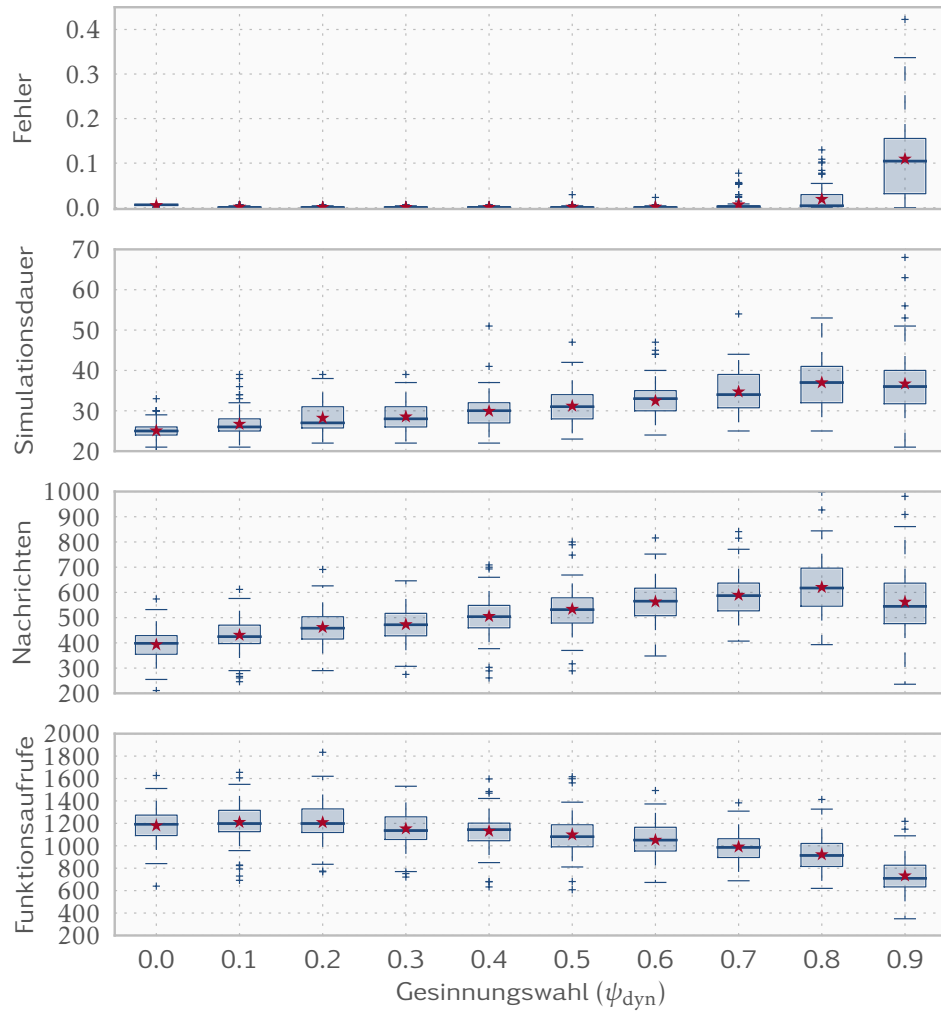


Abbildung 7.12: Verbleibender Fehler, Simulationsdauer, Kommunikationsaufwand und Rechenaufwand bei Simulation verschiedener Ausprägungen des dynamischen Gesinnungswahlparameters. Darstellung jeweils als Boxplot sowie Mittelwert (Sternsymbol) für 100 Ausführungen je Wert.

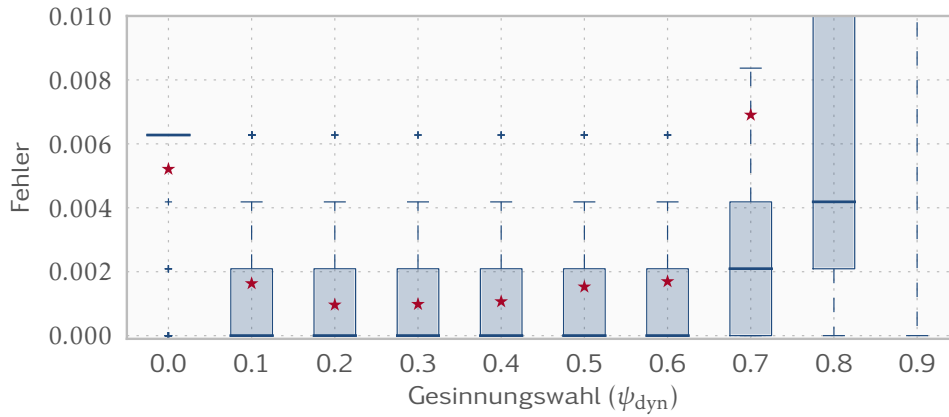


Abbildung 7.13: Detailansicht des verbleibenden Fehlers bei Simulation verschiedener Ausprägungen des dynamischen Gesinnungswahlparameters. Darstellung jeweils als Boxplot sowie Mittelwert (Sternsymbol) für 100 Ausführungen je Wert.

nungswahl ($\psi_{\text{dyn}} = 0$) zu einem verbleibenden Fehler von $\tilde{e} = 6.276 \cdot 10^{-3}$ führt (wie weiter oben bereits angedeutet, handelt es sich hierbei um ein lokales Minimum im Lösungsraum der betrachteten Problem Instanz), weisen die Ausprägungen $0.1 \leq \psi_{\text{dyn}} \leq 0.6$ deutlich kleinere verbleibende Fehler auf. Offenbar führt die durch ψ_{dyn} eingeführte Variation in den Entscheidungsprozessen dazu, dass die Heuristik das lokale Optimum leichter in Richtung besserer Lösungen verlassen kann. Dies ist erneut ein Hinweis auf den positiven Einfluss von *inter-agent variation* (siehe vorherigen Abschnitt).

7.4 SKALIERBARKEIT

Die betrachtete Problem Instanz $P(10, 5, 5)_{h=5}$ schreibt eine Verbundgröße von $m = 10$ Einheiten, eine Suchraumgröße von $n = 5$ Elementen sowie einen Planungshorizont von $q = 5$ Intervallen vor. Für die Evaluation der Skalierbarkeit werden diese Parameter individuell variiert. Hierdurch ergibt sich jedoch je Konfiguration eine individuelle Problem Instanz, z. B. entsteht $P(100, 10, 5)_{h=5}$ bei Verwendung von

$m = 100$. Die Performanz- und Effizienzindikatoren können dann nicht mehr separat über verschiedene Konfigurationen hinweg miteinander verglichen werden, da jede Konfiguration eine individuelle Problemistanz repräsentiert. Stattdessen müssen Kennzahlen verwendet werden, welche von der konkreten Problemistanz unabhängige Aussagen, und damit Vergleiche der einzelnen Konfigurationen zulassen. Hierzu ist es erforderlich, sowohl von den Instanzparametern m , n und q zu abstrahieren, als auch die Performanz- und Effizienzindikatoren in Relation zueinander zu betrachten, um den Einfluss der für eine Problemistanz spezifischen Struktur des Lösungsraumes (vgl. [Abschnitt 7.1.2](#)) zu eliminieren. Damit ergeben sich die folgenden Kennzahlen:

Definition 7.4.1 (Normierte Laufzeiteffizienz)

Sei c gemäß [Definition 5.2.2](#) und [Definition 5.2.4](#) eine terminierende Ausführungsfolge für eine gegebene Problemistanz $P(m, n, q)$, und bezeichne $\#steps$ die Anzahl der in c enthaltenen Zeitschritte. Sei weiterhin s der durch c erzeugte Lösungsvektor für $P(m, n, q)$. Dann beschreibt die Kennzahl

$$eff_{steps} = \frac{1 - e(s)}{\#steps} \quad (7.9)$$

die Laufzeiteffizienz von c in Abhängigkeit von der erreichten normierten Lösungsgüte. Es gilt $eff_{steps} \in [0, 1]$, wobei eine maximale Laufzeiteffizienz mit $eff_{steps} = 1$ genau dann gegeben ist, wenn s einen Fehler $e(s) = 0$ besitzt und in $\#steps = 1$ Zeitschritt gefunden wurde. Sowohl mit wachsendem Fehler, als auch mit längerer Laufzeit sinkt die Laufzeiteffizienz.

Definition 7.4.2 (Normierte Kommunikationseffizienz)

Sei c gemäß [Definition 5.2.2](#) und [Definition 5.2.4](#) eine terminierende Ausführungsfolge für eine gegebene Problemistanz $P(m, n, q)$, und bezeichne $\#msgs$ die Anzahl der in c insgesamt versendeten Nachrichten. Sei weiterhin s der durch c erzeugte Lösungsvektor für $P(m, n, q)$. Dann beschreibt die Kennzahl

$$eff_{msgs} = \frac{m \cdot (1 - e(s))}{\#msgs} \quad (7.10)$$

die Kommunikationseffizienz von c pro Agent und in Abhängigkeit von der er-

reichten normierten Lösungsgüte. Es gilt $\text{eff}_{\text{msgs}} \in [0, 1]$, wobei eine maximale Kommunikationseffizienz mit $\text{eff}_{\text{msgs}} = 1$ genau dann gegeben ist, wenn $\mathbf{s}$ einen Fehler $e(\mathbf{s}) = 0$ besitzt und dazu $\#\text{msgs} = m$ Nachrichten versendet wurden (d. h. von jedem Agenten genau eine). Sowohl mit wachsendem Fehler, als auch mit höherem Nachrichtenaufkommen sinkt die Kommunikationseffizienz.

Definition 7.4.3 (Normierte Berechnungseffizienz)

Sei c gemäß [Definition 5.2.2](#) und [Definition 5.2.4](#) eine terminierende Ausführungsfolge für eine gegebene Problem Instanz $P(m, n, q)$, und bezeichne $\#\text{calc}$ die Anzahl der in c insgesamt durchgeführten Zielfunktionsauswertungen. Sei weiterhin $\mathbf{s}$ der durch c erzeugte Lösungsvektor für $P(m, n, q)$. Dann beschreibt die Kennzahl

$$\text{eff}_{\text{calc}} = \frac{m \cdot (1 - e(\mathbf{s}))}{\#\text{calc}} \quad (7.11)$$

die Berechnungseffizienz von c pro Agent und in Abhängigkeit von der erreichten normierten Lösungsgüte. Es gilt $\text{eff}_{\text{calc}} \in [0, 1]$, wobei eine maximale Berechnungseffizienz mit $\text{eff}_{\text{calc}} = 1$ genau dann gegeben ist, wenn $\mathbf{s}$ einen Fehler $e(\mathbf{s}) = 0$ besitzt und dazu $\#\text{calc} = m$ Funktionsauswertungen durchgeführt wurden (d. h. von jedem Agenten genau eine). Sowohl mit wachsendem Fehler, als auch mit höherem Berechnungsaufwand sinkt die Berechnungseffizienz.

Mit diesen normierten Kennzahlen können nun Konfigurationen direkt miteinander verglichen werden, die sich in den Werten für die Instanzparameter m , n und q unterscheiden. Eine weitere Problematik ergibt sich jedoch aus der Verwendung des normierten Fehlers e . Für die Normierung ist es erforderlich, die besten und schlechtesten Gütwerte v_{best} und v_{worst} zu kennen, welche für eine gegebene Problem Instanz auftreten können. Da eine exakte Berechnung dieser Werte vergleichsweise rechenintensiv ist (bereits die kleinste betrachtete Instanz $P(10, 5, 5)$ weist eine Menge von $n^m = 5^{10} = 9\,765\,625$ möglichen Lösungen auf, vgl. [Abschnitt 7.1.2](#)), und für jede Problem Instanz separat hätte bestimmt werden müssen, wurde stattdessen eine Abschätzung vorgenommen. Für alle Instanzen wurde angenommen, dass ein Lösungsvektor $\mathbf{s}$ existiert, welcher dem Zielvektor $\mathbf{z}$ exakt entspricht, sodass $e = 0 \iff \mathbf{z} = \mathbf{s}$. Zur Abschätzung der schlechtesten Lösung für $e = 1$ wurde eine einfache Heuristik angewendet, in der die bzgl. der Summennorm größten und

kleinsten Elemente jeder Klasse jeweils kombiniert werden, und schließlich aus diesen zwei Kombinationen diejenige mit schlechterer Güte ausgewählt wird (siehe [Algorithmus 4](#)).

7.4.1 Verbundgröße

Der Instanzparameter m bestimmt die Anzahl der Klassen, aus denen jeweils ein Element für eine gültige Lösung des Optimierungsproblemekes gewählt werden muss. Dies entspricht im Anwendungskontext der Größe des Verbundes A . Für die vorliegende Studie wurde $m \in \{10, 20, 30, 40, 50, 100, 500\}$ untersucht, das Ergebnis dieser Simulationsreihe ist in [Abbildung 7.14](#) dargestellt. Für jede Simulation wurden die normierten Kennzahlen aus den Definitionen [7.4.1](#), [7.4.2](#) und [7.4.3](#) berechnet. Die Abbildung zeigt die statistische Verteilung dieser Kennzahlen über die durchgeführten Simulationen je Verbundgröße. Im Gegensatz zu allen anderen Konfigurationen konnten für $m = 500$ nur 10 Simulationen durchgeführt werden. Die geringe Streuung der Werte in dieser Konfiguration deutet jedoch darauf hin, dass diese Zahl für die vorliegende Auswertung ausreicht. Die Auswertung zeigt, dass sowohl die Laufzeit-, die Kommunikations- als auch die Berechnungseffizienz des Verfahrens von der vorliegenden Problemgröße in Form der Anzahl der Verbundteilnehmer abhängig ist: Mit wachsender Verbundgröße sinkt die Effizienz in jedem dieser drei Bereiche, wobei jeweils ein asymptotischer Verlauf zu erkennen ist (dabei ist zu beachten, dass die horizontale Achse als Ordinalskala formatiert ist, woraus sich der optische „Knick“ zwischen $m = 50$, $m = 100$ und $m = 500$ ergibt). Dies ist darauf zurückzuführen, dass bei einer unendlichen Verbundgröße niemals eine Lösung gefunden werden kann, sodass gelten muss:

$$\lim_{m \rightarrow \infty} \text{eff}_{\text{steps}} = \lim_{m \rightarrow \infty} \text{eff}_{\text{msgs}} = \lim_{m \rightarrow \infty} \text{eff}_{\text{calc}} = 0$$

Eine genauere Analyse der funktionalen Zusammenhänge zwischen Verbundgröße und Effizienz ist in [Abschnitt 7.5](#) zu finden. Die den dargestellten Effizienz-Kennzahlen zugrundeliegenden Rohdaten (Fehler, Simulationsdauer, Nachrichtenaufkommen und Zielfunktionsauswertungen) sind der Vollständigkeit halber in [Anhang A](#) abgebildet.

Algorithmus 4 Heuristik zur Abschätzung der schlechtesten Lösung für eine Problem-
 instanz $P(m, n, q)$

Parameter:

m	▷ Anzahl Klassen
n	▷ Anzahl Elemente pro Klasse
q	▷ Anzahl Dimensionen pro Element

Funktionen:

$f(s)$	▷ Berechnet Güte für Lösung s analog zu Definition 3.1.11
--------	---

```

1:  $W_{\min} \leftarrow \text{array}[1..m]$ 
2:  $W_{\max} \leftarrow \text{array}[1..m]$ 
3: for  $i \leftarrow 1, m$  do
4:    $size_{\min} \leftarrow \text{NONE}$ 
5:    $size_{\max} \leftarrow \text{NONE}$ 
6:   for  $j \leftarrow 1, n$  do
7:      $size \leftarrow 0$ 
8:     for  $k \leftarrow 1, q$  do
9:        $size \leftarrow size + w_{ijk}$ 
10:    end for
11:    if  $size_{\min}$  is NONE or  $size < size_{\min}$  then
12:       $size_{\min} \leftarrow size$ 
13:       $W_{\min}[i] \leftarrow w_{ij}$ 
14:    end if
15:    if  $size_{\max}$  is NONE or  $size > size_{\max}$  then
16:       $size_{\max} \leftarrow size$ 
17:       $W_{\max}[i] \leftarrow w_{ij}$ 
18:    end if
19:  end for
20: end for
21: return  $\max(f(W_{\min}), f(W_{\max}))$ 

```

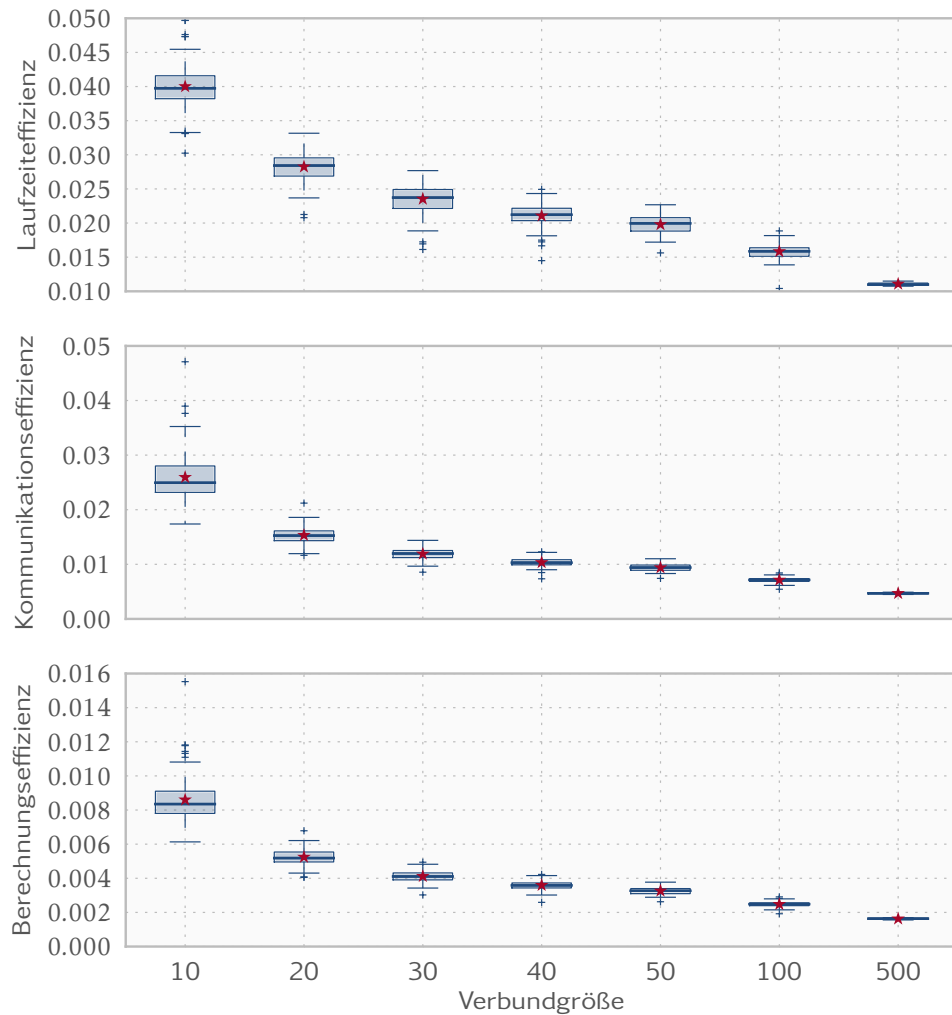


Abbildung 7.14: Laufzeiteffizienz, Kommunikationseffizienz und Berechnungseffizienz bei Simulation verschiedener Verbundgrößen. Darstellung jeweils als Boxplot sowie Mittelwert (Sternsymbol) für 100 Ausführungen je Verbundgröße.

7.4.2 Suchraumgröße

Mit dem Instanzparameter n wird die Anzahl der Elemente pro Klasse des Optimierungsproblems festgelegt. Im Anwendungskontext ist dies die Größe der Suchräume der Verbundteilnehmer, und entspricht damit der Anzahl möglicher Fahrpläne, aus denen der optimale gefunden werden muss. Im vorliegenden Fall wurde $n \in \{5, 10, 100, 1000, 10000\}$ evaluiert, [Abbildung 7.15](#) zeigt das Ergebnis der Simulationsreihe. Analog zu vorherigem Abschnitt wurden die Verteilungen der oben eingeführten Kennzahlen je Suchraumgröße erhoben. Die den dargestellten Effizienz-Kennzahlen zugrundeliegenden Rohdaten sind wiederum in [Anhang A](#) visualisiert. Im Gegensatz zur Verbundgröße ist das Verfahren bzgl. Laufzeit- und Kommunikationseffizienz stabil gegenüber variierenden Suchraumgrößen. Die Berechnungseffizienz sinkt im vorliegenden Fall jedoch dramatisch mit wachsenden Suchräumen, was aber unmittelbar auf die Verwendung des einfachen enumerierten Suchraummodells zurückzuführen ist. In diesem Modell erfolgt die Elementauswahl mittels einer sequenziellen Suche im als Liste organisierten Suchraum. Wie bereits in [Abschnitt 6.4](#) dargelegt, kann dieser Nachteil durch die Verwendung eines Stützvektorbasierten Modells komplett eliminiert werden. Eine entsprechende Auswertung wird später in [Abschnitt 7.6.2](#) gegeben. Weiterhin lässt sich ein Bezug zwischen Berechnungsaufwand in dem verwendeten Suchraummodell und der Reaktionszeit eines Verbundteilnehmers herstellen, siehe dazu den folgenden [Abschnitt 7.4.3](#).

7.4.3 Planungshorizont

Der Instanzparameter q definiert schließlich die Dimensionalität der Elemente, und damit die Länge des Planungshorizontes im Anwendungskontext. [Abbildung 7.16](#) zeigt das Ergebnis der Simulationsreihe mit $q \in \{5, 10, 20, 30, 40, 50, 100\}$. Es wurden wiederum die Verteilungen der oben eingeführten Kennzahlen je Suchraumgröße erhoben, die dazugehörigen Rohdaten befinden sich in [Anhang A](#). Die Daten bestätigen, dass das Verfahren bezüglich aller drei Effizienz-Kennzahlen stabil gegenüber variierenden Planungshorizonten ist. In Bezug auf die Berechnungseffizienz überrascht dieses Ergebnis auf den ersten Blick. Hierzu ist zu bemerken, dass die Berechnungseffizienz die *Anzahl* der Zielfunktionsauswertungen misst, aber den dabei entstehenden Rechenaufwand pro Zielfunktionsauswertung nicht berücksichtigt. Diese Abstraktionsebene der Kennzahl für die Berechnungseffizienz wurde hier

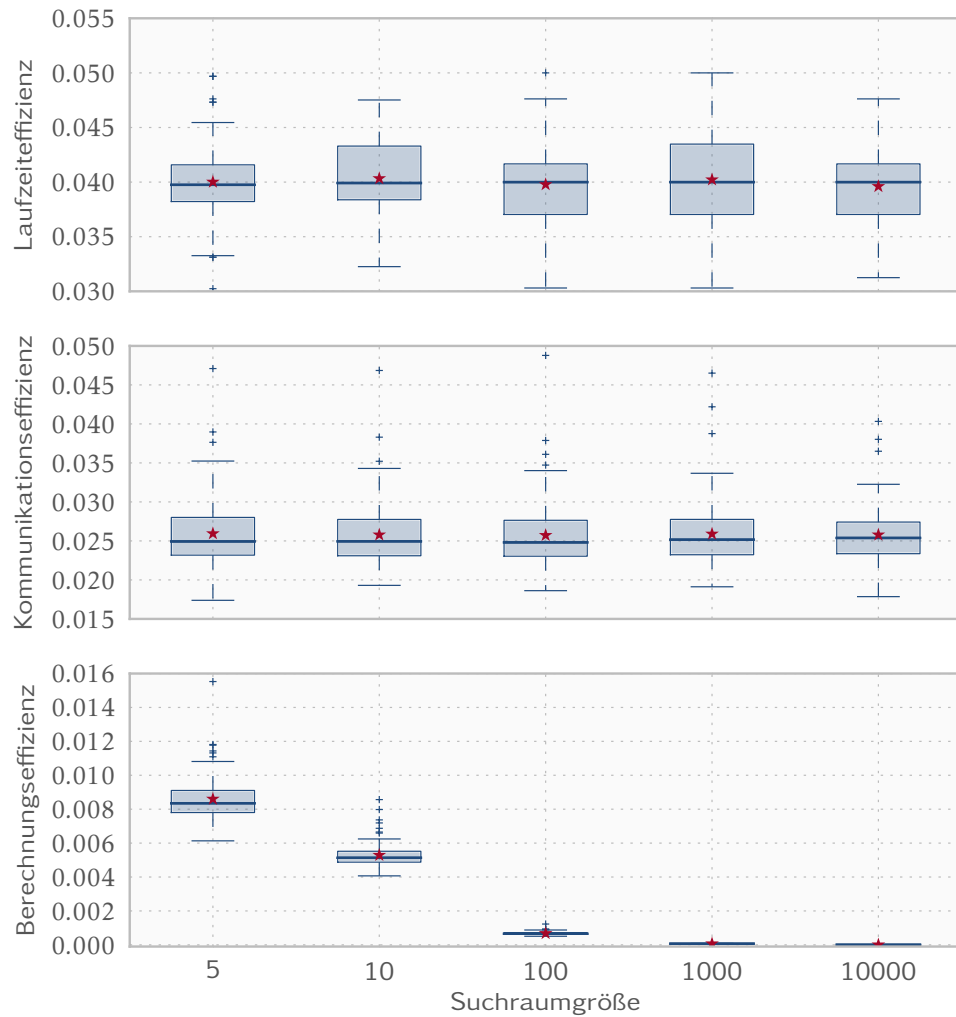


Abbildung 7.15: Laufzeiteffizienz, Kommunikationseffizienz und Berechnungseffizienz bei Simulation verschiedener Suchraumgrößen. Darstellung jeweils als Box-plot sowie Mittelwert (Sternsymbol) für 100 Ausführungen je Suchraumgröße.

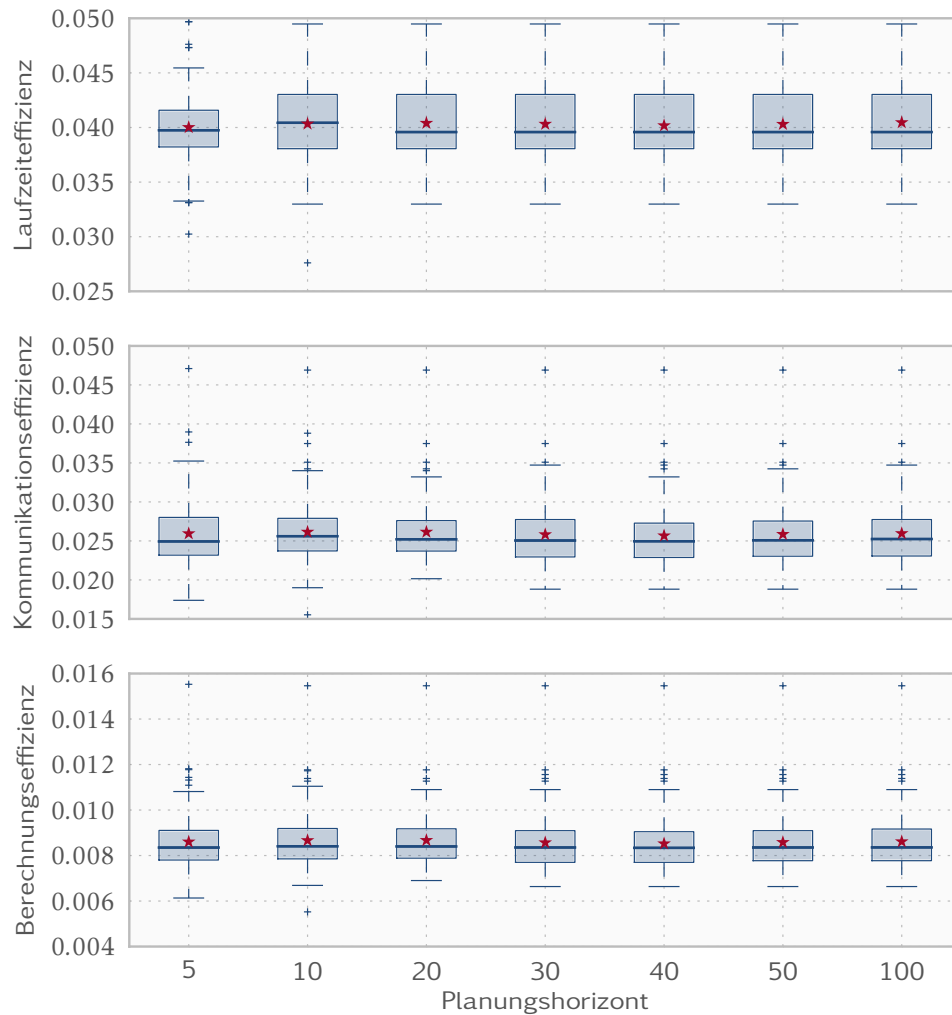


Abbildung 7.16: Laufzeiteffizienz, Kommunikationseffizienz und Berechnungseffizienz bei Simulation verschiedener Planungshorizonte. Darstellung jeweils als Box-plot sowie Mittelwert (Sternsymbol) für 100 Ausführungen je Planungshorizont.

gewählt, da sich der Rechenaufwand für die Zielfunktionsauswertungen in dem betrachteten verteilten System als Reaktionsgeschwindigkeit der Systemteilnehmer, und damit über die oben bereits evaluierte Reaktionszeit ausdrücken lässt. Der Vollständigkeit halber sei aber folgende Ergänzung mit Bezug zu dem in der Praxis verwendeten Suchraummodell gegeben. Im vorliegenden Fall beinhaltet das enumerierte Suchraummodell σ^{enum} eine einfache sequenzielle Suche, in welcher jedes Element komponentenweise ausgewertet wird, und ist somit in seinem Berechnungsaufwand linear von der Größe des Suchraumes sowie von der Dimensionalität der Elemente abhängig (siehe vorherigen Abschnitt zur Suchraumgröße). Die vorliegende Implementierung des Stützvektor-basierten Suchraummodells σ^{SVM} hingegen weist einen Rechenaufwand auf, der einerseits von der Struktur der gelernten Daten hinsichtlich der benötigten Iterationen für die Kern-Transformationen, und andererseits von der Anzahl der erforderlichen Stützvektoren zur Beschreibung des Suchraumes abhängt. Erstere liegt empirisch im ein- bis zweistelligen Bereich, während letztere wiederum linear von der Länge des Planungshorizontes abhängig ist (vgl. [HBS13]). Somit wächst der tatsächliche Rechenaufwand einer Zielfunktionsauswertung für beide verwendeten Suchraummodelle linear mit der Länge des Planungshorizontes. Es ist somit davon auszugehen, dass die Reaktionszeiten der Verbundteilnehmer je nach Suchraummodell mehr oder weniger stark mit der Länge des Planungshorizontes steigen, was aber die Robustheit des Verfahrens nicht beeinträchtigt (siehe [Abschnitt 7.3.3](#)).

7.5 REGRESSIONSANALYSE

In den vorigen Abschnitten wurden verschiedene Parameter der Einsatzplanungsheuristik COHDA hinsichtlich ihres Einflusses auf die Performanz und Effizienz des Verfahrens untersucht. Dabei konnte die jeweils resultierende Performanz hinreichend erklärt werden, zudem zeigten einige der Parameter keine signifikanten Effekte auf die Effizienz oder waren in ihrem Wertebereich beschränkt, sodass eine qualitative Auswertung der Simulationsergebnisse für diese Parameter ausreichte. Insbesondere jedoch einflussstarke sowie in ihrem Wertebereich offene Parameter sollten einer genaueren, quantitativen Betrachtung hinsichtlich des Einflusses auf Effizienzmerkmale unterzogen werden. So interessiert es beispielsweise, ob die Si-

mulationsdauer mit wachsender Reaktionszeit der Verbundteilnehmer exponentiell, linear oder nur logarithmisch steigt. Im Folgenden wird daher eine Regressionsanalyse bzgl. des Zusammenhangs zwischen Parametergröße und Effizienz-Effektgröße für diese Parameter durchgeführt. Konkret werden folgende Parameter betrachtet: Vernetzungsgrad des Overlay-Netzes, Nachrichtenverzögerung, Reaktionszeit, Verbundgröße, Suchraumgröße sowie Planungshorizont. Hierbei wird der Parameter als unabhängige und die Effektgröße als abhängige Variable deklariert. Da das Interesse hier dem prinzipiellen funktionalen Zusammenhang zwischen diesen Variablen gilt und eine exakte Bestimmung der Koeffizienten der erklärenden Funktionsgleichung nicht erforderlich ist, beschränkt sich die vorliegende Analyse auf das Verfahren der *linearen Regression*. Diese basiert auf der Methode der kleinsten Quadrate nach Carl Friedrich Gauß und ist in der Lage, den Zusammenhang zweier Variablen als Linearkombination darzustellen. Für die Schätzung eines Polynom ersten Grades bzgl. der Variablen x (unabhängig) und y (abhängig) würde das Verfahren etwa die Koeffizienten β_0, β_1 liefern, sodass das Residuum e in folgender Gleichung über der betrachteten Definitionsmenge D minimiert wird:

$$y = \beta_0 + \beta_1 \cdot x + e, \quad x \in D, \quad \beta_0, \beta_1, e \in \mathbb{R} \quad (7.12)$$

Prinzipiell sind auch Polynome höheren Grades in der linearen Regression verwendbar (die abhängige Variable kann also durchaus in der zweiten, dritten etc. Potenz vorkommen), solange der Ausdruck in den Koeffizienten $\beta_0, \dots, \beta_n$ linear ist. Im vorliegenden Fall reicht es jedoch aus, nur zwei Koeffizienten β_0, β_1 zu betrachten. Um hiermit nun verschiedene funktionale Zusammenhänge zu testen, wird dieses Verfahren mit entsprechenden Variablentransformationen zur Linearisierung verknüpft [UM11, Abs. 4.3]. Den Grundgedanken dazu erläutert [Beispiel 7.5.1](#) auf Seite 166. Die darin beschriebene Linearisierung samt linearer Regression kann nun für verschiedene Zusammenhangsmodelle durchgeführt werden. Ein Vergleich der Residuen der einzelnen Regressionen liefert dann ein Entscheidungskriterium dafür, welches der getesteten Modelle den realen Zusammenhang am besten erklärt. [Tabelle 7.2](#) zeigt eine Aufstellung der in dieser Arbeit getesteten Zusammenhangsmodelle und entsprechender Transformationen. Die Wahl der Basis für die Logarithmierung ist für die Regression unerheblich, da sie einerseits sowohl zur Linearisierung, als auch für die Rücktransformation verwendet wird und sich daher bzgl. der Regression

Beispiel 7.5.1

Es sei angenommen, dass zwischen zwei Variablen x, y ein exponentieller Zusammenhang existiere, sodass die einfache Inkrementierung von x ein exponentielles Inkrement von y bzgl. einer Konstanten c bewirkt:

$$y \propto c^x, \quad c \in \mathbb{R}$$

Dann lässt sich dies auch so ausdrücken, dass die logarithmische Inkrementierung von x zur Basis c ein einfaches Inkrement von y bewirkt:

$$\log_c(y) \propto x, \quad c \in \mathbb{R}$$

Da die linke und rechte Seite proportional zueinander sind, kann der so formulierte Zusammenhang als Linearkombination dargestellt werden:

$$\log_c(y) = \beta_0 + \beta_1 \cdot x, \quad c, \beta_0, \beta_1 \in \mathbb{R}$$

Mit $y' = \log_c(y)$, $c \in \mathbb{R}$ ergibt sich dann

$$y' = \beta_0 + \beta_1 \cdot x, \quad \beta_0, \beta_1 \in \mathbb{R}$$

Der ursprünglich nichtlineare Zusammenhang wurde somit durch eine Variablentransformation in einen linearen Zusammenhang überführt. Lässt sich dieser mittels der linearen Regression ausreichend erklären, d. h. es können Werte für die Koeffizienten β_0, β_1 gefunden werden, welche die verbleibenden Residuen ausreichend minimieren, so bestätigt dies die anfängliche Annahme über den exponentiellen Zusammenhang. Zur Überprüfung (oder Vorhersage, je nach Anwendungsfall) muss ggf. nach der Regression eine Rücktransformation durchgeführt werden, um unter Verwendung der soeben bestimmten Koeffizienten die Schätzgleichung für die Variable vor ihrer Linearisierung zu erhalten:

$$\hat{y} = c^{\beta_0 + \beta_1 \cdot x}, \quad c, \beta_0, \beta_1 \in \mathbb{R}$$

Diese Gleichung beschreibt dann den konkreten, geschätzten Zusammenhang zwischen x und y .

Zusammenhang	Transformation	Regressionsgleichung	Schätzggleichung
logarithmisch	lin-log	$y = \beta_0 + \beta_1 \cdot \ln x$	$\hat{y} = \beta_0 + \beta_1 \cdot \ln x$
potenziell	log-log	$\ln y = \beta_0 + \beta_1 \cdot \ln x$	$\hat{y} = e^{\beta_0} \cdot x^{\beta_1}$
linear	lin-lin	$y = \beta_0 + \beta_1 \cdot x$	$\hat{y} = \beta_0 + \beta_1 \cdot x$
quadratisch	sqrt-lin	$\sqrt{y} = \beta_0 + \beta_1 \cdot x$	$\hat{y} = (\beta_0 + \beta_1 \cdot x)^2$
exponentiell	log-lin	$\ln y = \beta_0 + \beta_1 \cdot x$	$\hat{y} = e^{\beta_0 + \beta_1 \cdot x}$

Tabelle 7.2: Transformationen für verschiedene Zusammenhangsmodelle bzgl. der linearen Regression.

neutral verhält, sowie andererseits weil die resultierende Schätzggleichung ggf. durch einen Basiswechsel leicht auf eine beliebige andere Basis transformiert werden kann und es sich damit lediglich um eine Frage der Darstellung handelt. Im vorliegenden Fall wird durchgehend der natürliche Logarithmus eingesetzt.

Die beschriebenen Modelle und Transformationen beschreiben in der vorliegenden Form den Zusammenhang in Form eines *Wachstums*. Antiproportionales Verhalten (die abhängige Variable y erfährt bei zunehmendem x eine *Abnahme*) lässt sich aber ebenso mit der vorgestellten Methode behandeln, indem zuvor eine Invertierung des Wertebereiches der abhängigen Variable durchgeführt wird, welche dann später in der Rücktransformation wiederum umgekehrt wird. Da sich an der grundlegenden Vorgehensweise hierdurch nichts ändert, ist diese zusätzliche Transformation in [Tabelle 7.2](#) nicht als eigenständiger Fall aufgeführt.

Die folgenden Abschnitte dokumentieren die Ergebnisse entsprechender Regressionstests für die Parameter Vernetzungsgrad des Overlay-Netzes, Nachrichtenverzögerung, Reaktionszeit, Verbundgröße, Suchraumgröße sowie Planungshorizont.

7.5.1 Topologie

[Abbildung 7.17](#) zeigt die Ergebnisse der Regressionstests für den *Small-World* Dichteparameter ϕ repräsentativ als Maß für die verwendete Overlay-Topologie. Die Darstellung visualisiert links jeweils für die drei Effizienzparameter Simulationsdauer, Kommunikationsaufwand sowie Berechnungsaufwand die geschätzten Zusammenhänge für alle der in [Tabelle 7.2](#) vorgestellten Modelle nach jeweils erfolgter Regression. Als Datenpunkte, für welche die Schätzungen hier durchgeführt wurden,

kamen die Mediane der entsprechenden Simulationsergebnisse aus [Abschnitt 7.3.1](#) zum Einsatz. Diese sind durch schwarze Punkte in den Diagrammen gekennzeichnet. Auf der rechten Seite ist dazu die Güte der jeweiligen Modelle anhand des Bestimmtheitsmaßes R^2 dargestellt, wobei R den Korrelationskoeffizienten nach Pearson und Bravais beschreibt. Das Bestimmtheitsmaß hat den Wertebereich $R^2 \in [0, 1]$ und beschreibt den Anteil der Variation der abhängigen Variable, welcher durch das verwendete Regressionsmodell erklärt werden konnte. Je höher dieser Wert, desto genauer stimmt somit der während der Regression zugrundegelegte Zusammenhang mit dem realen Zusammenhang überein. Im vorliegenden Fall kommen für alle drei Effizienzmerkmale nur das logarithmische oder das potenzielle Modell infrage. Hierbei ist festzustellen, dass für das potenzielle Modell der von der Regression geschätzte Wert des Koeffizienten β_1 jeweils im Wertebereich $[0, 1]$ liegt (in der Abbildung nicht explizit aufgeführt, aber an dem Verlauf der „log-log“ Kurve zu erkennen). Damit handelt es sich effektiv um ein Wachstum bzw. eine Abnahme in der Größenordnung einer Wurzelfunktion. Für beide jeweils zutreffenden Modelle, logarithmisch und potenziell, gilt also: Die Änderungsrate der Zu-/Abnahme *sinkt* mit zunehmender Größe der unabhängigen Variable ϕ . Mit zunehmendem Vernetzungsgrad wird somit sein Einfluss auf die Effizienz des Verfahrens in allen drei Merkmalen nach und nach geringer.

7.5.2 Nachrichtenverzögerung

[Abbildung 7.18](#) zeigt die Ergebnisse der Regressionstests für den Parameter *Nachrichtenverzögerung*. Die Darstellung folgt der Beschreibung des vorigen Abschnittes. Das Effizienzmerkmal der Simulationsdauer weist hier offenbar einen positiven linearen Zusammenhang zur unabhängigen Variable auf – sie wächst somit proportional mit der insgesamt auftretenden Nachrichtenverzögerung. Allerdings ist auch der Wert des Bestimmtheitsmaßes für das Modell „sqrt-lin“ vergleichsweise groß, sodass ein leichtes quadratisches Wachstum nicht vollständig ausgeschlossen werden kann. Für die verbleibenden beiden Effizienzparameter kann analog zum Vernetzungsgrad im vorigen Abschnitt ein abnehmendes Wachstum verzeichnet werden.

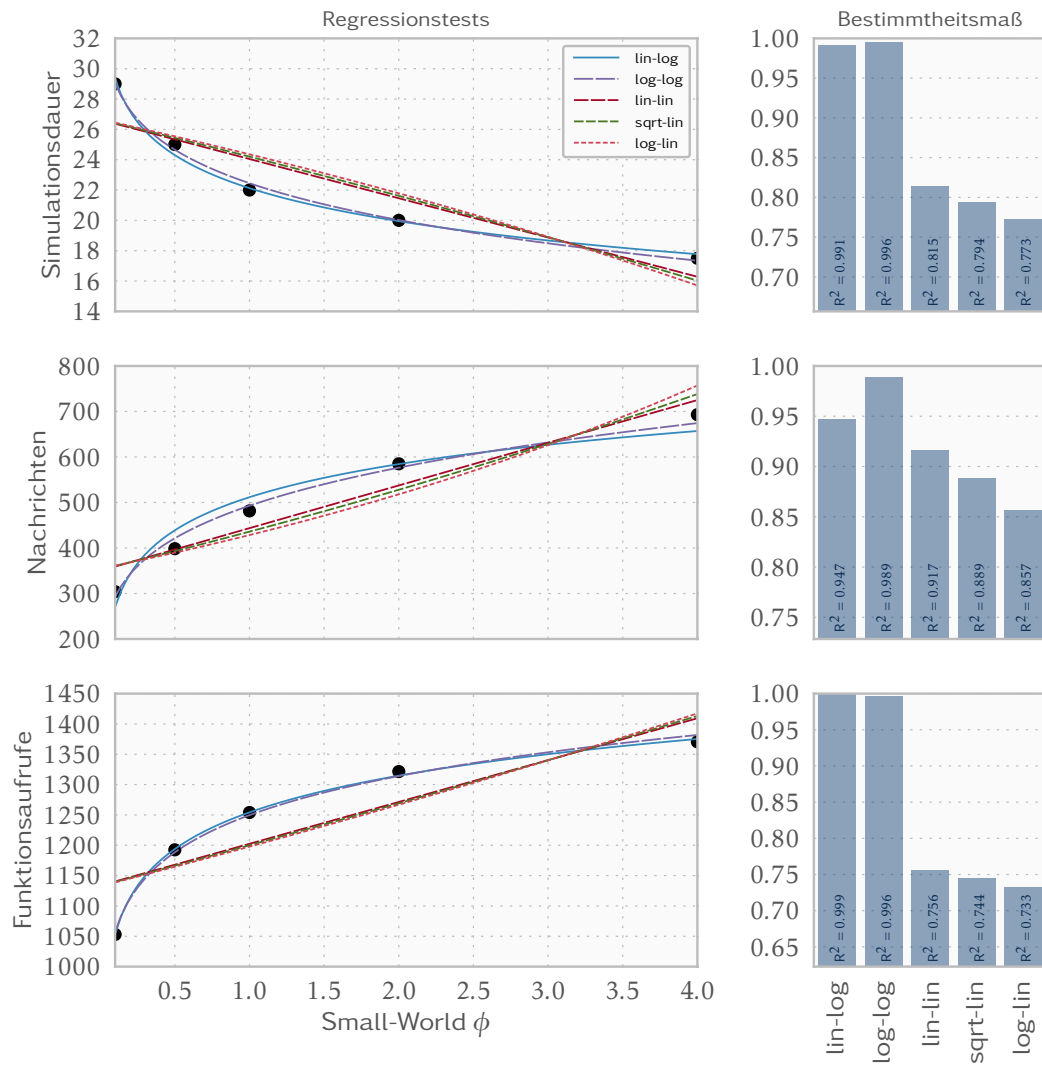


Abbildung 7.17: Ergebnisse der Regressionstests für den Zusammenhang zwischen Vernetzungsgrad und Effizienz des Verfahrens.

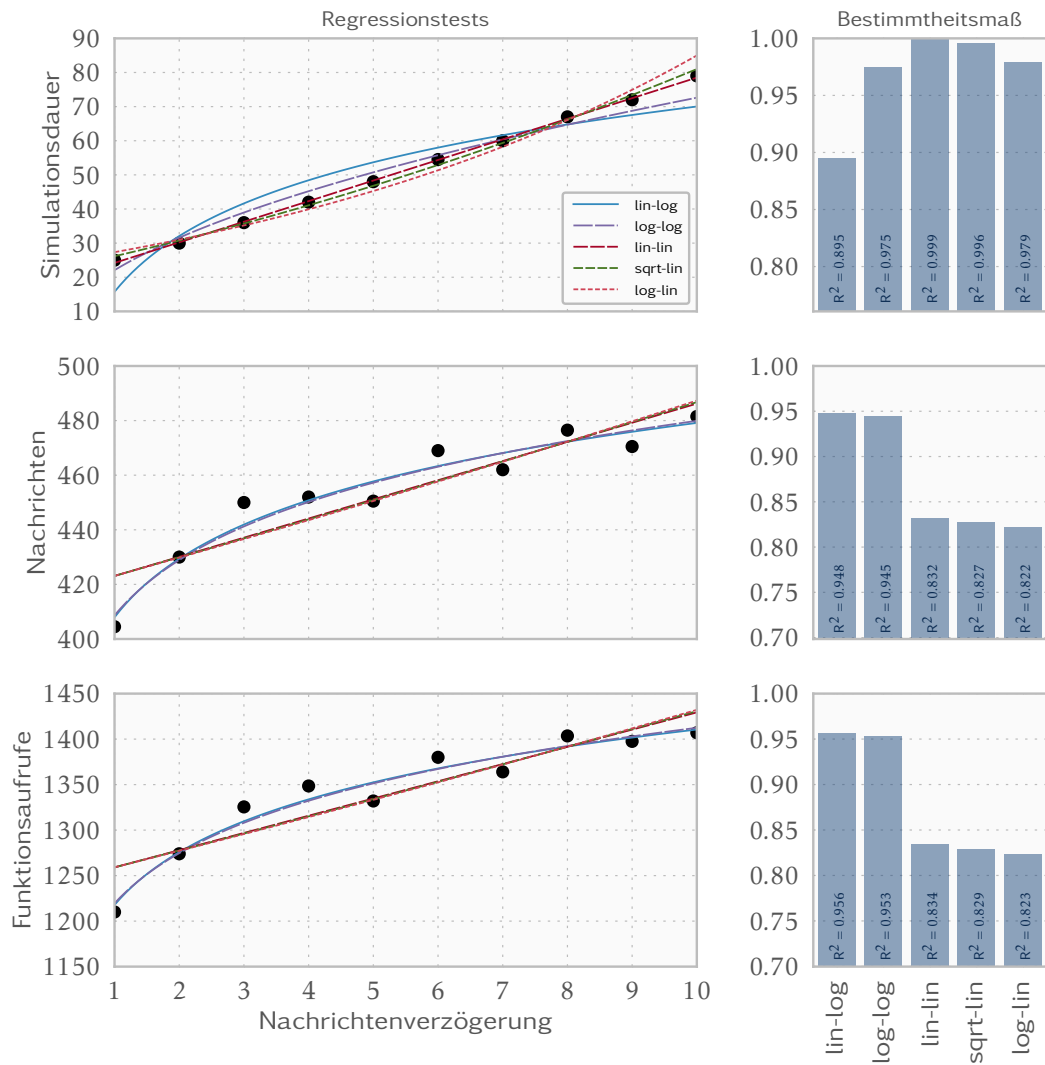


Abbildung 7.18: Ergebnisse der Regressionstests für den Zusammenhang zwischen Nachrichtenverzögerung und Effizienz des Verfahrens.

7.5.3 Reaktionszeit

[Abbildung 7.19](#) zeigt die Ergebnisse der Regressionstests für den Parameter *Reaktionszeit*. Die Werte des Bestimmtheitsmaßes für die Modelle des linearen und quadratischen Wachstums bei der Simulationsdauer liegen sehr nah beieinander, sodass analog zum vorigen Abschnitt nicht eindeutig entschieden werden kann, um welches Zusammenhangsmodell es sich hier handelt. Falls es sich um ein quadratisches Wachstum handeln sollte, ist dieses jedoch nur sehr schwach ausgeprägt. Für die verbleibenden Effizienzparameter gilt jeweils wieder, dass es sich um einen mit größer werdender Reaktionszeit abnehmenden Einfluss handelt.

7.5.4 Verbundgröße

Der Einfluss der Verbundgröße auf die Performanz und Effizienz des Verfahrens wurde aufgrund der schwierigen Vergleichbarkeit der Problem instanzen anhand von normierten Effizienzkennzahlen evaluiert. Hiermit ist jedoch eine Regressionsanalyse nicht sinnvoll durchführbar, da diese normierten Werte naturgemäß jeweils im Grenzwertverhalten gegen den Wert 0 gehen (siehe die Erläuterung in [Abschnitt 7.4.1](#)). Für den Einsatz des Verfahrens in der Praxis ist jedoch der tatsächliche Zeit-, Kommunikations- und Berechnungsaufwand von Interesse. Daher wurde hier eine Regressionsanalyse auf Basis der nicht normierten Rohdaten (vgl. [Abbildung A.5](#) in [Anhang A](#)) durchgeführt, deren Ergebnisse jedoch aufgrund der Diskrepanz in den verwendeten Problem instanzen unter Vorbehalt zu sehen sind. In [Abbildung 7.20](#) ist zu erkennen, dass die Simulationsdauer logarithmisch zur wachsenden Verbundgröße zunimmt – der Einfluss der Populationsgröße auf die Anzahl der benötigten Schritte bis zur Terminierung wird somit für große Zahlen umso geringer. Für den Kommunikations- und Berechnungsaufwand hingegen muss jedoch ein leicht überproportionaler Einfluss der Verbundgröße festgestellt werden. Konkret wurde jeweils ein potenzielles Wachstum $\hat{y} = e^{\beta_0} \cdot x^{\beta_1}$ mit $\beta_1 \approx 1.4$ als Schätzwert ermittelt, und liegt damit zwischen linearem und quadratischem Wachstum. Zur Absicherung dieser Ergebnisse wird in [Abschnitt 7.6.2](#) eine weitere Regressionsanalyse bzgl. der Verbundgröße für ein Szenario aus dem Anwendungsfall der Wirkleistungsbereitstellung durchgeführt.

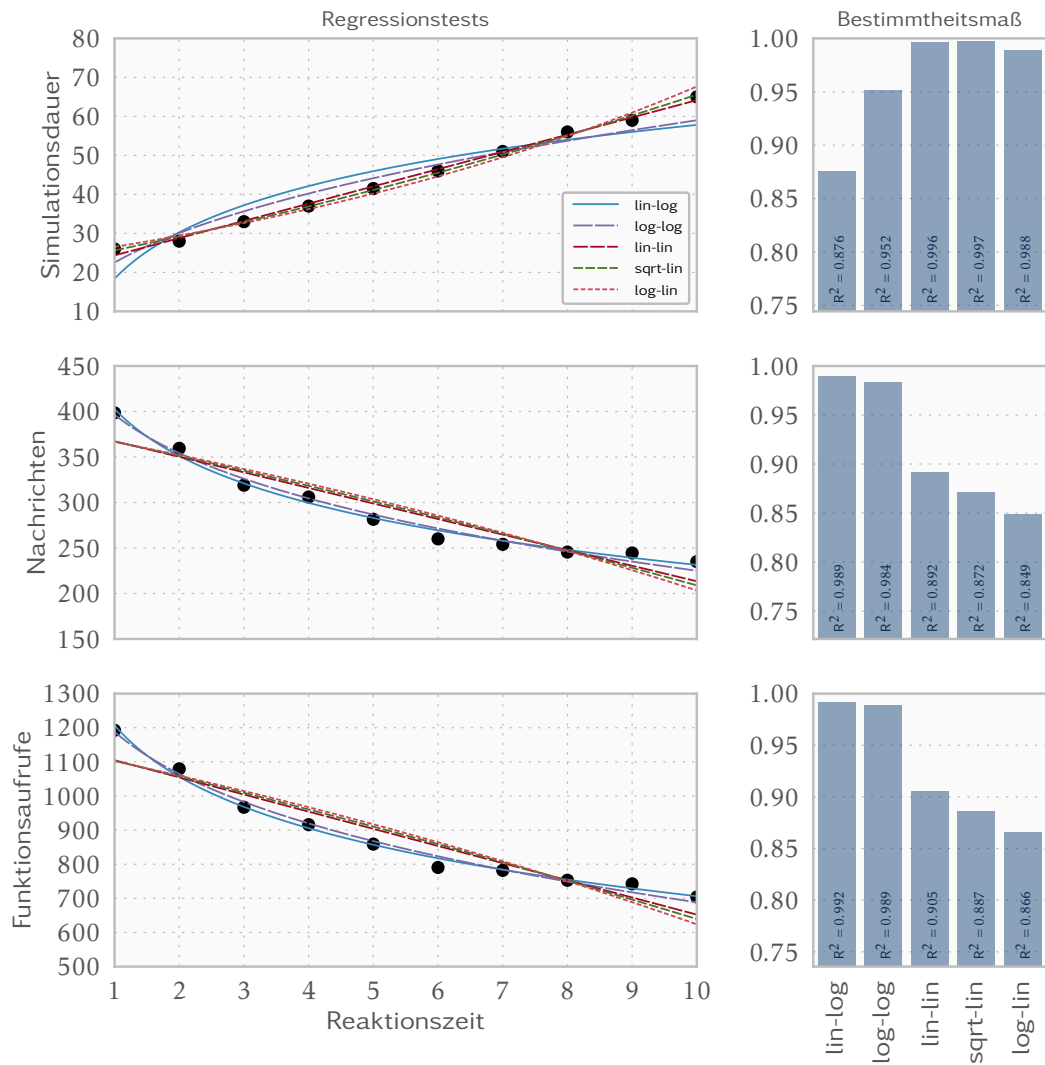


Abbildung 7.19: Ergebnisse der Regressionstests für den Zusammenhang zwischen Reaktionszeit und Effizienz des Verfahrens.

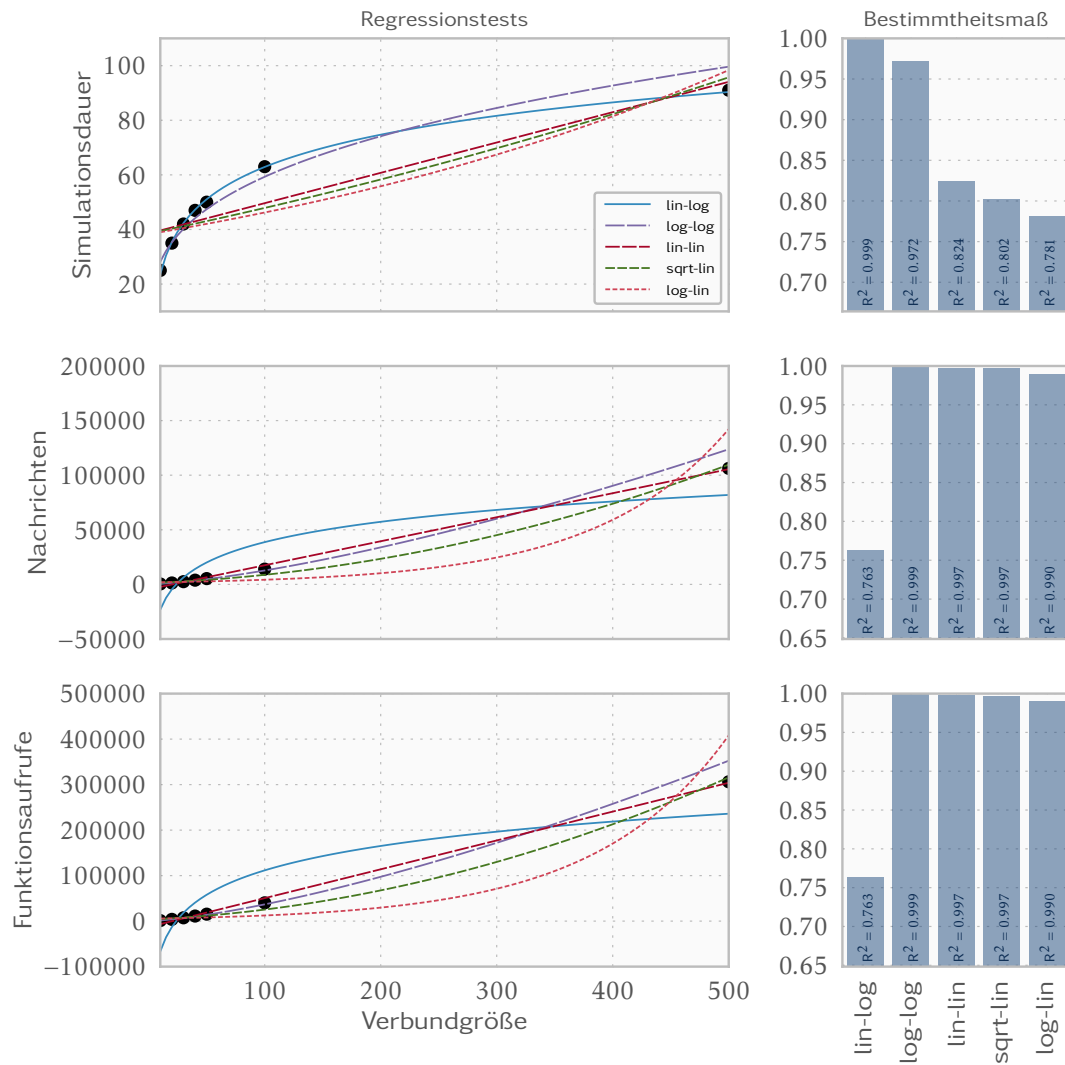


Abbildung 7.20: Ergebnisse der Regressionstests für den Zusammenhang zwischen Verbundgröße und Effizienz des Verfahrens.

7.5.5 Suchraumgröße

Analog zur Verbundgröße wurden die Regressionstests für die Suchraumgröße auf den nicht normierten Rohdaten durchgeführt, [Abbildung 7.21](#) zeigt die Ergebnisse. Die Bestimmtheitsmaße für die Regressionsanalyse der Simulationsdauer konnten hier nicht erhoben werden, da für alle Konfigurationen die gleiche Anzahl an Schritten benötigt wurde, und somit kein Wachstum (bzw. keine Abnahme) der abhängigen Variable messbar ist. Ein funktionaler Zusammenhang ist somit nicht vorhanden. Für den Kommunikationsaufwand ist ebenfalls keine sinnvolle Aussage möglich, da die Anzahl der versendeten Nachrichten keiner wohlgeordneten Zu- oder Abnahme folgen und nur sehr gering streuen. Auch hier scheint also kein funktionaler Zusammenhang zu bestehen. Für die Anzahl der Funktionsaufrufe wurde ein linearer Zusammenhang erkannt, welcher bereits in der qualitativen Betrachtung in [Abschnitt 7.4.2](#) argumentativ anhand der Arbeitsweise des hier verwendeten Suchraummodells hergeleitet wurde. Das hohe Bestimmtheitsmaß des Modells „log-log“ kommt hier zustande, weil der in der Potenz stehende Parameter (siehe [Tabelle 7.2](#)) mit $\beta_1 \approx 0.9$ geschätzt wurde. Dies entspricht somit nahezu einem linearen Wachstum und kann daher ignoriert werden.

7.5.6 Planungshorizont

Wie die Auswertungen in [Abschnitt 7.4.3](#) sowie [Anhang A](#) bereits zeigten, hat die Länge des Planungshorizontes keinen Einfluss auf die Effizienz des Verfahrens. Eine Regressionsanalyse ist daher aufgrund des nicht vorhandenen funktionalen Einflusses nicht sinnvoll durchführbar.

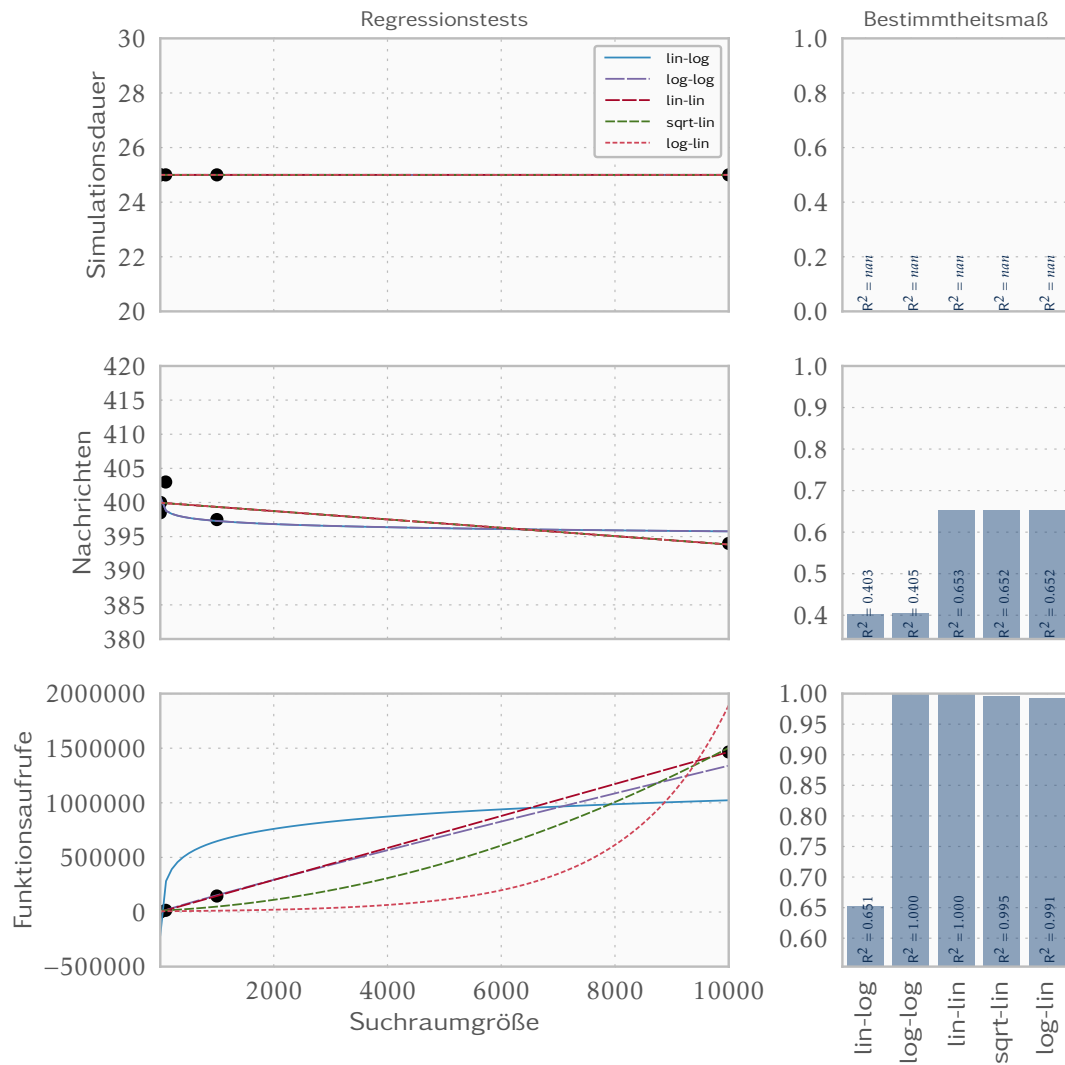


Abbildung 7.21: Ergebnisse der Regressionstests für den Zusammenhang zwischen Suchraumgröße und Effizienz des Verfahrens.

7.6 DOMÄNENSPEZIFISCHE SZENARIEN

Der in dieser Arbeit fokussierte Anwendungsfall ist die prädiktive Einsatzplanung dezentraler Akteure im Smart Grid. Die hierzu entwickelte Heuristik COHDA wurde im vorigen Abschnitt anhand synthetischer Probleminstanzen detailliert hinsichtlich Robustheit und Skalierbarkeit untersucht. Im Folgenden soll nun die Einsatzfähigkeit des Verfahrens bzgl. domänenspezifischer Szenarien evaluiert werden. Dazu wird zunächst in [Abschnitt 7.6.1](#) ein einfacher Verbund aus 30 Mikro-Blockheizkraftwerken betrachtet, um das prinzipielle Laufzeitverhalten zu untersuchen. Dieser Verbund wird anschließend in [Abschnitt 7.6.2](#) verwendet, um eine weitere Regressionsanalyse hinsichtlich des Einflusses der Verbundgröße auf die Effizienz des Verfahrens durchzuführen, und kommt erneut in [Abschnitt 7.6.3](#) für den Vergleich verschiedener Suchraummodelle zum Einsatz. In [Abschnitt 7.6.4](#) werden dann Szenarien auf der Grundlage des Auktionshandels an der Strombörse EPEX SPOT betrachtet. Abschließend zeigt [Abschnitt 7.6.5](#) die Einsatzfähigkeit des Verfahrens für nicht auf Wirkleistungsprodukten basierende Zielfunktionen anhand des Anwendungsfalls der Reduktion der Spreizung eines Tageslastprofils.

Die Berechnung des normierten Fehlers e erfolgt dabei – sofern in den einzelnen Abschnitten nicht anders angegeben – analog zu [Abschnitt 7.4](#) mittels einer Abschätzung. Dabei wird für $e = 0$ die Existenz eines realisierbaren Verbundfahrplanes angenommen, welcher das jeweils gegebene Optimierungsziel exakt erfüllt. Zur Abschätzung der schlechtesten Lösung für $e = 1$ wird wiederum die in [Algorithmus 4](#) beschriebene Heuristik eingesetzt. Bei der Interpretation der im Folgenden präsentierten Fehlerwerte ist daher zu berücksichtigen, dass das durch diese Heuristik produzierte Normierungsintervall aufgrund einer möglichen Unterschätzung der oberen Schranke entsprechend schmal ausfallen kann, was die produzierten Fehlerwerte potenziell groß erscheinen lässt. Der Fehler ist somit nur bedingt als Lagemaß für die Güte der produzierten Verbundfahrpläne geeignet, sondern sollte vielmehr wie in den vorigen Abschnitten als Vergleichsmaß für verschiedene Simulationen des gleichen Szenarios betrachtet werden.

7.6.1 Mikro-Blockheizkraftwerke

Als Grundlage für den in diesem und den folgenden beiden Abschnitten betrachteten Verbund aus Mikro-Blockheizkraftwerken dient ein Simulationsmodell eines EcoPower 4.7 inkl. 800 l Warmwasserspeicher des Herstellers Vaillant². Das Modell wird im wärmegeführten Modus betrieben, die Datengrundlage hierzu liefert der gemäß [JV01] abgeleitete Wärmebedarf eines Vierfamilienhauses. Um einen sehr großen Handlungsspielraum zu simulieren, wird es dem Modell erlaubt, seine Motordrehzahl in 5000 Stufen zu modulieren. Als Nebenbedingung beinhaltet das Modell, dass sich die Drehzahl von einem Zeitschritt zum nächsten um nicht mehr als 100 Modulationsstufen verändern darf. Die produzierte elektrische Leistung des Modells bewegt sich im eingeschalteten Zustand je nach Modulationsstufe im Bereich $P_{el} \in [1.3 \text{ kW}, 4.7 \text{ kW}]$. Zusätzlich kann die Anlage ausgeschaltet werden ($P_{el} = 0 \text{ kW}$), hierbei muss dann jedoch eine Mindestruhezeit von 120 Minuten eingehalten werden.

Aus dem derart konfigurierten Simulationsmodell wurden für das vorliegende Szenario $m = 30$ konkrete Instanzen des Mikro-Blockheizkraftwerkes erzeugt. Diese unterschieden sich einerseits in der initialen Temperatur des Pufferspeichers, welche gleichverteilt über dem erlaubten Intervall $[65^\circ\text{C}, 75^\circ\text{C}]$ gebildet wurde. Zudem wurde der Wärmebedarf für jede Instanz nach [JV01] leicht verrauscht. Für die Instanzen wurden dann jeweils $n_i = 200$ realisierbare Fahrpläne für 96 Planungsintervalle $T = \{t_0, \dots, t_{95}\}$ mit $\Delta t = 15$ Minuten zufällig erzeugt, und damit die jeweiligen Suchraummodelle σ_i^{enum} gebildet. Als zu realisierendes Wirkleistungsprodukt kam ein zufällig erzeugter Verbundfahrplan aus den gebildeten Suchräumen zum Einsatz. Für dieses Szenario ist die optimale Lösung also bekannt, sodass leicht evaluiert werden kann, ob die Heuristik dieses Optimum erreicht bzw. wie weit davon entfernt sie terminiert.

Abbildung 7.22 zeigt exemplarisch den Verlauf einer einzelnen Durchführung dieses Szenarios. Wie bereits in Abschnitt 7.2 visualisiert, ist auch hier der monoton fallende Verlauf des Fehlerwertes über den Fortschritt der Heuristik zu sehen. Dies veranschaulicht nochmals die in Lemma 1 gezeigte *anytime* Eigenschaft des Verfahrens, welche es damit prinzipiell ermöglicht, dem Wirkleistungsprodukt einen benutzerdefinierten Schwellenwert in Form eines Zielwertes für die Güte eines Lö-

²<http://www.vaillant.de>, letzter Zugriff 6. Januar 2014

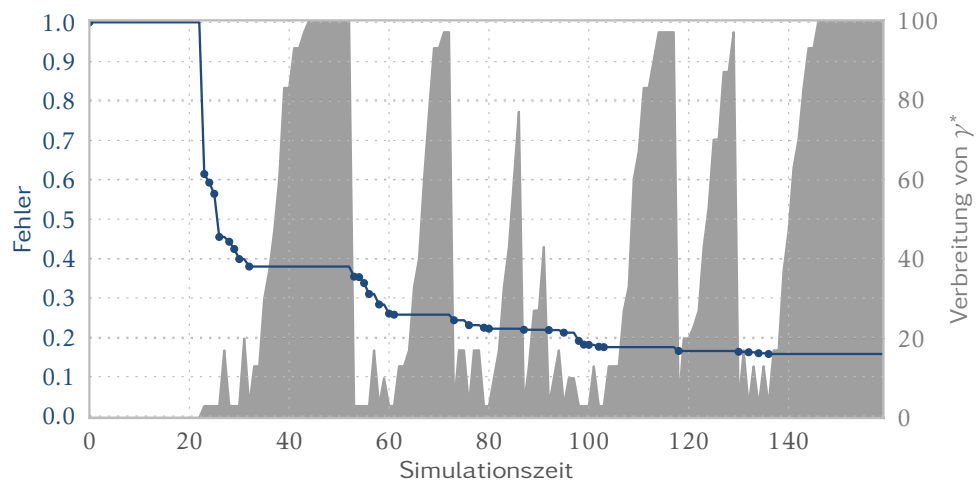


Abbildung 7.22: Laufzeitverhalten einer Simulation für einen Verbund aus 30 Mikro-Blockheizkraftwerken und zufälligem, aber realisierbarem Wirkleistungsprodukt.

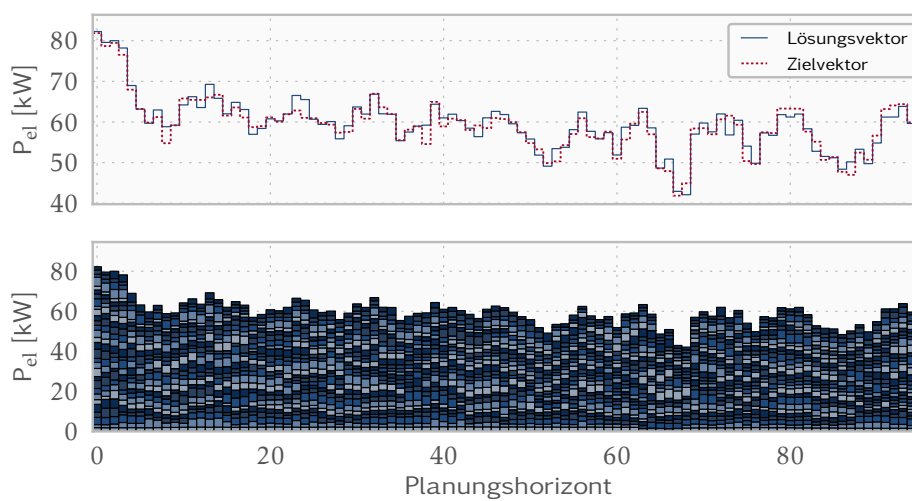


Abbildung 7.23: Resultat einer Simulation für einen Verbund aus 30 Mikro-Blockheizkraftwerken und zufälligem, aber realisierbarem Wirkleistungsprodukt.

sungskandidaten mitzugeben. Wie bei Heuristiken oft üblich [Tal09], kann ein solcher Schwellenwert als Abbruchkriterium eingesetzt werden, um das Verfahren bei einer gewünschten Güte vorzeitig terminieren zu lassen. Im vorliegenden Fall wurde ein solches Abbruchkriterium nicht implementiert – die Heuristik terminiert also erst, wenn keine Verbesserung der Lösung mehr erreicht werden kann. Dies ist im dargestellten Verlauf nach 162 Zeitschritten der Fall. Der finale Lösungskandidat γ^* wurde in Zeitschritt 136 gebildet und besitzt einen verbleibenden Fehler von $e = 0.158$ gemäß Gleichung 7.7. Wie bereits in der Einleitung zu diesem Abschnitt argumentiert, scheint dieser Fehlerwert im Vergleich zu den exakt bestimmten Fehlerwerten in Abschnitt 7.1 relativ groß zu sein. Daher kann zur weiteren Interpretation ergänzend der resultierende Verbundfahrplan betrachtet werden, welcher in Abbildung 7.23 dargestellt ist. Der obere Teil zeigt das vorgegebene Wirkleistungsprodukt im Vergleich zum Verbundfahrplan. Die sichtbare Differenz beträgt kumuliert $\|\zeta - \Theta\|_1 = 124.033 \text{ kW}$, und entspricht damit einer Abweichung von 2.18 % in Relation zur Gesamtleistung des Wirkleistungsproduktes. Die Heuristik war in diesem Fall somit nicht in der Lage, das vorgegebene Wirkleistungsprodukt exakt zu rekonstruieren. Stattdessen wurde eine Kombination von Anlagenfahrplänen gefunden, die das Wirkleistungsprodukt zu 97.82 % in Bezug zur Gesamtleistung erfüllt. Diese Kombination von Fahrplänen ist im unteren Teil der Abbildung visualisiert.

Abbildung 7.24 zeigt ergänzend das Ergebnis einer Reihe von 100 Durchführungen dieses Szenarios. Anhand der Zielabweichung in der linken Grafik ist zu sehen, dass sich die auftretenden Abweichungen gleichmäßig über den Planungshorizont im Bereich von wenigen kW bewegen, dabei aber eher in das Negative reichen, sodass eine tendenzielle Unterdeckung der Wirkleistungsprodukte zu erwarten ist. Der Median des insgesamt verbleibenden Fehlers liegt bei $\tilde{e} = 0.18$, $\text{IQR} = 0.028$. Dies entspricht einer kumulierten Differenz von 132.866 kW, und damit einer Abweichung von 2.34 % in Relation zum Median der Gesamtleistungen aller Wirkleistungsprodukte. Insgesamt konnte die Heuristik in diesem Szenario die gegebenen Wirkleistungsprodukte somit in 50 % der Fälle zu mindestens 97.66 % erfüllen.

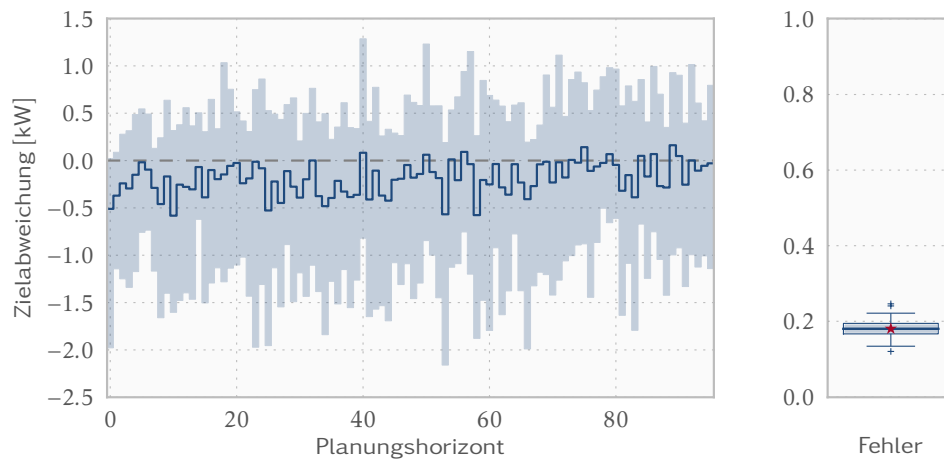


Abbildung 7.24: Resultat von 100 Simulationen eines Verbundes aus 30 Mikro-Blockheizkraftwerken und zufälligem, aber realisierbarem Wirkleistungsprodukt. Zielabweichung (links) dargestellt als Median und oberes sowie unteres Quartil. Verbleibender Fehler für die Gesamtlösung (rechts) dargestellt als Boxplot und Mittelwert (Sternsymbol).

7.6.2 Einfluss der Verbundgröße

Ergänzend zu [Abschnitt 7.5.4](#) soll nun eine Regressionsanalyse mit Verbünden aus dem fokussierten Anwendungskontext der Arbeit durchgeführt werden. Hierzu wird das im vorigen Abschnitt beschriebene Simulationsmodell eines Mikro-Blockheizkraftwerkes verwendet. Es wurden Verbundgrößen $m \in \{10, 20, 30, 40, 50, 100\}$ simuliert. Im Unterschied zu den in den bisherigen Regressionstests verwendeten synthetischen Problem instanzen ist für diese Verbünde unabhängig von der Verbundgröße die optimale Lösung des Optimierungsproblems existent und bekannt, da das Wirkleistungsprodukt mittels einer zufällig ausgewählten Kombination von Einzelfahrplänen der Anlagen gebildet wurde. Somit ist die Vergleichbarkeit der Simulationen über verschiedene Verbundgrößen hinweg gewährleistet. [Abbildung 7.25](#) zeigt die Ergebnisse der Regressionsanalyse dieses Szenarios. Die dargestellten Ergebnisse bestätigen die bereits in [Abschnitt 7.5.4](#) dokumentierten Funde: Für die Simulationsdauer wurde ein logarithmischer Zusammenhang mit der Verbundgröße identifiziert, während für Nachrichtenaufkommen sowie Berechnungsaufwand

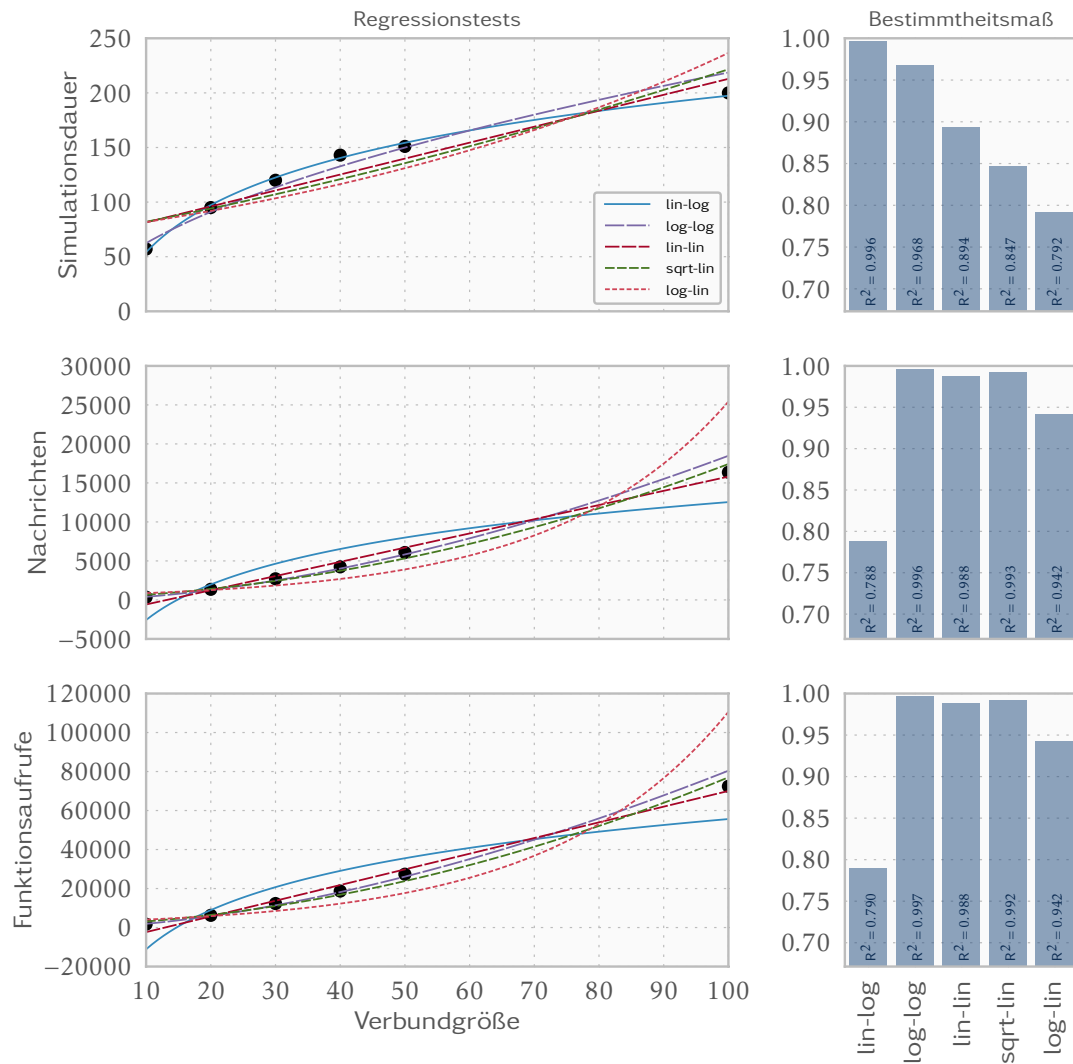


Abbildung 7.25: Ergebnisse der Regressionstests für den Zusammenhang zwischen Verbundgröße und Effizienz des Verfahrens anhand von Verbünden bestehend aus Mikro-Blockheizkraftwerken.

jeweils ein leicht überproportionales Wachstum erkennbar ist. Im vorliegenden Fall wurde für die entsprechenden Schätzgleichungen $\hat{y} = e^{\beta_0} \cdot x^{\beta_1}$ bzgl. Nachrichtenaufkommen sowie Verbundgröße jeweils $\beta_1 \approx 1.6$ ermittelt. Dieser Wert ist etwas größer im Vergleich zur Auswertung der synthetischen Problem instanzen (dort $\beta_1 \approx 1.4$), der Zusammenhang liegt jedoch weiterhin zwischen linear und quadratisch.

7.6.3 Suchraummodelle

Die Modellierung der Problemstellung in [Abschnitt 3.1](#) erlaubt es, verschiedene Realisierungen von Suchraummodellen in der Heuristik zu verwenden. Dazu wurden in [Abschnitt 6.4](#) exemplarisch ein Suchraummodell σ^{enum} auf Basis der sequenziellen Suche, und ein Suchraummodell σ^{SVM} auf Basis von maschinellen Lernverfahren vorgestellt. Letzteres bietet den Vorteil, bei geringerem Speicher- und Rechenaufwand einen deutlich größeren Handlungsspielraum einer Anlage abzudecken, und damit die Flexibilität der Anlage effizienter abzubilden [[HBS13](#)]. Um die Auswirkungen dieser Eigenschaften auf die Heuristik COHDA zu untersuchen, wurden sowohl Simulationen unter Verwendung von σ^{enum} mit einer Suchraumgröße $\forall a_i \in A : n_i = 200$, als auch Simulationen unter Verwendung von σ^{SVM} mit einer Lernbasis von $\forall a_i \in A : n_i = 200$ Elementen durchgeführt, und dann einander gegenübergestellt. Da sich die synthetischen Problem instanzen aus dem vorigen Teil der Evaluation aufgrund ihrer geringen Größe sowie der inhärenten linearen Abhängigkeiten der konstruierten Elemente nicht zur Verwendung mit σ^{SVM} eignen, liegt diesen Simulationen analog zu [Abschnitt 7.6.1](#) ein Verbund aus $m = 30$ simulierten Mikro-Blockheizkraftwerken zugrunde. Im Vergleich zu obigem Szenario wurde das Wirkleistungsprodukt jedoch nicht aus vorhandenen Fahrplänen und damit explizit als realisierbar erzeugt, sondern stattdessen zufällig in einer Größenordnung generiert, die der mittleren realisierbaren Gesamtleistung des Verbundes entsprach. Dieses Vorgehen wurde hier gewählt, da das Suchraummodell σ^{enum} ansonsten ggf. einen Vorteil durch die garantierte Existenz einer perfekten Lösung gehabt hätte, welche bei dem auf Schätzungen beruhenden σ^{SVM} nicht notwendigerweise vorhanden wäre. Für beide Konfigurationen mit dem jeweiligen Suchraummodell wurden jeweils wieder 100 Simulationen durchgeführt, die Ergebnisse sind in [Abbildung 7.26](#) visualisiert. Der Vergleich der beiden Suchraummodelle σ^{enum} und σ^{SVM} zeigt, dass unter Verwendung des SVM-Modells zumeist bessere Ergeb-

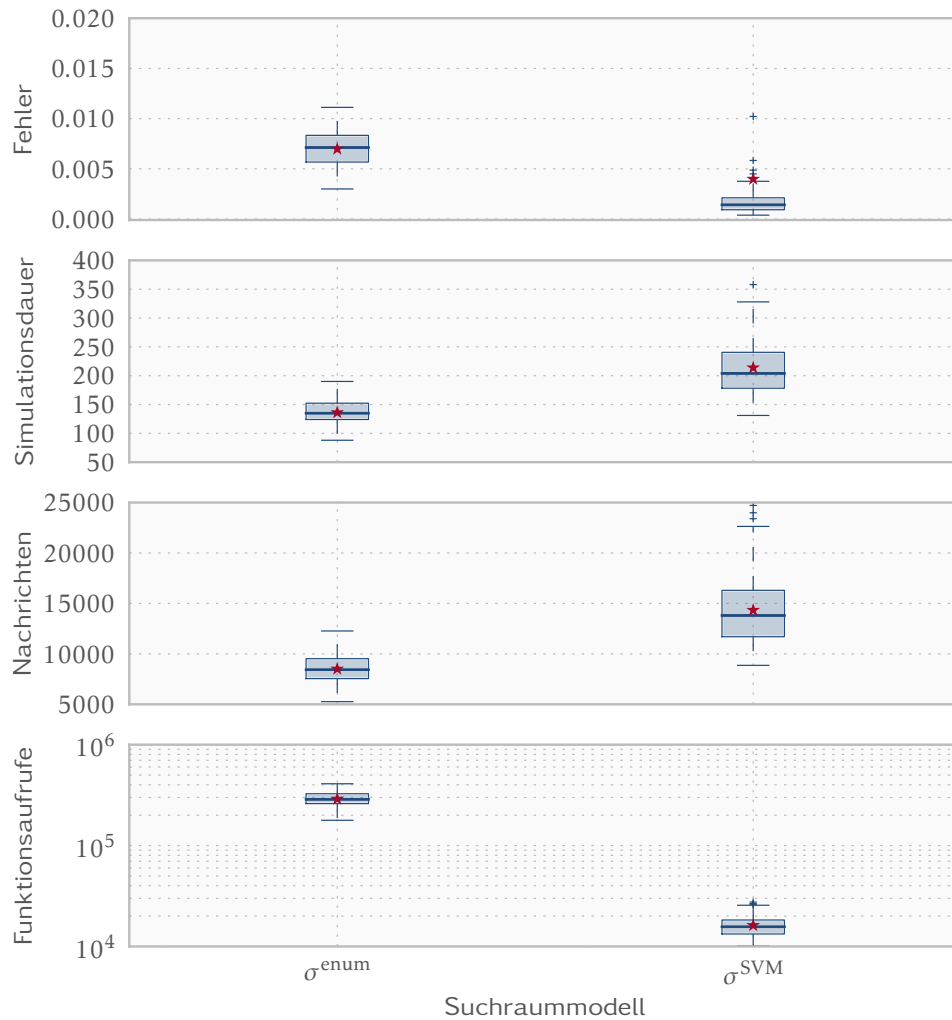


Abbildung 7.26: Verbleibender Fehler, Simulationsdauer, Kommunikationsaufwand und Rechenaufwand bei Verwendung verschiedener Suchraummodelle. Darstellung jeweils als Boxplot sowie Mittelwert (Sternsymbol) für 100 Ausführungen je Konfiguration.

nisse hinsichtlich des verbleibenden Fehlers erreicht werden können. Da es sich bei der im Suchraummodell verwendeten Dekoder-Suchfunktion jedoch um eine auf Schätzungen basierende Näherungsbestimmung von realisierbaren Fahrplänen handelt, können hierbei Schätzfehler auftreten. Im vorliegenden Fall existiert in den 100 durchgeführten Simulationen der SVM-Konfiguration beispielsweise eine einzelne Simulation mit einem vergleichsweise großen Fehlerwert $e = 0.226$ (in der Abbildung nicht zu sehen, aber erkennbar an der Position des Mittelwertes relativ zum Median). Diese Unsicherheit muss somit bei Verwendung von σ^{SVM} bedacht werden. Die Simulationsdauer sowie das Nachrichtenaufkommen steigen zudem im Vergleich zum einfachen enumerierten Suchraummodell σ^{enum} . Dies ist auf die deutlich höhere Granularität des Suchraums zurückzuführen, wodurch sich kleinere Annäherungsschritte im Verlauf der Konsensbildung ergeben und somit ein größerer Koordinationsaufwand besteht. Hervorzuheben ist jedoch die Anzahl der benötigten Zielfunktionsaufrufe, welche bei Verwendung von σ^{SVM} deutlich reduziert ist.

Zur genaueren Analyse wurde eine zweite Auswertung vorgenommen, in welcher die Simulationen bei einem erreichten verbleibenden Fehler von $e = 0.01$ gestoppt wurden, um auf diese Weise von der Konvergenzgeschwindigkeit unabhängige Aussagen treffen zu können. Das Ergebnis ist in Form der Effizienzkennzahlen aus [Abschnitt 7.4](#) in [Abbildung 7.27](#) dargestellt. Hier wird deutlich, dass die Effizienz des Suchraummodells σ^{SVM} nach Normierung auf die erreichte Lösungsgüte im Mittel höher ist, aber eine größere Streuung aufweist. Bei einer Umsetzung des Einsatzplanungsverfahrens unter Verwendung dieses Suchraummodells sollte dies berücksichtigt werden. Zudem ist ein zusätzliches Abbruchkriterium für die Heuristik COHDA in Erwägung zu ziehen, welches das Verfahren bei nachlassender Konvergenzgeschwindigkeit vorzeitig terminiert (dies kann analog zur Terminierungserkennung durch den Verbundsprecher erfolgen, siehe [Abschnitt 4.3](#)). Hierdurch können geringe Performanzeinbußen gegen einen großen Effizienzvorteil getauscht werden.

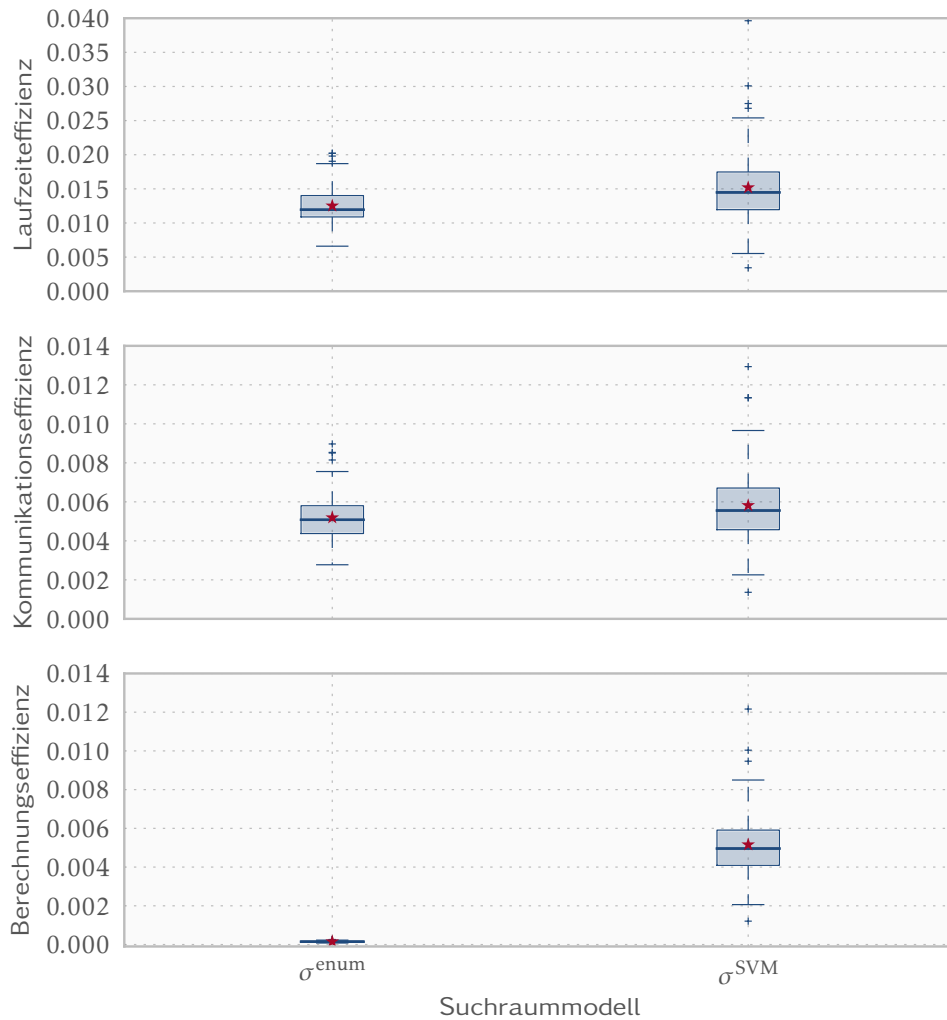


Abbildung 7.27: Laufzeiteffizienz, Kommunikationseffizienz und Berechnungseffizienz bei Simulation verschiedener Suchraummodelle. Darstellung jeweils als Box-plot sowie Mittelwert (Sternsymbol) für 100 Ausführungen je Verbundgröße bis zu einem Fehlerwert von $\epsilon = 0.01$.

7.6.4 Auktionshandel EPEX SPOT

Die *European Power Exchange* (EPEX SPOT) ist nach eigenen Angaben³ „die Börse für Stromspotmärkte im Herzen Europas. Sie deckt die Märkte Frankreich, Deutschland, Österreich und Schweiz ab.“ Dabei handelt es sich sowohl um *Day-Ahead*-Märkte, auf denen elektrische Wirkleistung für den jeweiligen Folgetag auktioniert wird, als auch um *Intraday*-Märkte, auf denen kontinuierlicher Handel mit einem minimalen Zeitfenster von 45 Minuten bis zur Lieferung betrieben wird. Im ersten Fall können Blöcke von mindestens 60 Minuten gehandelt werden, während die minimale zeitliche Auflösung im zweiten Fall 15 Minuten beträgt. Durch Kombinationen von diesen Basisblöcken können beliebig aggregierte Blockgebote erzeugt und am Markt platziert werden. Die EPEX SPOT hat hierzu eine Reihe von standardisierten Blockgeboten vorkonfiguriert. So existiert bspw. ein Block *Baseload*, der das maximal handelbare Intervall von 24 Stunden umfasst, sowie ein Block *Sun-Peak* für die Stunden 11 bis 16 eines Tages, und einige weitere. Für beide Markttypen gilt ein Mindestvolumen von $P_{el,min} = 100 \text{ kW}$ (wahlweise positive oder negative Leistung) pro Basisblock und pro Blockgebot. Im Folgenden soll demonstriert werden, dass sich die in der vorliegenden Arbeit betrachtete Verbundbildung von elektrischen Kleinanlagen samt entworfenem prädiktiven Einsatzplanungsverfahren dazu eignet, im Rahmen einer Vortagsplanung die Markteintrittsbarriere von mindestens $P_{el,min} = \pm 100 \text{ kW}$ anzubietender Leistung zu erfüllen.

Simulationsparameter

Aufgrund der Anwendungsnähe soll, im Vergleich zu [Abschnitt 7.6.1](#), im vorliegenden Fall exemplarisch die Situation an einem typischen Mittelspannungsknoten (MS-Knoten) eines ländlichen Elektrizitätsnetzes, und dabei verschieden groß dimensionierte steuerbare Erzeuger sowie steuerbare Verbraucher als potenzielle Verbundteilnehmer in Betracht gezogen werden. Hierzu wird die Szenariendefinition des Forschungsverbundes Smart Nord (siehe [Abschnitt 1.2](#)) herangezogen [[Bla+13](#)]. Diese beinhaltet Konstruktionsvorschriften zur synthetischen Erzeugung von Mittel- und Niederspannungsnetzen und legt einen Verteilungsschlüssel für verschiedene Anlagentypen für die betrachteten Netzabschnitte fest. Nach diesem Verteilungs-

³http://www.epexspot.com/de/Unternehmen/ueber_epex_spot, letzter Zugriff 6. Januar 2014

schlüssel wird eine Mikro-Blockheizkraftwerk-Durchdringung (d. h. durchschnittliche Anzahl von Anlagen je Anschlussstelle) von 0.24 % für Geräte der Leistungsklasse 1 kW P_{el} bzw. 0.48 % für Geräte der Leistungsklasse 4.7 kW P_{el} an dem betrachteten MS-Knoten postuliert. Für Wärmepumpen wird hier eine Durchdringung von 6.6 % für Geräte der Leistungsklasse 2 kW P_{el} angenommen.

Für die Mikro-Blockheizkraftwerke werden die Modelle EcoPower 1.0 und ~4.7 des Herstellers Vaillant⁴ samt Warmwasserspeichern betrachtet, da diese exakt den geforderten Leistungsklassen entsprechen. Zusätzlich wird das aktuell in der Markteinführung befindliche Modell EcoPower 20.0 als Vertreter einer größeren Leistungsklasse hinzugezogen. Die Modelle wurden prototypisch implementiert, indem ein generisches Simulationsmodell eines Blockheizkraftwerkes sowie eines Warmwasserspeichers anhand der den Datenblättern zu obigen Anlagen entnommenen Kennzahlen entsprechend parametrisiert wurden. Zu beachten ist hier, dass das Modell EcoPower 1.0 nicht modulierend ist, und daher einer einfachen Zweipunktregelung folgt. Das Modell EcoPower 4.7 hingegen kann die Motordrehzahl zwischen 1400 und 3600 Umdrehungen pro Minute in 100er Schritten variieren, wodurch sich 22 Modulationsstufen ergeben, die sich unmittelbar auf die Menge der produzierten elektrischen sowie thermischen Leistung auswirken. Für das Modell EcoPower 20.0 ist aufgrund der aktuellen Markteinführung der Anlage zum Zeitpunkt der Anfertigung dieser Arbeit noch kein Datenblatt verfügbar, deshalb wird vereinfachend eine Modulationsfähigkeit von ebenfalls 22 Stufen angenommen. Für die Wärmepumpen wurde analog zu oben für das Modell WPF 10 des Herstellers Stiebel Eltron⁵ ein generisches Simulationsmodell einer Sole/Wasser-Wärmepumpe samt Warmwasserspeicher entsprechend parametrisiert, da dieses mit $P_{el} \approx 2.3$ kW in etwa der geforderten Leistungsklasse entspricht. Diese Anlage kann ihre Leistung nicht modulieren, sondern folgt wie auch das EcoPower 1.0 einer Zweipunktregelung.

Als Datengrundlage für die Simulation wird der nach [VDI4655], unter Verwendung von Referenz-Wetterdaten [TRY3] berechnete Wärmebedarf eines Mehrfamilienhauses verwendet. Dieser wird, wie auch die Größe des Warmwasserspeichers, für jede erzeugte Instanz entsprechend der jeweiligen Anlagendimensionierung passend skaliert. Zusätzlich werden alle Anlagenparameter in geringem Maße (einige Prozent) mittels einer Normalverteilung verrauscht, um Effizienzunterschiede durch

⁴<http://www.vaillant.de>, letzter Zugriff 6. Januar 2014

⁵<http://www.stiebel-eltron.de>, letzter Zugriff 6. Januar 2014

Aufstellungsort, Alter etc. zu berücksichtigen. Alle Anlagenmodelle beinhalten die Nebenbedingung, dass ein eingenommener Betriebsmodus (Ein/Aus, Modulationsstufe) nach Möglichkeit für 60 Minuten eingehalten werden muss, bevor in einen abweichenden Betriebsmodus gewechselt werden kann. Die Suchräume der Anlagen, die diese für die in den nachfolgenden Experimenten durchgeführten Einsatzplanungen benötigen, werden anhand von 200 zufällig erzeugten Fahrplänen pro Anlage für den jeweiligen Planungshorizont gebildet.

Als Wirkleistungsprodukt kommt exemplarisch das von der EPEX SPOT standardisierte Blockgebot *Peakload* für die Stunden 9 bis 20 eines Tages zum Einsatz. Für Verbünde, die vorwiegend aus Verbrauchern bestehen, wird das Wirkleistungsprodukt als konstant zu erbringende negative Wirkleistung $P_{el}^- = -100 \text{ kW}$ modelliert, während für vorwiegend aus Erzeugern bestehende Verbünde konstant $P_{el}^+ = 100 \text{ kW}$ vorgegeben wird (die konkreten Verbundzusammensetzungen werden im nachfolgenden Abschnitt behandelt). Intern arbeiten die verwendeten Simulationsmodelle mit einer Auflösung von 1 Minute, die Fahrplan-Schnittstelle (sowohl zum Erzeugen von zufälligen Fahrplänen für die Suchräume, als auch zum Vorgeben von abzufahrenden Fahrplänen für die Anlagen) basiert auf Daten in einer Auflösung von 15 Minuten, und das an der EPEX SPOT gehandelte Blockgebot liegt in einer Auflösung von 60 Minuten vor. Aus der Sicht des Einsatzplanungsverfahrens umfasst das Blockgebot *Peakload* somit 44 Viertelstundenintervalle. Um dieses Blockgebot bestmöglich zu erfüllen, wird den Anlagen im vorliegenden Szenario erlaubt, bereits im Vorfeld ihren Betrieb im Hinblick auf das Blockgebot zu planen. So soll es ihnen bspw. möglich sein, ihre Pufferspeicher vor dem Beginn des Blockgebotes zu leeren, um mit dem Beginn des Blockgebotes möglichst optimale Zustände hinsichtlich der Erbringung des Wirkleistungsproduktes einzunehmen. Daraus folgt für das Einsatzplanungsverfahren, dass der betrachtete Planungshorizont sowohl das Blockgebot selbst, als auch den Zeitraum vom Beginn des Handelstages bis zum Beginn des Blockgebotes umfasst, und sich somit über $|T| = 80$ Viertelstundenintervalle erstreckt. [Abbildung 7.28](#) skizziert diesen Zusammenhang. Zu beachten ist hierbei, dass für die Zielfunktion der in der Einsatzplanung durchgeführten Optimierung lediglich der Bereich des Blockgebotes relevant ist. Der Bereich davor, also der Vorlauf vor Beginn des Blockgebotes, kann von den Anlagen beliebig eingeplant werden und geht nicht mit in die Bewertung der Zielfunktion mit ein.

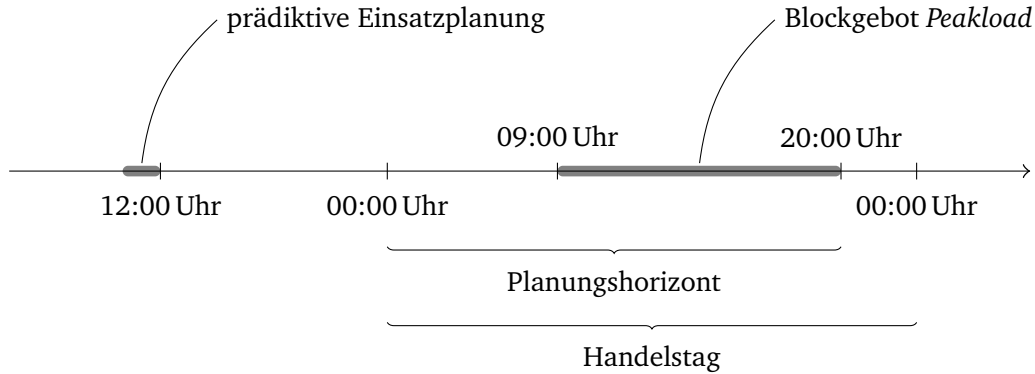


Abbildung 7.28: Betrachteter Handelstag des Szenarios „Auktionshandel EPEX SPOT“. Der Planungshorizont umfasst das elfstündige Blockgebot *Peakload*, sowie eine neunstündige, frei planbare Vorlaufzeit.

Das zu realisierende Wirkleistungsprodukt stellt sich somit je nach betrachtetem Verbund wie folgt dar (vgl. [Definition 3.1.6](#) und [3.1.7](#)):

$$\begin{aligned} \zeta^+ : \quad & \forall t \in \{t_0, \dots, t_{35}\} : t \mapsto \text{nicht definiert} \quad \wedge \\ & \forall t \in \{t_{36}, \dots, t_{79}\} : t \mapsto 100 \text{ kW} \quad \text{bzw.} \\ \zeta^- : \quad & \forall t \in \{t_0, \dots, t_{35}\} : t \mapsto \text{nicht definiert} \quad \wedge \\ & \forall t \in \{t_{36}, \dots, t_{79}\} : t \mapsto -100 \text{ kW} \end{aligned}$$

Um ein solches Wirkleistungsprodukt mit relevanten und nicht relevanten Bereichen als Zielvorgabe für das Einsatzplanungsverfahren nutzen zu können, muss das verwendete Suchraummodell jedoch bzgl. der Auswahl eines Fahrplans in der Entscheidungsphase eines Verbundteilnehmers Suchraumanfragen mit regelbasiert spezifizierten Eigenschaften unterstützen, siehe [Beispiel 7.6.1](#). Ist diese Möglichkeit nicht gegeben, kann eine Optimierung nur in begrenztem Rahmen durchgeführt werden. Da das in [Abschnitt 6.4](#) vorgestellte Suchraummodell σ^{SVM} derartige Formulierungen nicht unterstützt, wird im Folgenden ausschließlich σ^{enum} verwendet.

Beispiel 7.6.1

Sei a_i ein Teilnehmer eines Verbundes, welcher eine Optimierung für ein Blockgebot mit frei planbarer Vorlaufzeit durchführt. Eine exemplarische, zur Veranschaulichung verbalisierte Anfrage von a_i an sein Suchraummodell σ_i könnte dann etwa wie folgt aussehen:

„Gesucht ist ein Fahrplan, dessen resultierender Wirkleistungsverlauf im Intervall $[t_{36}, t_{79}]$ des Planungshorizontes möglichst hohe Werte besitzt. Die verbleibenden Zeitpunkte können frei belegt werden.“

Die eigentliche Erbringung des Wirkleistungsproduktes geschieht wie oben beschrieben an der EPEX SPOT nur in vollen Stunden, daher werden die nachfolgenden Auswertungen mit einer Auflösung von 60 Minuten erfolgen. Dazu werden sowohl die von der Einsatzplanung erzeugten Fahrpläne, als auch die von den Simulationsmodellen gewonnenen Zeitreihen im Anschluss an die Durchführung der Simulation über jeweils 60 Minuten gemittelt. Wie bereits in [Abschnitt 3.1](#) (siehe Kommentar zu [Definition 3.1.8](#) auf S. 42) beschrieben, wird hier keine Unterscheidung zwischen einer Fahrplanvorgabe und dem tatsächlichen Abfahren des Fahrplanes getroffen. Im realen Betrieb wäre zu erwarten, dass ein Fahrplan aufgrund von Prognoseungenauigkeiten nicht exakt realisiert werden kann. Da die Behandlung solcher Ereignisse aber über den Kontext der vorliegenden Arbeit hinausgeht und erst im Rahmen einer reaktiven Einsatzplanung relevant wird (vgl. [Abschnitt 3.8](#)), werden die nachfolgenden Untersuchungen wie gehabt auf *Fahrpläne* beschränkt.

Der betrachtete Handelstag ist ein Werktag im April und liegt damit in der Übergangsjahreszeit. Es wurde jeweils eine Vorlaufzeit von 14 Tagen bis zum Handelstag simuliert, um etwaige Einschwingverhalten der Modelle zu berücksichtigen, sowie eine Nachlaufzeit von 24 Stunden nach dem Handelstag, um die kurz- und mittelfristigen Auswirkungen der Einsatzplanung erfassen zu können.

Verbundzusammensetzungen

Bei durchschnittlich 1675 Anschlussstellen an einem typischen MS-Knoten eines ländlichen Elektrizitätsnetzes (entnommen aus [\[Bla+13\]](#)) resultieren aus den oben

angenommenen Durchdringungen 4 KWK-Anlagen des Typs EcoPower 1.0, 8 KWK-Anlagen des Typs EcoPower 4.7 sowie 111 Wärmepumpen des Typs WPF 10. Diese werden im vorliegenden Szenario in verschiedenen Verbundzusammensetzungen hinsichtlich des Handels an der EPEX SPOT evaluiert. Die 12 KWK-Anlagen sind zusammen aufgrund ihrer geringen Nennleistung offensichtlich nicht in der Lage, die Markteintrittsbarriere von $P_{el,min} = 100 \text{ kW}$ zu erfüllen. Um dennoch einen rein aus steuerbaren Erzeugern bestehenden Verbund betrachten zu können, werden stellvertretend 10 Anlagen des Typs EcoPower 20.0 ($P_{el} = 20 \text{ kW}$ Nennleistung) hinzugezogen. Konkret werden folgende Verbünde gebildet:

V_{10}^+ Als initiales Experiment wird ein Verbund bestehend aus 10 Mini-Blockheizkraftwerken des Typs EcoPower 20.0 betrachtet. Da die Durchdringung dieser Anlagen noch nicht bekannt ist, ist dieses Experiment hier als *Proof-of-Concept* für einen ausschließlich aus modulierenden Erzeugern bestehenden Verbund mit geringer Anlagenzahl anzusehen. Als Wirkleistungsprodukt wird ζ^+ festgelegt.

Fragestellung F_1 : Dieser Verbund soll als initiales Experiment demonstrieren, dass das Einsatzplanungsverfahren für ein Blockgebot mit frei planbarer Vorlaufzeit prinzipiell anwendbar ist.

V_{111}^- Dieser Verbund besteht aus 111 Wärmepumpen des Typs WPF 10. Im Gegensatz zu obigem Verbund handelt es sich somit ausschließlich um steuerbare Verbraucher, daher wird als Wirkleistungsprodukt ζ^- festgelegt.

Fragestellung F_2 : Mit diesem Verbund soll untersucht werden, ob das Einsatzplanungsverfahren auch unter Verwendung einer größeren Anzahl nicht-modulierender Anlagen sinnvoll eingesetzt werden kann. Insbesondere soll gezeigt werden, ob auch bei einem durch die fehlende Modulationsfähigkeit stark reduzierten Handlungsspielraum der einzelnen Anlagen ein vorgegebener Wirkleistungsverlauf produziert werden kann.

$V_{123}^\pm$ In einem dritten Experiment soll schließlich ein Verbund aus 111 Wärmepumpen des Typs WPF 10, 4 KWK-Anlagen des Typs EcoPower 1.0 sowie 8 KWK-Anlagen des Typs EcoPower 4.7 betrachtet werden. Dieser Verbund repräsentiert somit die oben beschriebene Situation an einem typischen MS-Knoten eines ländlichen Elektrizitätsnetzes. Als Wirkleistungsprodukt wird aufgrund

der in der Summe negativen Nennleistung des Gesamtverbundes wiederum ζ^- festgelegt.

Fragestellung F_3 : Da die Einflüsse der Verbraucher und Erzeuger in diesem heterogenen Verbund entgegengesetzt wirken, ist zu untersuchen wie sich die resultierende Planungsqualität im Vergleich zu dem homogenen Verbund V_{111}^- verhält.

Ergebnisse

In [Abbildung 7.29](#) ist das Ergebnis der Simulation des Verbundes V_{10}^+ visualisiert. Die Abbildung umfasst sowohl den betrachteten Handelstag (linke Hälfte) als auch den zusätzlich simulierten Folgetag (rechte Hälfte), visuell getrennt durch eine vertikale Linie in der Mitte. Das Zeitintervall des Blockgebotes *Peakload* am Handelstag ist durch eine leichte Einfärbung des Hintergrundes im linken Teil visualisiert. Die obere Grafik zeigt in diesem eingefärbten Teil die Zielvorgabe durch das Wirkleistungsprodukt als durchgezogene Linie. Der durch den von der Heuristik COHDA erzeugten Verbundfahrplan resultierende Wirkleistungsverlauf des Verbundes ist als ausgefüllte Fläche dargestellt. Zusätzlich wurde in einer zweiten Simulation der Wirkleistungsverlauf des Verbundes *ohne* Einsatzplanung ermittelt und ist als Referenz in der Abbildung als gestrichelte Linie sichtbar. Die untere Grafik zeigt die zeitlichen Verläufe der Temperaturen der Pufferspeicher der einzelnen Anlagen (im gesteuerten Betrieb, also beim Abfahren des jeweiligen Fahrplans) samt deren Mittelwert.

Es ist deutlich sichtbar, dass die Anlagen im Bereich des Blockgebotes eine um etwa 50 % höhere Leistung bereitstellen als es im ungesteuerten Verlauf der Fall wäre. Um dies zu erreichen, wurde die Leistung sowohl vor als auch nach dem Blockgebot im Vergleich zum ungesteuerten Verlauf abgesenkt – es hat also wie gewünscht eine Konzentration der Leistungsbereitstellung der Anlagen auf den Zeitraum des zu handelnden Wirkleistungsproduktes stattgefunden. Dies geschah im Rahmen des Handlungsspielraumes, indem vor dem Beginn des Blockgebotes die Pufferspeicher weitestgehend geleert wurden, um sie dann während des Blockgebotes zu füllen und damit positive Wirkleistung bereitzustellen, wie an dem Verlauf der Speichertemperaturen zu sehen ist.

Für eine quantitative Aussage zur Performanz der Einsatzplanung wird wiederum die bereits in den vorangegangenen Szenarien verwendete Summennorm heran-

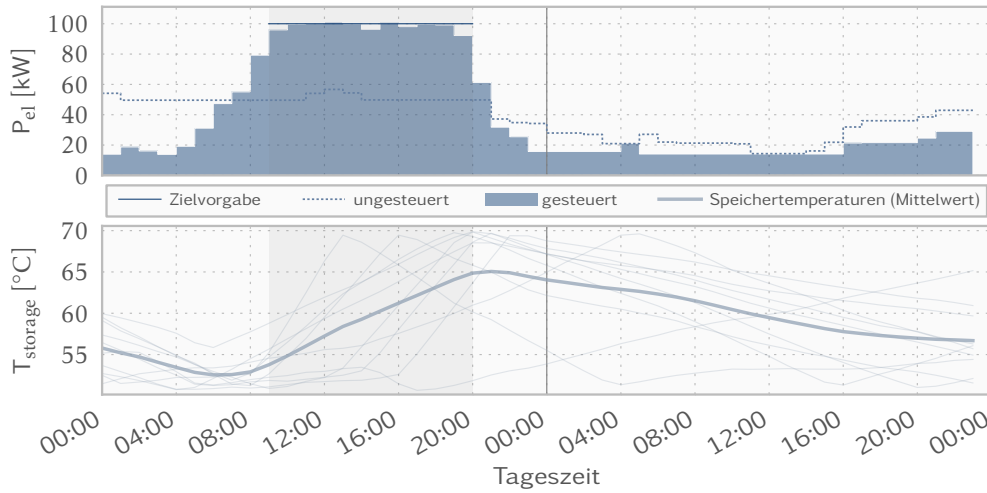


Abbildung 7.29: Ergebnis der Simulation des Verbundes V_{10}^+ für das Blockgebot *Peakload* mit konstant 100 kW als Wirkleistungsprodukt.

gezogen, und die Güte eines Verbundfahrplanes hinsichtlich der Abdeckung des Wirkleistungsproduktes in Anlehnung an [Definition 3.1.11](#) wie folgt berechnet:

$$f_{\text{EPEX}}(\zeta, \Theta) = \max\left(0, 1 - \frac{\|\zeta - \Theta\|_1}{\|\zeta\|_1}\right) \quad (7.13)$$

Nach dieser Berechnungsvorschrift hat f_{EPEX} den Wertebereich $[0, 1]$ und kann als prozentuale Abdeckung des Wirkleistungsproduktes interpretiert werden. Der ungesteuerte Verlauf des Verbundes V_{10}^+ würde demnach eine Abdeckung von 51.16 % des Wirkleistungsproduktes erreichen. Durch die Einsatzplanung konnte dies auf eine Abdeckung von 98.13 % des Wirkleistungsproduktes gesteigert werden. Die diesem Verbund assoziierte Fragestellung F_1 kann somit positiv beantwortet werden:

Das Einsatzplanungsverfahren ist prinzipiell für ein Blockgebot mit frei planbarer Vorlaufzeit anwendbar.

Dies schränkt die Anwendbarkeit des Verfahrens aber keineswegs auf Szenarien mit frei planbarer Vorlaufzeit ein. Auch ohne eine solche Vorlaufzeit könnte ein Verbund ein Blockgebot am Markt platzieren und seinen Betrieb ausschließlich im Zeitraum dieses Gebotes planen. Voruntersuchungen haben gezeigt, dass die Anlagen in die-

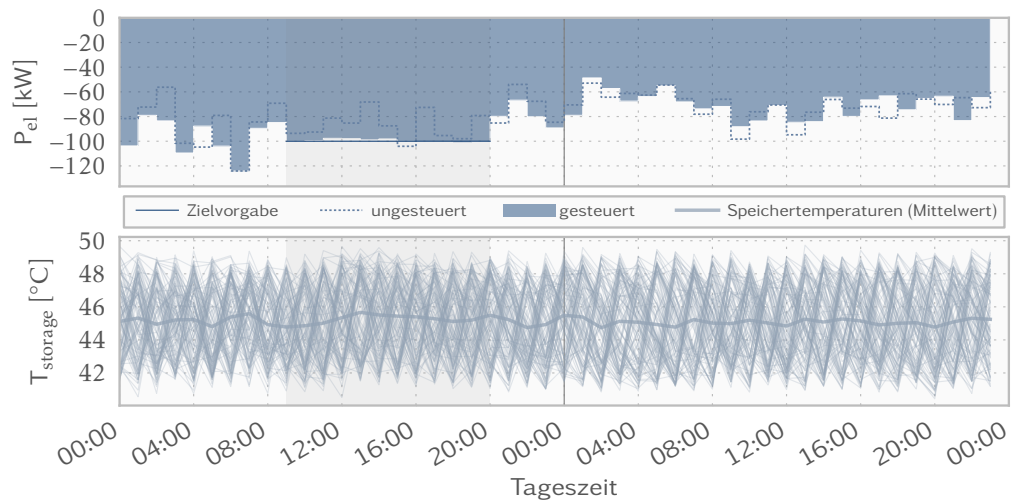


Abbildung 7.30: Ergebnis der Simulation des Verbundes V_{111}^- für das Blockgebot *Peakload* mit konstant -100 kW als Wirkleistungsprodukt.

sem Fall weniger Handlungsspielraum zur Verfügung haben würden, und je nach inhärenter Flexibilität des Verbundes mit höherer Wahrscheinlichkeit insbesondere zu Beginn des Planungshorizontes nicht in der Lage wären, das gewünschte Wirkleistungsprodukt zu erbringen.

In [Abbildung 7.30](#) ist das Ergebnis der Simulation des Verbundes V_{111}^- visualisiert. An dem vergleichsweise volatilen Lastverlauf sowie der deutlich hochfrequenten Schaltvorgänge der hier beteiligten nicht-modulierenden Anlagen (sichtbar an den kürzeren Temperaturzyklen der Pufferspeicher) ist sofort erkennbar, dass hier auf einer feingranulareren Ebene geplant werden muss als bei den trägeren und dabei zugleich flexibleren modulierenden Mini-Blockheizkraftwerken des zuvor betrachteten Verbundes V_{10}^+ . Im Ergebnis ist im Bereich des Blockgebotes zu sehen, dass der gesteuerte Lastverlauf dennoch wie gewünscht auf einem konstanten Level gehalten werden konnte, um das vorgegebene Wirkleistungsprodukt zu erfüllen. Gemäß [Gleichung 7.13](#) ergibt sich hier ungesteuert eine Abdeckung des Wirkleistungsproduktes von 86.33 %, welche durch die Einsatzplanung auf 99.14 % erhöht werden konnte. Gleichwohl ist festzuhalten, dass der Unterschied in der Abdeckung des Wirkleistungsproduktes zwischen ungesteuertem und gesteuertem Verlauf nicht so ausge-

prägt ist wie in dem Verbund V_{10}^+ . Dies hängt augenscheinlich mit der im Absolutwert höheren Nennleistung des Verbundes V_{111}^- zusammen, wodurch der ungesteuerte Verlauf naturgemäß schon näher an dem zu erfüllenden Wirkleistungsprodukt liegt. Voruntersuchungen haben aber gezeigt, dass der Verbund hier bereits an der Grenze seiner Flexibilität ist – bei einer Erhöhung des zu erfüllenden Wirkleistungsproduktes auf -130 kW (die ungesteuerte Abdeckung liegt dann bei etwa 50 %, und ist damit vergleichbar zur Situation von V_{10}^+) konnte die Einsatzplanung lediglich noch eine Abdeckung von 90.8 % erreichen. Im Fazit kann somit die reduzierte Flexibilität auch durch eine im Vergleich zu V_{10}^+ zehnfache Anlagenzahl nicht vollständig kompensiert werden. Dies ist aber weniger eine Eigenschaft des Einsatzplanungsverfahrens, sondern den Gegebenheiten des Verbundes zuzusprechen. Insofern kann Fragestellung F_2 ebenfalls positiv beantwortet werden:

Das Einsatzplanungsverfahren kann auch unter Verwendung einer größeren Anzahl nicht-modulierender Anlagen sinnvoll eingesetzt werden und ist in der Lage, auch bei einem durch die fehlende Modulationsfähigkeit stark reduzierten Handlungsspielraum der einzelnen Anlagen zufrieden stellende Ergebnisse zu liefern.

Abschließend zeigt [Abbildung 7.31](#) ist das Ergebnis der Simulation des Verbundes $V_{123}^\pm$. Die heterogene Zusammensetzung dieses Verbundes ist an den disjunkten Temperaturverläufen der Pufferspeicher zu erkennen, da die Mikro-Blockheizkraftwerke einen Temperaturbereich von $[50^\circ\text{C}, 70^\circ\text{C}]$ besitzen, während sich die Temperaturen der an die Wärmepumpen gekoppelten Pufferspeicher im Bereich $[40^\circ\text{C}, 50^\circ\text{C}]$ bewegen. In der Grafik ist der mittlere Temperaturverlauf für diese beiden Bereiche separat aufgetragen und veranschaulicht nochmals deutlich den Unterschied in der Schaltfrequenz der verschiedenen Anlagentypen. Für den Wirkleistungsverlauf des Verbundes liegt im ungesteuerten Fall eine Abdeckung des Wirkleistungsproduktes von 77.35 % vor, welche durch das Einsatzplanungsverfahren auf 99.63 % erhöht werden konnte. Offenbar führte das Hinzufügen der flexibleren Mikro-Blockheizkraftwerke trotz deren entgegengesetztem Vorzeichen im Leistungsverlauf (und der damit reduzierten Produktabdeckung im ungesteuerten Verlauf) zu einer Steigerung der Flexibilität des Gesamtverbundes. Die Fragestellung F_3 kann somit wie folgt beantwortet werden:

Die resultierende Planungsqualität in diesem heterogenen Verbund kann-

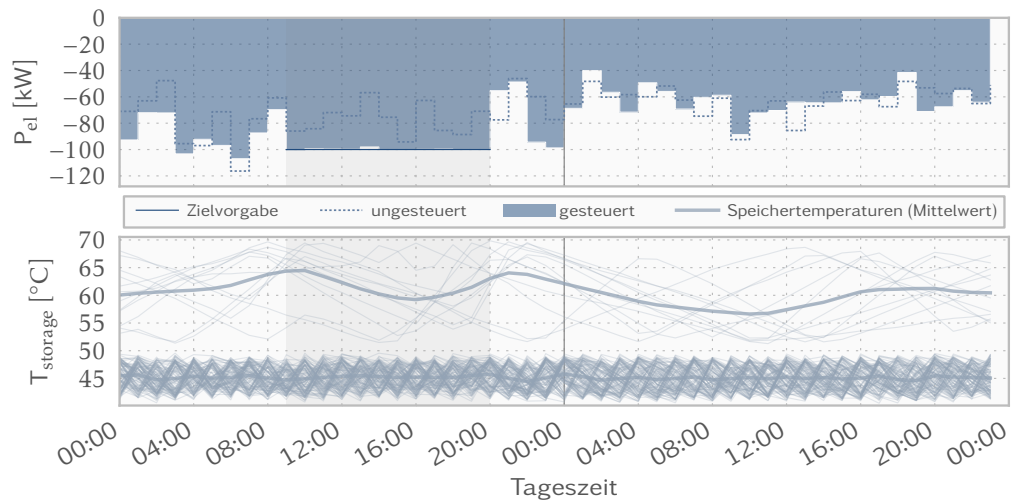


Abbildung 7.31: Ergebnis der Simulation des Verbundes $V_{123}^{\pm}$ für das Blockgebot *Peakload* mit konstant -100 kW als Wirkleistungsprodukt.

te im Vergleich zu dem homogenen Verbund V_{111}^{-} durch die gewonnene Flexibilität der hinzugefügten Mikro-Blockheizkraftwerke leicht gesteigert werden.

In der Summe konnten somit alle drei Fragestellungen positiv beantwortet werden. Zur abschließenden Bewertung dieses Szenarios soll nun noch kurz auf mögliche Nebeneffekte einer Einsatzplanung eingegangen werden.

Synchronisationseffekte

Die hier betrachtete Einsatzplanung eines Anlagenverbundes bewirkt die Orchestrierung der enthaltenen Anlagen hinsichtlich eines spezifischen Zielfahrplans. Im ungesteuerten Fall operieren die Anlagen unabhängig voneinander, und haben somit aufgrund von objektspezifischen Unterschieden wie etwa dem thermischen Bedarf des Standortes oder abweichenden Arbeitsparametern, bedingt durch Nutzerpräferenzen, zueinander asynchron verlaufende Schaltzeitpunkte. Werden jedoch Steuerungseingriffe in dieses Regelverhalten vorgenommen, so kann es zu Synchronisationen der Schaltzeitpunkte, und damit zu sichtbaren Schwingungen auf dem gemeinsamen Lastgang kommen. Während dies für den Zeitraum eines Steuerungseingriffes (wie

etwa dem Planungshorizont im vorliegenden Anwendungsfall) erwünscht und erwartet ist, kann ein schwingender Lastverlauf im Nachgang einer Steuerung jedoch hinsichtlich der Netzstabilität problematisch sein. Im Kontext von flexiblen Kleinverbrauchen wurde dies bereits untersucht, siehe etwa [Sta+07; Sta+09]. Im vorliegenden Fall ist ein solcher Synchronisationseffekt bspw. in den Simulationsergebnissen des Verbundes V_{10}^+ (Abbildung 7.29) anhand der Temperaturen der Pufferspeicher zu erkennen: Während des Planungshorizontes (d. h. von 0:00 Uhr – 20:00 Uhr des Handelstages) werden die Zustände der beteiligten Anlagen bewusst synchronisiert, um durch ein gemeinsames Auffüllen der Pufferspeicher im Zeitraum des Blockgebotes die benötigte Leistung bereitstellen zu können. Dies führt jedoch auch im Nachgang der Planung zu einer anhaltenden Synchronisation, welche sich in einem gleichmäßigen Abfall der Temperaturen und damit einhergehend in einer reduzierten eingespeisten Leistung äußert. Je nach Standortbedingungen und Anlagentyp kann diese Synchronisation noch längere Zeit anhalten und somit eine Oszillation auf dem Leistungsprofil bewirken. Gegebenenfalls kann eine solche Synchronisation mittels einer zielgerichteten Desynchronisation der Anlagenzustände nach erfolgtem Steuereingriff aufgehoben werden. In [Hin08; HVS09] wurden bereits mögliche Strategien zur Desynchronisation von thermischen Kleinverbrauchern untersucht. Diese eignen sich aufgrund der Ähnlichkeit der betrachteten Anlagen auch für den vorliegenden Anwendungsfall und sollten bei einer möglichen Umsetzung des hier vorgestellten Verfahrens berücksichtigt werden.

7.6.5 Spreizungsreduktion Tageslastprofil

Die Einsatzplanungsheuristik COHDA wurde im Hinblick auf die Erbringung eines vorgegebenen Wirkleistungsproduktes entwickelt. Die grundlegende Eigenschaft des Verfahrens besteht jedoch darin, eine verteilte kombinatorische Optimierung durchzuführen, weshalb es sich auch für von konkreten Wirkleistungsprodukten abstrahierende Zielsetzungen eignet. Im Rahmen einer vorläufigen Version der Heuristik wurde dies bereits für den Anwendungsfall der Frequenzhaltung durch einen selbstorganisierten Kraftwerkspark gezeigt [And+12]. Die dort betrachtete Problemstellung stellt eine Optimierung auf kontinuierlichen Wertebereichen unter Existenz von Nebenbedingungen dar und weicht daher stark von dem hier fokussierten Einsatzzweck ab. Daher wird die Einsatzfähigkeit des Verfahrens für ein kombinatorisches,

nicht auf Wirkleistungsprodukten basierendes Problem im Folgenden anhand eines Szenarios zur Spreizungsreduktion („Glättung“) eines Tageslastprofils demonstriert.

Simulationsparameter

Als zu glättendes Tageslastprofil wird ein synthetisches Standardlastprofil der Gruppe H0 (Haushalt) verwendet.⁶ Da dieses in normierter Form vorliegt, muss es zunächst auf die betrachtete Leistungsklasse skaliert werden. Im vorliegenden Szenario soll analog zum Szenario „EPEX SPOT“ in [Abschnitt 7.6.4](#) exemplarisch die Situation an einem typischen Mittelspannungsknoten betrachtet werden, daher werden die gleichen Voraussetzungen angenommen. Aus den enthaltenen Beispielrechnungen in der verwendeten Szenariendefinition des Forschungsverbundes Smart Nord [[Bla+13](#)] lässt sich ableiten, dass ein Knoten des Mittelspannungsnetzes im Durchschnitt 10 Niederspannungsnetze mit insgesamt 1675 Anschlussstellen subsummiert, und der mittlere Jahresverbrauch an einem solchen Knoten in der Größenordnung von 5 GWh liegt. Damit ergibt sich an einem solchen Knoten ein durchschnittlicher Tagesverbrauch von 13.6 MWh. Dieser Tagesverbrauch wird nun als Bemessungsgröße für die Skalierung des betrachteten Standardlastprofils verwendet, sodass dieses die über den Tag verteilte Last von 1675 Anschlussstellen unter einem Mittelspannungsknoten repräsentiert.

Die übrigen Simulationsparameter wurden analog zur Vorgehensweise des Szenarios „EPEX SPOT“ konfiguriert, lediglich die zeitliche Auflösung der Auswertung wurde von Stundenprodukten auf Intervalle der Länge 15 Minuten erhöht. Die Verbundzusammensetzungen V_{111}^- und V_{123}^+ wurden ebenfalls beibehalten. Da es bei der Spreizungsreduktion aber keine Markteintrittsbarriere zu überwinden gilt, wird statt des stark hypothetischen, aus 20 kW-Anlagen bestehenden Verbundes V_{10}^+ ein aus der Szenariendefinition in [[Bla+13](#)] abgeleiteter Verbund V_{12}^+ (bestehend aus 4 Anlagen vom Typ EcoPower 1.0 und 8 Anlagen vom Typ EcoPower 4.7) mit deutlich geringerer Nennleistung betrachtet.

⁶http://www.eon-mitte.com/de/netz/veroeffentlichungen/strom_/standardlastprofil_verfahren, letzter Zugriff 6. Januar 2014

Zielfunktion und Suchraummodell

Um eine Reduktion der Lastgangspreizung zu erreichen, wird die Zielfunktion der Optimierung wie folgt definiert.

Definition 7.6.1 (Optimierungsziel Spreizungsreduktion)

Sei A ein Verbund, T ein Planungshorizont und π_{SLP} ein über T prognostiziertes Tageslastprofil. Ferner sei $\hat{\pi}_A$ der für den Verbund prognostizierte Wirkleistungsverlauf im ungesteuerten Fall und es gelte $\hat{\pi}_A \in \pi_{\text{SLP}}$, d. h. das Tageslastprofil berücksichtigt bereits den prognostizierten Wirkleistungsverlauf des Verbundes. Gesucht ist nun ein Verbundfahrplan Θ_{opt} als Alternative zu $\hat{\pi}_A$, welcher die Spreizung von π_{SLP} minimiert:

$$\Theta_{\text{opt}} = \arg \min_{\Theta} (\text{range}(\pi_{\text{SLP}} - \hat{\pi}_A + \Theta)) \quad (7.14)$$

mit

$$\text{range}(\pi) = \max(\pi) - \min(\pi) \quad (7.15)$$

Der in dieser Zielfunktion erforderliche prognostizierte Wirkleistungsverlauf des Verbundes im ungesteuerten Fall kann vom Sprecher des Verbundes vor Beginn der Einsatzplanung erhoben werden (vgl. [Abschnitt 4.1](#)). Mit dieser Formulierung der Zielfunktion kann die Einsatzplanungsheuristik nun wie gewohnt durchgeführt werden. Analog zum Szenario „EPEX SPOT“ sollte das zum Einsatz kommende Suchraummodell bzgl. der Auswahl eines Fahrplans in der Entscheidungsphase eines Verbundteilnehmers Suchraumanfragen mit regelbasiert spezifizierten Eigenschaften unterstützen. Daher wird wie auch im Szenario „EPEX SPOT“ im Folgenden ausschließlich das Suchraummodell σ^{enum} verwendet.

Ergebnisse

Das Ergebnis der Simulation des Verbundes V_{12}^+ ist in [Abbildung 7.32](#) dargestellt. Die Grafik zeigt im oberen Teil wie bereits im vorhergehenden Szenario den ungesteuerten und gesteuerten Wirkleistungsverlauf des Verbundes, sowie im mittleren Teil die Verläufe der einzelnen Pufferspeichertemperaturen. Ergänzend beinhaltet der untere Teil der Abbildung das ursprüngliche zu optimierende Tageslastprofil π_{SLP} (in der Legende als „ungesteuert“ betitelt) sowie das durch die Einsatzplanung veränderte

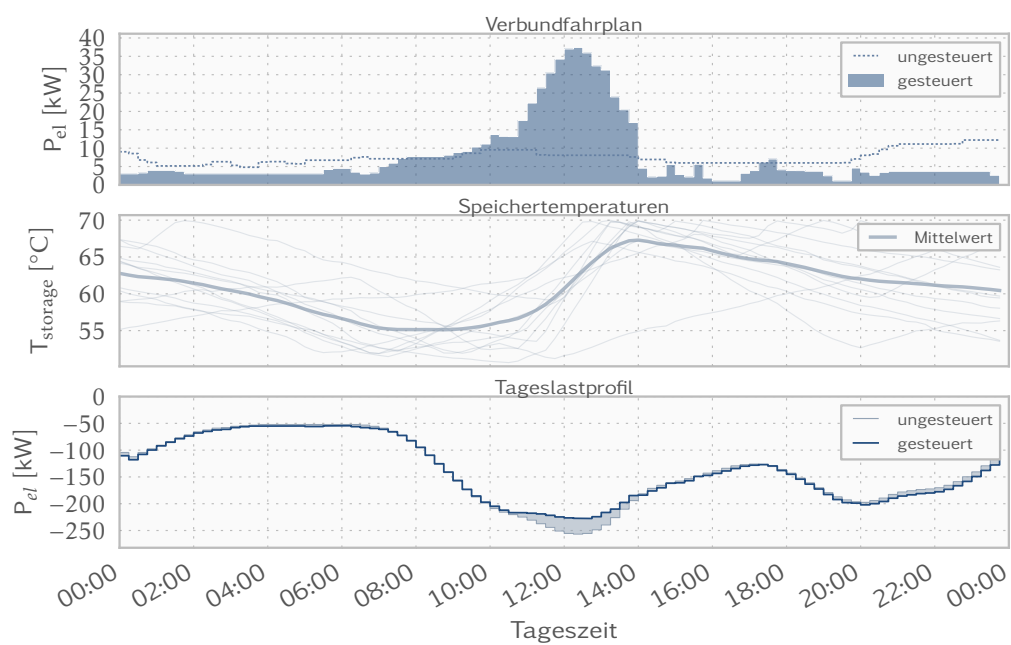


Abbildung 7.32: Ergebnis der Simulation des Verbundes V_{12}^+ im Szenario „Spreizungsreduktion Tageslastprofil“.

Tageslastprofil π_{SLP}' (in der Legende als „gesteuert“ betitelt). Die erreichte Spreizungsreduktion ist als Differenz zwischen diesen beiden Verläufen als ausgefüllte Fläche dargestellt. Die Einsatzplanung war offenbar für den betrachteten Verbund in der Lage, die Spreizung des Tageslastprofils durch eine gegenläufige Leistungsbereitstellung insbesondere zur Mittagszeit sichtbar zu reduzieren. Zur Quantifizierung kann die verbleibende Spreizung des resultierenden Tageslastprofils mittels [Gleichung 7.15](#) normiert und dann prozentual interpretiert werden. Hierzu wird die Spreizung von π_{SLP} als untere Grenze (d. h. gar keine Spreizungsreduktion, bzw. 0 %) und ein konstantes Tageslastprofil ohne jegliche Spreizung als obere Grenze (d. h. eine höchstmögliche Spreizungsreduktion, also 100 %) definiert. Nach dieser Rechnung konnte die Einsatzplanung im Verbund V_{12}^+ eine Spreizungsreduktion von 15.29 % bewirken.

[Abbildung 7.33](#) zeigt analog dazu das Ergebnis der Simulation des Verbundes V_{111}^- . Auch mit diesem Verbund aus nicht-modulierenden, hochfrequent schaltenden Verbrauchern konnte die Einsatzplanung eine deutliche Spreizungsreduktion erzielen. Im Gegensatz zum reinen Erzeuger-Verbund V_{12}^+ wurde dies hier maßgeblich durch eine *Verringerung* der Leistungsaufnahme in den Mittagsstunden erreicht. Die resultierende Spreizungsreduktion beträgt 18.32 % und liegt damit leicht über dem Niveau von V_{12}^+ . Es ist jedoch zu bemerken, dass das resultierende Tageslastprofil im mittleren Leistungsbereich (z. B. in den Abendstunden) deutlich unregelmäßiger verläuft. Dies ist der Formulierung des Optimierungsziels geschuldet, da dieses im vorliegenden Szenario lediglich die Spreizung, d. h. die Minimal- und Maximalwerte des Lastprofils bewertet, und den übrigen Verlauf nicht in die Bewertung einfließen lässt. Dies ließe sich durch Umformulierung der Zielfunktion in [Definition 7.6.1](#) aber leicht ergänzen.

Abschließend zeigt [Abbildung 7.34](#) das Ergebnis der Simulation des heterogenen Verbundes $V_{123}^\pm$. Offensichtlich konnten die unterschiedlichen Freiheitsgrade der steuerbaren Erzeuger und Verbraucher durch die Einsatzplanung sinnvoll kombiniert werden, wodurch sich hier eine Spreizungsreduktion von 25.2 % ergibt.

Insgesamt kann das in dieser Arbeit entwickelte Einsatzplanungsverfahren COHDA durch entsprechende Umformulierung der Zielfunktion somit auch für verwandte kombinatorische, nicht auf Wirkleistungsprodukten basierende Einsatzplanungsprobleme eingesetzt werden.

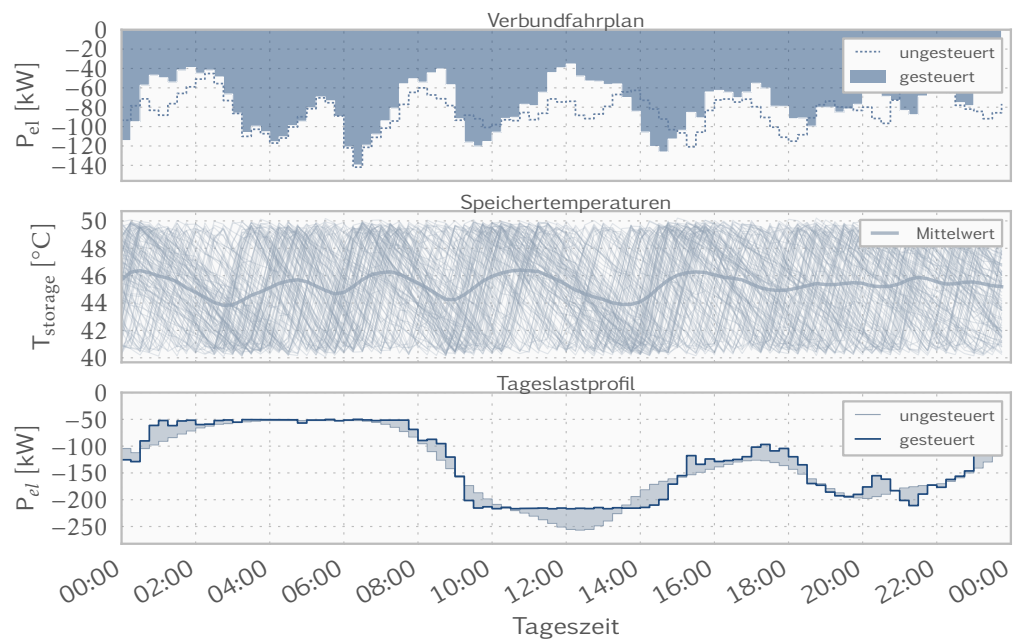


Abbildung 7.33: Ergebnis der Simulation des Verbundes V_{111}^- im Szenario „Spreizungsreduktion Tageslastprofil“.

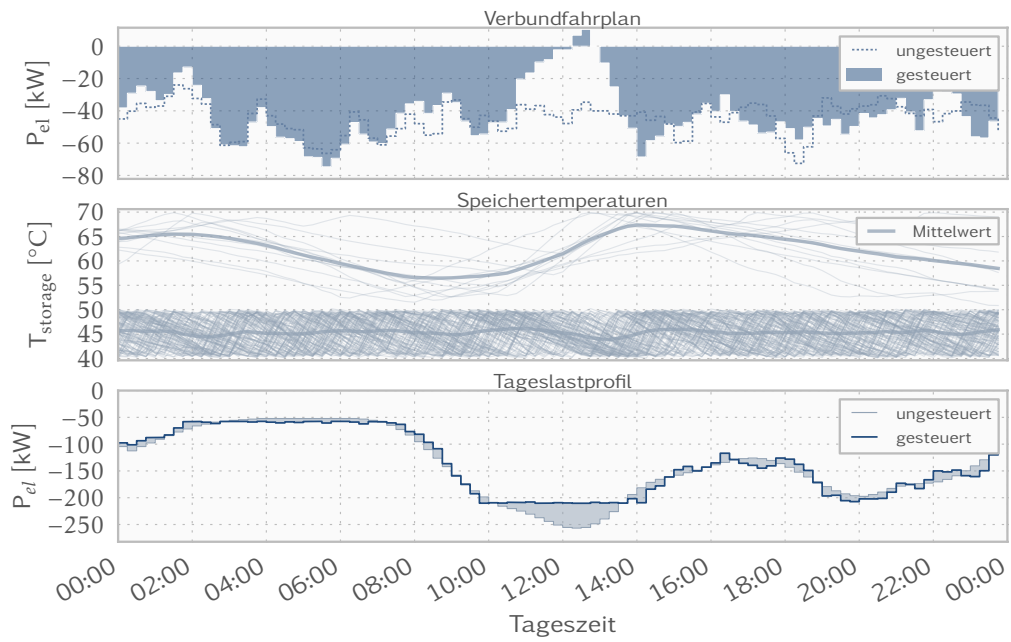


Abbildung 7.34: Ergebnis der Simulation des Verbundes $V_{123}^{\pm}$ im Szenario „Spreizungsreduktion Tageslastprofil“.

7.7 VERGLEICH MIT EXISTIERENDEN VERFAHREN

Das in dieser Arbeit vorgestellte Konzept zur selbstorganisierten Konsensbildung zeichnet sich durch eine vollständige Verteiltheit, Asynchronität sowie Berücksichtigung von verzögerungsbehafteten Kommunikationsnetzen aus, und hebt sich damit von den in [Abschnitt 1.3](#) diskutierten verwandten Arbeiten ab. Von den dort vorgestellten Verfahren ist der *Stigspace* Ansatz nach [\[LPJ10\]](#) am ehesten mit der vorliegenden Methodik vergleichbar, da dort ebenfalls eine asynchrone, iterative Konsensbildung durch Informationsaustausch erfolgt. Zwar ist das Vorgehen speziell auf das dort betrachtete Szenario der Spitzenlastreduktion unter Verwendung von zweipunktgeregelten thermischen Geräten ausgerichtet, eine prädiktive Einsatzplanung hinsichtlich eines Wirkleistungsproduktes sollte unter Verwendung des zugrundeliegenden Koordinationsparadigmas aber möglich sein. Zugleich entspricht dieses Koordinationsparadigma bei entsprechender Konfiguration der in der Einleitung der Arbeit ([Abschnitt 1.2](#)) als Motivation herangezogenen sequenziellen Heuristik aus [\[BS12\]](#). Daher wird dieses Verfahren im Folgenden im direkten Vergleich mit der COHDA Heuristik diskutiert.

Im Gegensatz zu COHDA existiert bei dem *Stigspace* Ansatz keine explizite Nachbarschaftsbeziehung zwischen den Verbundteilnehmern, und damit kein direkter Nachrichtenaustausch zwischen Agenten. Stattdessen wird eine gemeinsame Kommunikationsplattform eingesetzt, das sog. *Stigspace*, welches von allen Agenten zugreifbar ist. Der ausgeführte Algorithmus entspricht einer iterativen Annäherung der Beiträge der einzelnen Verbundteilnehmer an das globale Ziel, bis kein Agent mehr eine Verbesserung vornehmen kann. Dies gleicht dem Grundgedanken, wie er auch in COHDA umgesetzt ist. Während COHDA aber explizit eine auf Interaktionen der Verbundteilnehmer basierende Koordination herbeiführt, und durch den Austausch von Lösungskandidaten die Konvergenz und Terminierung garantiert, soll das Optimierungsziel im *Stigspace* Ansatz durch ausschließlich indirekte Koordination über das *Stigspace* erreicht werden. Die Konvergenz des Verfahrens wird simulativ belegt, eine Terminierung kann aber nicht garantiert werden – daher wird neben der natürlichen Terminierung des Verfahrens durch Inaktivität der Agenten zusätzlich ein artifizielles Abbruchkriterium in Form einer zeitlichen Begrenzung der Ausführung des Verfahrens eingesetzt. Weiterhin ist die *anytime* Eigenschaft nicht gegeben, da die Lösungsbeiträge der teilnehmenden Akteure auf zufälligen Entscheidungen ba-

sieren, welche unmittelbar in die aktuell bestehende Lösung des Systems einfließen (d. h. es existieren keine Lösungskandidaten), sodass sich die vom System produzierte Lösungsgüte im Laufe des Verfahrens durchaus verschlechtern kann, wenn mehrere parallele Modifikationen stattfinden, die gemeinsam zu einer Verringerung der Zielerreichung führen.

Das Verfahren wurde zur detaillierteren Betrachtung in der in [Kapitel 6](#) beschriebenen Simulationsumgebung implementiert. Die konkrete technische Realisierung des zentralen *Stigspace* für die in [\[LPJ10\]](#) durchgeführte Evaluation ist jedoch nicht dokumentiert, die Patentschrift [\[LPJ08\]](#) beinhaltet zu dieser Frage den folgenden Satz:

The precise implementation of the repository 106 and corresponding data links 108, 110 is not critical, so long as each agent, eg 102, 104, is able to post information and messages to the repository 106, and retrieve information and messages posted to the repository 106 by other agents.

Damit bleibt jedoch insbesondere offen, auf welche Weise parallele Zugriffe in dem System behandelt werden – in der Terminologie nach [\[Cha+11\]](#) ist somit der *Adjustment Schedule* des Verfahrens nicht vollständig definiert. Zwei prinzipielle Möglichkeiten hierzu sind wie folgt denkbar:

1. Parallele Zugriffe werden sequenzialisiert, dies ist etwa durch Warteschlangen realisierbar.
2. Parallele Zugriffe arbeiten auf Schnappschüssen der Daten. Alle darin enthaltenen Schreiboperationen werden dann anschließend in einer isolierten atomaren Aktion auf die realen Daten angewendet. Da jeder Agent im betrachteten System nur Informationen über sich selbst modifiziert, können hierbei keine Konflikte auftreten.

Die erste Option würde jedoch die von den Autoren postulierte Asynchronität verletzen. Daher wurde hier die zweite Option betrachtet, und das *Stigspace* in Form eines zentralen Datenspeichers mit Schnappschuss-basierten Zugriffen wie oben beschrieben implementiert. Dabei konnten unter Verwendung des in [Abschnitt 7.6.1](#) modellierten Verbundes aus Mikro-Blockheizkraftwerken folgende Erkenntnisse erzielt werden:

- Mit konstanter Reaktionszeit der Verbundteilnehmer sowie konstanter Nachrichtenverzögerung ergibt sich der Effekt, dass alle Agenten gleichzeitig zunächst lesend auf das *Stigspace* zugreifen, einen neuen eigenen Lösungsbeitrag auswählen und diese Entscheidung dann wiederum alle gleichzeitig an das *Stigspace* senden. Mit diesem Verhaltensmuster konnte in den durchgeführten Simulationen allerdings keine Konvergenz erzielt werden. In 40 % der Fälle begann das System zwischen zwei Lösungen zu oszillieren. Die restlichen Simulationen wurden nach jeweils 100 Simulationsschritten abgebrochen, da kein konvergierendes Verhalten erkennbar war. Alle Simulationen endeten in Konfigurationen mit im Vergleich zur initialen Startlösung des Systems deutlich schlechteren Lösungsgüten. Es kann somit bei starker Parallelität keine Koordination herbeigeführt werden.
- Durch Erhöhung und Diversifizierung der Reaktionszeiten oder Nachrichtenverzögerungen kann die Parallelität reduziert werden, da die *Stigspace* Zugriffe der Verbundteilnehmer hierdurch auf verschiedene Zeitpunkte verteilt werden. Im vorliegenden Fall wird dann in den produzierten Lösungen über den Simulationsverlauf hinweg ein konvergierender Trend sichtbar, d. h. die produzierten Lösungen nähern sich – nicht monoton, aber im Mittel – im Laufe einer Simulation der optimalen Lösung an. Die selbstständige Terminierung des Verfahrens ergibt sich jedoch erst mit einer starken Diversifizierung der Aktionszeitpunkte in der Größenordnung $r \approx m$ (Größe des Verbundes). Dies bewirkt, dass kaum parallele Zugriffe auf das *Stigspace* erfolgen. Der Median des verbleibenden Fehlers über alle 100 durchgeführten Simulationen in dieser Konfiguration beträgt $\tilde{e} = 0.17$, IQR = 0.03 (siehe [Abbildung 7.35](#)) und liegt damit auf dem Niveau der Ergebnisse mit der COHDA Heuristik aus [Abschnitt 7.6.1](#).

Die Erhöhung der Reaktionszeiten auf $r \approx m$ realisiert damit allerdings gerade die oben in der ersten Option genannte Sequentialisierung der *Stigspace* Zugriffe. Dies wiederum entspricht im Grunde dem bereits in der *Random Walk* Heuristik ([Abschnitt 7.1.2](#)) sowie in der zu Beginn der Arbeit referenzierten sequenziellen Heuristik aus [BS12] eingesetzten Rundlauf-Scheduling und ermöglicht es somit den Agenten, stets mit aktuellstem Wissen zu arbeiten. Offenbar ist das Verfahren in Ermangelung einer expliziten Koordination zwischen Verbundteilnehmern entgegen der oben referenzierten Aussagen insgesamt also nicht in der Lage, in einem verteil-

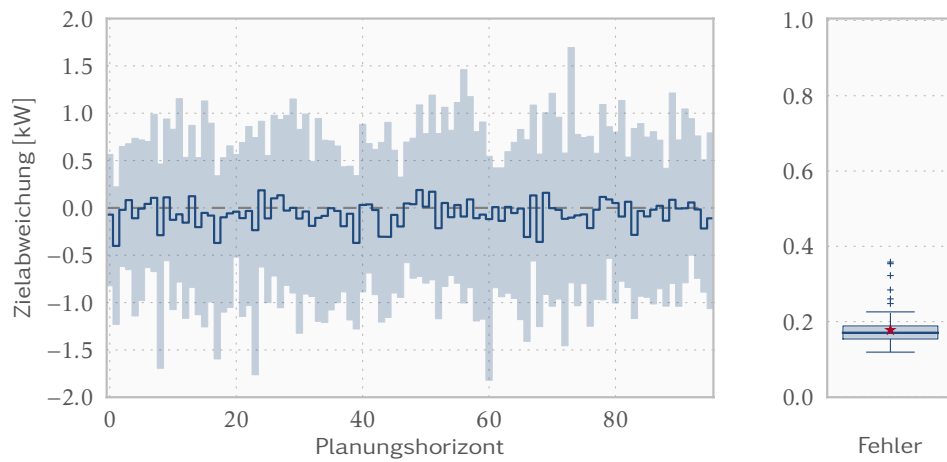


Abbildung 7.35: Resultat von 100 Simulationen des *Stigspace* Ansatzes mit reduzierter Parallelität für einen Verbund aus 30 Mikro-Blockheizkraftwerken und zufälligem, aber realisierbarem Wirkleistungsprodukt. Zielabweichung (links) dargestellt als Median und oberes sowie unteres Quartil. Verbleibender Fehler für die Gesamtlösung (rechts) dargestellt als Boxplot und Mittelwert (Sternsymbol).

ten, asynchronen Kontext zuverlässige Ergebnisse zu liefern. Wird die Voraussetzung der Asynchronität fallen gelassen, so liefert das *Stigspace* Verfahren im betrachteten Szenario etwa die gleiche Lösungsqualität wie COHDA.

8

SCHLUSS

New ideas pass through three periods:

- 1) It can't be done.
- 2) It probably can be done, but it's not worth doing.
- 3) I knew it was a good idea all along!

(Sir Arthur C. Clarke)

Der derzeit stattfindende Wandel des Energieversorgungssystems zu einem von dezentralen Energiewandlungsanlagen geprägten System mit starkem Fokus auf erneuerbare Energiequellen führt zu einem Anpassungsbedarf der historisch gewachsenen, zentralisierten Kontrollstrukturen dieses Systems. Eine im Wissenschaftskontext aktuell vielfach diskutierte Möglichkeit, diesem Bedarf zu begegnen, ist die Flexibilisierung des Gesamtsystems durch Nutzbarmachung der Freiheitsgrade dezentraler Energieanlagen auf Basis einer umfassenden informations- und kommunikationstechnischen Infrastruktur. Die vorliegende Arbeit liefert einen Beitrag zu dieser Diskussion, indem mit COHDA ein Verfahren zur prädiktiven fahrplanbasierten Einsatzplanung solcher Anlagen vorgeschlagen wird, welches aufgrund seiner vollständig verteilt und asynchron durchgeführten Konsensbildung einerseits die Autonomie der beteiligten Akteure gewährleistet, und andererseits durch die Erfüllung der *anytime* Eigenschaft und der damit garantierten Konvergenz eine Planungssicherheit sowohl für Anlagenbetreiber als auch für Netzbetreiber ermöglicht. Im Folgenden werden die in dieser Arbeit erzielten Ergebnisse nochmals kurz zusammengefasst.

8.1 ERGEBNISSE

Zur strukturierten Betrachtung der Ergebnisse dieser Arbeit werden die *Design-Science Research Guidelines* nach Hevner et al. [Hev+04] herangezogen. Diese bestehen aus sieben Richtlinien und sind zur Übersicht in [Tabelle 8.1](#) zitiert. Im Folgenden werden diese in angepasster Reihenfolge mit den durchgeführten Schritten dieser Arbeit in Beziehung gesetzt.

Relevanz (Problem Relevance) Die Durchführung dieser Forschungsarbeit wurde durch aktuelle Entwicklungen in der elektrischen Energieversorgung motiviert. Der

Guideline	Description
Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Design as an Artefact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Tabelle 8.1: *Design-Science Research Guidelines* nach [Hev+04, Tab. 1] (Reihenfolge verändert).

Fokus lag hierbei auf der unter anderem politisch motivierten verstärkten Einbindung dezentraler Energiewandlungsanlagen (bspw. KWK-Anlagen) in das deutsche Elektrizitätsnetz, und der damit verbundenen Forderung nach der Nutzbarmachung der Flexibilität dieser Anlagen hinsichtlich eines systemstabilisierenden Betriebs ([Kapitel 1](#)). Die Relevanz eines vollständig verteilten Koordinationsansatzes hierzu wurde anhand verschiedener Expertenmeinungen belegt, nach denen die Flexibilisierung des Systems mit einer Dezentralisierung der Kontrollstrukturen einhergehen muss. Dabei wurden diverse zu erhaltende Eigenschaften (wie etwa die Autonomie der Anlagenbetreiber und der Datenschutz) als wesentliche Herausforderungen identifiziert ([Abschnitt 1.1](#)). Nach Betrachtung existierender Lösungsansätze zu vergleichbaren Problemstellungen ([Abschnitt 1.3](#)) wurde schließlich das konkrete Forschungsziel der Arbeit formuliert ([Abschnitt 1.4](#)).

Lösungsansatz (*Design as a Search Process*) Durch die formale Betrachtung der gegebenen Problemstellung wurde eine vom Anwendungskontext losgelöste Formulierung gefunden, welche zur Beschreibung des zugrundeliegenden mathematischen Problems in Form der verteilten kombinatorischen Optimierung führte ([Abschnitt 2.1](#)). Dies ermöglichte die Suche nach geeigneten Lösungsansätzen aus verwandten Disziplinen ([Abschnitte 2.1.1 bis 2.1.4](#)). Diese Suche verlief erfolglos, woraus sich der Bedarf nach einer Neuentwicklung ergab. Als Mittel zur technischen Umsetzung wurden Multiagentensysteme als geeignet befunden ([Abschnitt 2.2](#)), während für die konzeptuelle Herangehensweise Methoden der Selbstorganisation identifiziert wurden ([Abschnitt 2.3](#)).

Methoden (*Research Rigor*) Zur strukturierten Neuentwicklung des angestrebten Koordinationsansatzes wurde nach eingehender Betrachtung unterschiedlicher Möglichkeiten die Hybridisierung eines deskriptiven mit einem konstruktiven Entwicklungsmodell als Vorgehensweise ausgewählt ([Kapitel 3](#)). Die Anwendung dieses Modells resultierte zunächst in einer formalen Beschreibung aller am System beteiligten Entitäten und ihrer grundlegenden Eigenschaften ([Abschnitte 3.1 bis 3.3](#)). Anschließend erfolgte die schrittweise Ausgestaltung des Koordinationsansatzes unter Berücksichtigung der einleitend formulierten Ziele ([Abschnitte 3.4 bis 3.9](#)).

Artefakt (*Design as an Artefact*) Das aus den vorhergehenden Schritten resultierende Artefakt ist die Heuristik COHDA, ein Verfahren zur Lösung verteilter kombinatorischer Optimierungsprobleme ([Kapitel 4](#)). Der primäre Anwendungszweck ist die prädiktive fahrplanbasierte Einsatzplanung dezentraler Energiewandlungsanlagen im Smart Grid.

Evaluation (*Design Evaluation*) Zur kritischen Untersuchung des entwickelten Artefaktes hinsichtlich der Erfüllung der gesetzten Ziele und damit der Einsatzfähigkeit im fokussierten Anwendungsgebiet wurde die Heuristik COHDA zuerst formal auf ihre Konvergenzeigenschaften untersucht. Hierbei wurde sowohl die *anytime* Eigenschaft, als auch die garantierte Konvergenz unter verschiedenen Bedingungen nachgewiesen ([Kapitel 5](#)). Mittels einer prototypischen Umsetzung des Verfahrens ([Kapitel 6](#)) konnten dann umfangreiche Simulationsstudien durchgeführt werden. Hierzu wurden zunächst geeignete synthetische Probleminstanzen erzeugt ([Abschnitt 7.1](#)), mit denen das grundlegende Laufzeitverhalten betrachtet ([Abschnitt 7.2](#)) und Untersuchungen zur Robustheit ([Abschnitt 7.3](#)) sowie zur Skalierbarkeit ([Abschnitt 7.4](#)) durchgeführt werden konnten. Neben qualitativen Beobachtungen wurden zudem quantitative Zusammenhänge mittels Regressionstests ermittelt ([Abschnitt 7.5](#)). Weitere simulative Untersuchungen anhand von domänenspezifischen Szenarien belegten die Einsatzfähigkeit des Verfahrens im motivierten Anwendungskontext. Neben Studien zur prinzipiellen Funktionsweise des Verfahrens ([Abschnitt 7.6.1](#)) sowie zur ergänzenden Betrachtung der Skalierbarkeit ([Abschnitt 7.6.2](#)) und verschiedener Suchraummodelle ([Abschnitt 7.6.3](#)) wurden mit dem Auktionshandel an der Strombörse EPEX SPOT ([Abschnitt 7.6.4](#)) und der Spreizungsreduktion eines synthetischen Tageslastprofils ([Abschnitt 7.6.5](#)) auch exemplarische Szenarien aus der aktuellen Energiewirtschaft demonstriert. Abschließend erfolgte ein Vergleich mit einem alternativen Ansatz ([Abschnitt 7.7](#)).

Erkenntnisgewinn (*Research Contributions*) Mit COHDA wurde ein Verfahren zur selbstorganisierten Einsatzplanung dezentraler Akteure im Smart Grid entwickelt, das alle zu Beginn der Arbeit formulierten Anforderungen erfüllt (vgl. [Abschnitt 1.4](#)): Mittels einer vollständig verteilten Koordinationsstrategie können die Flexibilitäten dezentraler Energiewandlungsanlagen im Rahmen einer prädiktiven fahrplanbasierten Einsatzplanung nutzbar gemacht werden. Dabei liegen die für die Durchführung

des Verfahrens benötigten Informationen dezentral vor, und auch die ausgeführten Aktionen der eingebundenen Agenten werden nicht von übergeordneter Stelle koordiniert. Lediglich der Start der Einsatzplanung, d. h. Bilden und Bekanntgeben des Overlay-Netzes samt Optimierungsziel, muss von einem einzelnen Akteur durchgeführt werden, welcher aber beliebig bestimmt werden kann (siehe [Abschnitt 4.1](#)). Die Interessen der jeweiligen, am Prozess beteiligten Anlagenbetreiber bleiben in Form von Autonomie und Datenschutz gewahrt. Dies wird durch die vollständige Verteiltheit in Verbindung mit der sog. freien Gesinnungswahl erreicht, wodurch die Akteure selbst entscheiden können, welche realisierbaren Fahrpläne sie in das Verfahren einbringen. Die von einem Akteur veröffentlichten Informationen werden zwar innerhalb des Verbundes weitergereicht und können von anderen Akteuren gespeichert und verarbeitet werden, allerdings beinhalten diese lediglich die durch den Akteur im Rahmen der Lösungssuche vorgeschlagenen Fahrpläne ohne individuelle Informationen bzgl. dessen persönlicher Bewertung oder ähnlichem, und stellen nur einen kleinen Ausschnitt seines eigentlichen Handlungsspielraumes dar. Mit der zusätzlich eingeschränkten zeitlichen Gültigkeit der Daten im betrachteten Planungshorizont sind diese somit nicht sinnvoll für missbräuchliche Handlungen verwendbar. Trotz dieser Eigenschaften ermöglicht das Verfahren eine Planungssicherheit für Anlagenbetreiber und Netzbetreiber, da mit der *anytime* Eigenschaft und der damit einhergehenden Konvergenz sichergestellt ist, dass das Verfahren nach einer kurzen Einschwingphase (siehe [Abschnitt 5.4](#)) ausschließlich gültige Verbundfahrpläne produziert, die sich im weiteren Verlauf der Heuristik nur verbessern können. Es konnte bewiesen werden, dass das Verfahren mit der besten gefundenen Lösung in einem konsistenten Zustand terminiert, in welchem jedem Verbundteilnehmer der individuell auszuführende Fahrplan bereits bekannt ist. Eine nachträgliche Bekanntmachung der finalen Lösung durch eine zentrale Komponente entfällt somit. Weiterhin wurde in der Zielsetzung der Arbeit eine hohe Effizienz bzgl. Berechenbarkeit, Zeitaufwand und Kommunikationsaufwand gefordert. Diese wurden jeweils durch ausführliche Simulationsstudien belegt. Durch die explizit modellierte Asynchronität samt Berücksichtigung verzögerungsbehafteter Kommunikationsnetze sowie unterschiedlicher Reaktionszeiten der Verbundteilnehmer konnte auch die Robustheit des Verfahrens gegenüber der im fokussierten Einsatzgebiet inhärent vorliegenden Dynamik autonomer dezentraler Akteure erreicht werden. Die Unabhängigkeit von einer bestimmten, starren Kommunikationsstruktur ermöglicht dabei den Einsatz der für den jeweiligen

Anwendungsfall optimalen Topologie.

Aus algorithmischer Sicht wurde ein Verfahren entwickelt, mit der eine Menge von autonomen Akteuren ein gemeinsames, nicht separierbares Optimierungsproblem lösen kann. Zur zielgerichteten Lösung solcher nicht separierbarer Probleme ist es erforderlich, dass die daran beteiligten dezentralen Akteure Informationen austauschen, da sie ansonsten nicht ermitteln können, ob sich ihre individuellen Entscheidungen positiv oder negativ auf die Gesamtlösung auswirken. Im vorliegenden Fall ist der Constraint-Graph des Problems vollvernetzt (siehe [Abschnitt 2.1](#)), sodass prinzipiell *alle* Akteure jeweils miteinander kommunizieren müssen um das Problem zu lösen, da sich die Entscheidung eines einzelnen Akteurs bzgl. der zu optimierenden Zielfunktion auf alle anderen auswirkt. Mit COHDA wurde ein Ansatz entwickelt, der diesen Informationsaustausch so gering wie möglich hält, und dennoch eine effiziente Suche im Lösungsraum ermöglicht. Dabei konnte der Kommunikationsaufwand in Relation zur Populationsgröße von ursprünglich quadratisch (im vollvernetzten Constraint-Graph) auf eine gebrochene Konvergenzordnung im leicht überlinearen Bereich reduziert werden ([Abschnitt 7.6.2](#)). Hierzu wurde auf Techniken der Selbstorganisation zurückgegriffen: Durch die Einschränkung der Kommunikationstopologie auf ein im Vergleich zum vollvernetzten Constraint-Graph reduziertes Overlay-Netz wurden lokale Sichten der beteiligten Akteure auf das Gesamtsystem erzeugt, sodass zwei im Overlay-Netz nicht benachbarte Akteure prinzipiell voneinander unabhängige Entscheidungen treffen können. Zusammen mit der Informationssynthese realisiert dies die in [Abschnitt 2.3.1](#) beschriebene positive Rückkopplung, wodurch mehrere Akteure im System parallel verschiedene Bereiche des Lösungsraumes durchsuchen. Mittels des Austausches von guten Zwischenlösungen in Form von Lösungskandidaten wird dem entgegengesetzt zudem die negative Rückkopplung realisiert, wodurch die Dynamik des Systems wiederum gedämpft wird und somit ein Equilibrium angestrebt wird. Diese beiden Kräfte sind auch als *diversification/intensification* oder als *exploration/exploitation* bekannt (siehe z. B. [\[BR03\]](#)). Insgesamt ermöglicht COHDA somit eine vollständig verteilte, d. h. selbstorganisierte Konsensbildung autonomer Akteure. In der auf Metaheuristiken fokussierten Taxonomie nach Talbi stellt dies eine Realisierung gemäß dem *Cooperative Algorithmic-Level Parallel Model* dar, da mehrere parallel ausgeführte Instanzen zur Problemlösung miteinander interagieren, um auf kooperative Weise zu einem Gesamtergebnis zu gelangen [[Tal09](#), Abs. 6.1.1.2]. Aus der Sicht von DCOP

Algorithmen (siehe [Abschnitt 2.1.2](#)) entspricht die Vorgehensweise der von Chapman et al. eingeordneten *local iterative approximate best-response* Methode, nach welcher die autonomen Instanzen wiederholt versuchen, das globale Ziel durch lokale Entscheidungen bestmöglich anzunähern [[Cha+11](#)]. In der Theorie der Multiagentensysteme ist dies als reaktive Koordinierung (engl. *reactive coordination* oder *eco-problem solving*) bekannt [[Fer99](#)].

Eine weitere Besonderheit des entwickelten Verfahrens ist die Trennung modellspezifischer Handlungsoptionen der beteiligten Energieanlagen von der eigentlichen Einsatzplanung. Während bestehende Ansätze (siehe [Abschnitt 1.3](#)) zumeist typspezifische Lastverschiebungsoptionen in eine Einsatzplanung integrieren (wie etwa die Berechnung von bestimmten Schaltzeitpunkten eines zweipunktgeregelten thermischen Gerätes) und daher für ihre jeweiligen Anwendungsfälle spezialisiert sind, wird im vorliegenden Fall zunächst ein typspezifisches Suchraummodell erstellt (nicht Fokus dieser Arbeit, vgl. [Abschnitt 6.4](#)), auf welchem die Einsatzplanung dann *typunabhängig* operieren kann.

Veröffentlichung (*Communication of Research*) Die *Design-Science Research Guidelines* schließen mit der Forderung nach Verbreitung der Forschungsergebnisse in der Fachwelt. Dies ist hier mittels der in [Abschnitt 1.6](#) beschriebenen eigenen Vorarbeiten bereits partiell geschehen wird mit der vorliegenden Arbeit vervollständigt.

8.2 ANKNÜPFENDER FORSCHUNGSBEDARF

Wie in obigem Abschnitt argumentiert, stellt die vorliegende Arbeit ein in sich geschlossenes Artefakt dar. Naturgemäß mussten dazu an einigen Stellen spezifische Voraussetzungen formuliert werden, deren Auflockerung oder Ersetzung erste Einstiegspunkte für anknüpfende Forschungsarbeiten darstellen können.

Verhalten der Agenten

Kooperation Die Heuristik COHDA fordert eine grundlegende Kooperationsbereitschaft der partizipierenden Akteure. Da die monetäre Vergütung im betrachteten Anwendungsfall der verbundbasierten Wirkleistungsbereitstellung durch die tatsächliche Erbringung des Wirkleistungsproduktes bestimmt ist und das Einbringen von

Flexibilität positiv in den Mehrwertverteilungsschlüssel innerhalb des Verbundes eingeht (vgl. [Abschnitt 4.4](#)), wodurch kooperatives Handeln belohnt wird, ist diese Annahme hier zulässig. Es ist allerdings auch durchaus denkbar, dass ein Akteur in mehreren Verbünden aktiv ist, davon aber nur in einem kooperativ handelt. Er kann damit die Chancen dieses Verbundes am Markt erhöhen, indem andere Verbünde durch bewusste Fehlinformationen zu Fehlplanungen verleitet werden. Derartige Angriffsszenarien werden etwa in [\[RUS13\]](#) untersucht. Möglichkeiten zur Behandlung der daraus resultierenden Unsicherheiten werden in z. B. in [\[And+13b\]](#) betrachtet. Diese sollten vor einer Umsetzung des hier vorgestellten Verfahrens bzgl. einer Integration geprüft werden.

Reflexion und Antizipation Auch in Bezug zu effizienz- und performanzsteigernden Maßnahmen besteht noch Optimierungspotenzial im Verhalten der Agenten. So wird in [\[Cha+11\]](#) festgestellt, dass die Performanz eines Verfahrens durch Ergänzung der von myopischen Agenten – welche in der vorliegenden Arbeit postuliert werden – durchgeführten rein reaktiven Handlungen mit Reflexion (Lernen aus vergangenen Erfahrungen) sowie Antizipation (vorausschauendes Verhalten) verbessert werden kann, ohne dabei die Effizienz des Verfahrens zu beeinträchtigen.

Gesinnungswahl Als Folgerung aus den obigen beiden Punkten sind die möglichen Ausprägungen der in dieser Arbeit modellierten freien Gesinnungswahl genauer zu betrachten. Wie können konkrete individuelle Zielsetzungen der Verbundteilnehmer aussehen, und welche Auswirkungen hat dies (ergänzend zu den in [Abschnitt 7.3.4](#) vereinfachend dargestellten Effekten) auf das Einsatzplanungsverfahren? Diese Frage wird insbesondere dann interessant, wenn Agenten wie oben beschrieben reflektiv/antizipatorisch handeln und zu individuellen Schlussfolgerungen in der Lage sind. Hier können dann bspw. lokale Risikoabschätzungen und die Bewertung von Gruppendynamiken zum Einsatz kommen.

Erweiterung der Heuristik

Hybride Zielfunktionen Die im Rahmen des betrachteten Anwendungsfalls verwendeten Zielfunktionen für das Optimierungsproblem beinhalteten neben der abstrakt formulierten freien Gesinnungswahl primär die möglichst exakte Abdeckung eines Wirkleistungsproduktes. Hinsichtlich einer nachfolgenden reaktiven Einsatzplanung zur Kompensation unerwarteter Ereignisse ist es jedoch zielführend, möglichst viele Freiheitsgrade der beteiligten Energieanlagen zu erhalten. Es stellt sich somit die Frage, wie ein solches sekundäres Optimierungsziel in das Verfahren integriert werden kann, da dies weder rein private noch rein globale Eigenschaften besitzt und daher nicht unmittelbar über die freie Gesinnungswahl und auch nicht über die globale Zielfunktion abgebildet werden kann. Hier muss ein Weg gefunden werden, derartige hybride Zielfunktionen in dem vorliegenden vollständig verteilten System zu realisieren.

Parallele Verbundmitgliedschaften Für die Entwicklung der Einsatzplanungsheuristik COHDA wurde ein zugrundeliegender Verbund als gegeben angenommen, und die daran beteiligten Akteure exklusiv als diesem Verbund zugehörig betrachtet. Es ist jedoch denkbar, dass Akteure in mehreren Verbünden zugleich aktiv sind. Hier ist zunächst zu unterscheiden, ob diese Verbünde sich bzgl. ihrer Planungshorizonte überschneiden oder nicht. In beiden Fällen ergeben sich zusätzliche Abhängigkeiten zwischen den eigentlich getrennt voneinander ausgeführten Einsatzplanungsprozessen der betreffenden Verbünde: Überschneiden sich die Planungshorizonte, so können die von den jeweiligen Einsatzplanungen erzeugten Lösungskandidaten nicht mehr unabhängig voneinander betrachtet werden, da für einen Verbundteilnehmer aufgrund der Verpflichtung zur potenziellen Einhaltung eines einmal publizierten Fahrplans keine zwei unterschiedlichen Fahrpläne in den einzelnen Einsatzplanungsprozessen existieren dürfen. Die Optimierungsprobleme müssten also gemeinsam behandelt werden. Überschneiden sich die Planungshorizonte nicht, können die Einsatzplanungsprozesse dennoch nicht vollständig unabhängig voneinander ausgeführt werden, da die Wahl eines Fahrplans eines Verbundteilnehmers für einen bestimmten Planungshorizont den resultierenden Handlungsspielraum aller darauf folgenden Zeiträume beeinflusst und umgekehrt. Wird also die Möglichkeit zur gleichzeitigen Mitgliedschaft in mehreren Verbünden gewünscht, so ergibt sich ein nochmals kom-

plexeres Einsatzplanungsproblem mit zusätzlichen Abhängigkeiten. Dies müsste genau analysiert werden, um weiterhin die Konfliktfreiheit und garantierte Konvergenz von COHDA gewährleisten zu können.

Adaptive Nachbarschaften Das in COHDA verwendete Overlay-Netz wird zu Beginn der Heuristik festgelegt und ist für die gesamte Dauer der Einsatzplanung statisch. Aus algorithmischer Sicht ist dies aber nicht erforderlich: Solange die Kommunikationstopologie zu jeder Zeit zusammenhängend ist, ist zur Laufzeit prinzipiell eine Modifikation dieser Topologie möglich. Hier ist zu untersuchen, inwieweit solche Modifikationen zur Steigerung der Effizienz beitragen können, indem bspw. Akteure mit besonders viel Freiheitsgraden dynamisch zur Laufzeit bzgl. der Topologie näher zueinander platziert werden.

Systemoffenheit Die Zusammensetzung eines Verbundes wurde für diese Arbeit als statisch angenommen. Bzgl. technischer Ausfälle von Akteuren (und damit von Knoten des Overlay-Netzes) ist diese Annahme zweckmäßig, da nach dem berühmten *FLP impossibility proof* nach Fischer, Lynch und Paterson in asynchronen Systemen mit Knotenausfällen keine verteilte Konsensbildung möglich ist [FLP85], und somit durch geeignete technische Maßnahmen geschützt werden muss (eine Möglichkeit hierzu wurde in [Abschnitt 4.4.1](#) bereits vorgeschlagen). Kontrollierte Modifikationen der Verbundzusammensetzung jedoch sollten zur Laufzeit möglich sein. Dies könnte etwa im Sinne des Autonomieerhalts auftreten, wenn sich zur Laufzeit der Einsatzplanung bspw. abzeichnet, dass keine hinreichend gute Lösung gefunden werden kann, und Akteure daher den Verbund vorzeitig verlassen möchten. Hier ist somit zu untersuchen, wie hinzukommende/wegfallende Akteure effizient in der Konsensbildung berücksichtigt werden können.

Vorzeitige Terminierung Die Evaluation des Laufzeitverhaltens der Heuristik zeigte, dass zu Beginn des Verfahrens eher starke Verbesserungen der Lösungsgüte erzielt werden, welche dann im weiteren Verlauf abflachen und eher langsam gegen den finalen Wert konvergieren (Abschnitte [7.2](#) und [7.6.1](#)). Insbesondere bei Verwendung des Stützvektor-basierten Suchraummodells σ^{SVM} (siehe [Abschnitt 7.6.3](#)) sollte hier im Sinne einer schnellen Entscheidung die Möglichkeit geschaffen werden, die Einsatzplanung vorzeitig zu beenden. Dies ist bspw. realisierbar, indem ein

Signal definiert wird, welches etwa vom Verbundsprecher ausgeht und alle Akteure dazu veranlasst, keine *neuen* Lösungskandidaten mehr zu produzieren. Gemäß der Betrachtungen aus [Abschnitt 5.4](#) würde das Verfahren dann durch Ausbreitung des zu diesem Zeitpunkt besten Lösungskandidaten nach spätestens $m \cdot (m - 1)$ Schritten terminieren, wenn m die Verbundgröße ist. Diese theoretische Überlegung ist durch eine geeignete Untersuchung zu bestätigen.

A

SYNTHETISCHE PROBLEMINSTANZEN

In [Abschnitt 7.1](#) wurde die Konstruktion von synthetischen Probleminstanzen für die Evaluation des in dieser Arbeit entwickelten Einsatzplanungsverfahrens vorgestellt. [Abbildung A.1](#) zeigt eine Auflistung in NumPy Notation¹ der in der Evaluation verwendeten Probleminstanz $P(10, 5, 5)$, deren Werte nach Konstruktionsvorschrift G-C(L)-D(S) aus [\[HLS10\]](#) erzeugt wurden. [Abbildung A.2](#) zeigt die in der Evaluation verwendeten Zielvektoren der Probleminstanzen für $h \in [1, 100]$. Für eine Simulation der Instanz $P(10, 5, 5)_{h=5}$ zeigt [Abbildung A.3](#) das Protokoll der Beobachterkomponente (siehe dazu [Abschnitt 7.2](#)). Als Ergänzung zur Evaluation des Topologieparameters in [Abschnitt 7.3.1](#) zeigt [Abbildung A.4](#) einen Vergleich der *Small-World* und der Gitter-Topologie für verschiedene Verbundgrößen m . Die Abbildungen [A.5](#) bis [A.7](#) visualisieren die Rohdaten, welche den Effizienzbetrachtungen in den Abschnitten [7.4.1](#) bis [7.4.3](#) zugrundeliegen.

¹<http://www.numpy.org/>

[[1, 1, 1, 1, 1],	[[60, 58, 57, 56, 56],
[4, 3, 3, 3, 3],	[62, 61, 60, 59, 58],
[6, 6, 6, 6, 6],	[65, 63, 62, 61, 61],
[9, 8, 8, 8, 8],	[67, 66, 65, 64, 63],
[11, 11, 11, 11, 11]],	[70, 68, 67, 66, 66]],
[[13, 12, 12, 12, 12],	[[71, 70, 68, 67, 67],
[15, 15, 15, 14, 14],	[74, 72, 71, 70, 69],
[18, 17, 17, 17, 17],	[76, 75, 73, 72, 72],
[20, 20, 20, 19, 19],	[79, 77, 76, 75, 74],
[23, 22, 22, 22, 22]],	[81, 80, 78, 77, 77]],
[[25, 24, 23, 23, 23],	[[83, 81, 80, 78, 78],
[27, 26, 26, 25, 25],	[85, 83, 82, 81, 80],
[30, 29, 28, 28, 28],	[88, 86, 85, 83, 83],
[32, 31, 31, 30, 30],	[90, 88, 87, 86, 85],
[35, 34, 33, 33, 33]],	[93, 91, 90, 88, 88]],
[[36, 35, 35, 34, 34],	[[95, 92, 91, 90, 89],
[39, 38, 37, 36, 36],	[97, 95, 93, 92, 91],
[41, 40, 40, 39, 39],	[100, 97, 96, 95, 94],
[44, 43, 42, 41, 41],	[102, 100, 98, 97, 96],
[46, 45, 45, 44, 44]],	[105, 102, 101, 100, 99]],
[[48, 47, 46, 45, 45],	[[106, 104, 102, 101, 100],
[50, 49, 48, 48, 47],	[109, 106, 105, 103, 102],
[53, 52, 51, 50, 50],	[111, 109, 107, 106, 105],
[55, 54, 53, 53, 52],	[114, 111, 110, 108, 107],
[58, 57, 56, 55, 55]],	[116, 114, 112, 111, 110]]]

Abbildung A.1: Auflistung der Gewichtsvektoren der Problemistanz $P(10, 5, 5)$, erzeugt nach Konstruktionsvorschrift G-C(L)-D(S) aus [\[HLS10\]](#).

h001 = [538, 524, 515, 507, 505]	h034 = [571, 557, 548, 540, 538]	h067 = [604, 590, 581, 573, 571]
h002 = [539, 525, 516, 508, 506]	h035 = [572, 558, 549, 541, 539]	h068 = [605, 591, 582, 574, 572]
h003 = [540, 526, 517, 509, 507]	h036 = [573, 559, 550, 542, 540]	h069 = [606, 592, 583, 575, 573]
h004 = [541, 527, 518, 510, 508]	h037 = [574, 560, 551, 543, 541]	h070 = [607, 593, 584, 576, 574]
h005 = [542, 528, 519, 511, 509]	h038 = [575, 561, 552, 544, 542]	h071 = [608, 594, 585, 577, 575]
h006 = [543, 529, 520, 512, 510]	h039 = [576, 562, 553, 545, 543]	h072 = [609, 595, 586, 578, 576]
h007 = [544, 530, 521, 513, 511]	h040 = [577, 563, 554, 546, 544]	h073 = [610, 596, 587, 579, 577]
h008 = [545, 531, 522, 514, 512]	h041 = [578, 564, 555, 547, 545]	h074 = [611, 597, 588, 580, 578]
h009 = [546, 532, 523, 515, 513]	h042 = [579, 565, 556, 548, 546]	h075 = [612, 598, 589, 581, 579]
h010 = [547, 533, 524, 516, 514]	h043 = [580, 566, 557, 549, 547]	h076 = [613, 599, 590, 582, 580]
h011 = [548, 534, 525, 517, 515]	h044 = [581, 567, 558, 550, 548]	h077 = [614, 600, 591, 583, 581]
h012 = [549, 535, 526, 518, 516]	h045 = [582, 568, 559, 551, 549]	h078 = [615, 601, 592, 584, 582]
h013 = [550, 536, 527, 519, 517]	h046 = [583, 569, 560, 552, 550]	h079 = [616, 602, 593, 585, 583]
h014 = [551, 537, 528, 520, 518]	h047 = [584, 570, 561, 553, 551]	h080 = [617, 603, 594, 586, 584]
h015 = [552, 538, 529, 521, 519]	h048 = [585, 571, 562, 554, 552]	h081 = [618, 604, 595, 587, 585]
h016 = [553, 539, 530, 522, 520]	h049 = [586, 572, 563, 555, 553]	h082 = [619, 605, 596, 588, 586]
h017 = [554, 540, 531, 523, 521]	h050 = [587, 573, 564, 556, 554]	h083 = [620, 606, 597, 589, 587]
h018 = [555, 541, 532, 524, 522]	h051 = [588, 574, 565, 557, 555]	h084 = [621, 607, 598, 590, 588]
h019 = [556, 542, 533, 525, 523]	h052 = [589, 575, 566, 558, 556]	h085 = [622, 608, 599, 591, 589]
h020 = [557, 543, 534, 526, 524]	h053 = [590, 576, 567, 559, 557]	h086 = [623, 609, 600, 592, 590]
h021 = [558, 544, 535, 527, 525]	h054 = [591, 577, 568, 560, 558]	h087 = [624, 610, 601, 593, 591]
h022 = [559, 545, 536, 528, 526]	h055 = [592, 578, 569, 561, 559]	h088 = [625, 611, 602, 594, 592]
h023 = [560, 546, 537, 529, 527]	h056 = [593, 579, 570, 562, 560]	h089 = [626, 612, 603, 595, 593]
h024 = [561, 547, 538, 530, 528]	h057 = [594, 580, 571, 563, 561]	h090 = [627, 613, 604, 596, 594]
h025 = [562, 548, 539, 531, 529]	h058 = [595, 581, 572, 564, 562]	h091 = [628, 614, 605, 597, 595]
h026 = [563, 549, 540, 532, 530]	h059 = [596, 582, 573, 565, 563]	h092 = [629, 615, 606, 598, 596]
h027 = [564, 550, 541, 533, 531]	h060 = [597, 583, 574, 566, 564]	h093 = [630, 616, 607, 599, 597]
h028 = [565, 551, 542, 534, 532]	h061 = [598, 584, 575, 567, 565]	h094 = [631, 617, 608, 600, 598]
h029 = [566, 552, 543, 535, 533]	h062 = [599, 585, 576, 568, 566]	h095 = [632, 618, 609, 601, 599]
h030 = [567, 553, 544, 536, 534]	h063 = [600, 586, 577, 569, 567]	h096 = [633, 619, 610, 602, 600]
h031 = [568, 554, 545, 537, 535]	h064 = [601, 587, 578, 570, 568]	h097 = [634, 620, 611, 603, 601]
h032 = [569, 555, 546, 538, 536]	h065 = [602, 588, 579, 571, 569]	h098 = [635, 621, 612, 604, 602]
h033 = [570, 556, 547, 539, 537]	h066 = [603, 589, 580, 572, 570]	h099 = [636, 622, 613, 605, 603]
		h100 = [637, 623, 614, 606, 604]

Abbildung A.2: Auflistung der Zielvektoren der Problem Instanz $P(10, 5, 5)$, erzeugt nach Konstruktionsvorschrift G-C(L)-D(S) aus [HLS10], für $h \in [1, 100]$.

```

[INFO ] sent msg | obj calls | time | Init: 2013-10-25_14:59:09
[INFO ]          0 |      0 |      | d_min=2.000000 [normalized 0.000000]
[INFO ]          0 |      0 |      | d_max=480.000000 [normalized 1.000000]
[INFO ]          0 |      0 |      | d_avg=237.220000 [normalized 0.492092]
[INFO ]          0 |      0 |      | Saving agent data to temporary files.
[INFO ]          0 |      0 |      | Creating 10 agents of type AgentMMMSSP
[INFO ]          0 |      0 |      | Connecting agents.
[INFO ]          0 |      0 |      | Notifying speaker (a9)
[INFO ]          0 |      2 |      | bkc-value | bkc-size | bkc-dist
[SOLUTION ] 0 |      3 | 0.0 | 1.000000 | 0.00 | 0.00
[SOLUTION ] 0 |      2 | 1.0 | 4.276151 | 0.10 | 0.10
[STATS ] 2 |      7 | 2.0 | 4.276151 | 0.10 | 0.10
[STATS ] 2 |     15 | 3.0 | 3.215481 | 0.20 | 0.10
[STATS ] 10 |     25 | 4.0 | 3.215481 | 0.20 | 0.10
[STATS ] 22 |     66 | 5.0 | 2.868201 | 0.40 | 0.10
[STATS ] 40 |    124 | 6.0 | 2.399582 | 0.50 | 0.10
[STATS ] 63 |    196 | 7.0 | 0.753138 | 0.70 | 0.10
[SOLUTION ] 82 |    266 | 8.0 | 0.895397 | 1.00 | 0.10
[SOLUTION ] 102 |    330 | 9.0 | 0.790795 | 1.00 | 0.10
[SOLUTION ] 123 |    395 | 10.0 | 0.686192 | 1.00 | 0.10
[SOLUTION ] 142 |    455 | 11.0 | 0.581590 | 1.00 | 0.10
[SOLUTION ] 167 |    522 | 12.0 | 0.581590 | 1.00 | 0.40
[SOLUTION ] 193 |    595 | 13.0 | 0.267782 | 1.00 | 0.10
[SOLUTION ] 221 |    672 | 14.0 | 0.190377 | 1.00 | 0.10
[SOLUTION ] 249 |    751 | 15.0 | 0.006276 | 1.00 | 0.10
[SOLUTION ] 277 |    829 | 16.0 | 0.006276 | 1.00 | 0.60
[SOLUTION ] 305 |    909 | 17.0 | 0.006276 | 1.00 | 0.90
[SOLUTION ] 333 |    988 | 18.0 | 0.006276 | 1.00 | 1.00
[STATS ] 348 |   1046 | 19.0 | 0.006276 | 1.00 | 1.00
[STATS ] 368 |   1096 | 20.0 | 0.006276 | 1.00 | 1.00
[STATS ] 387 |   1146 | 21.0 | 0.006276 | 1.00 | 1.00
[STATS ] 398 |   1185 | 22.0 | 0.006276 | 1.00 | 1.00
[STATS ] 398 |   1196 | 23.0 | 0.006276 | 1.00 | 1.00
[INFO ] 398 |   1196 | 23.0 | Nodes inactive, stopping now.
[INFO ] 398 |   1196 | Target | [542 528 519 511 509]
[INFO ] 398 |   1196 | Result | [543 529 520 512 510]
[INFO ] 398 |   1196 | End | 2013-10-25_14:59:34

```

Abbildung A.3: Protokoll einer Simulation der Problem Instanz $P(10, 5, 5)_{h=5}$.

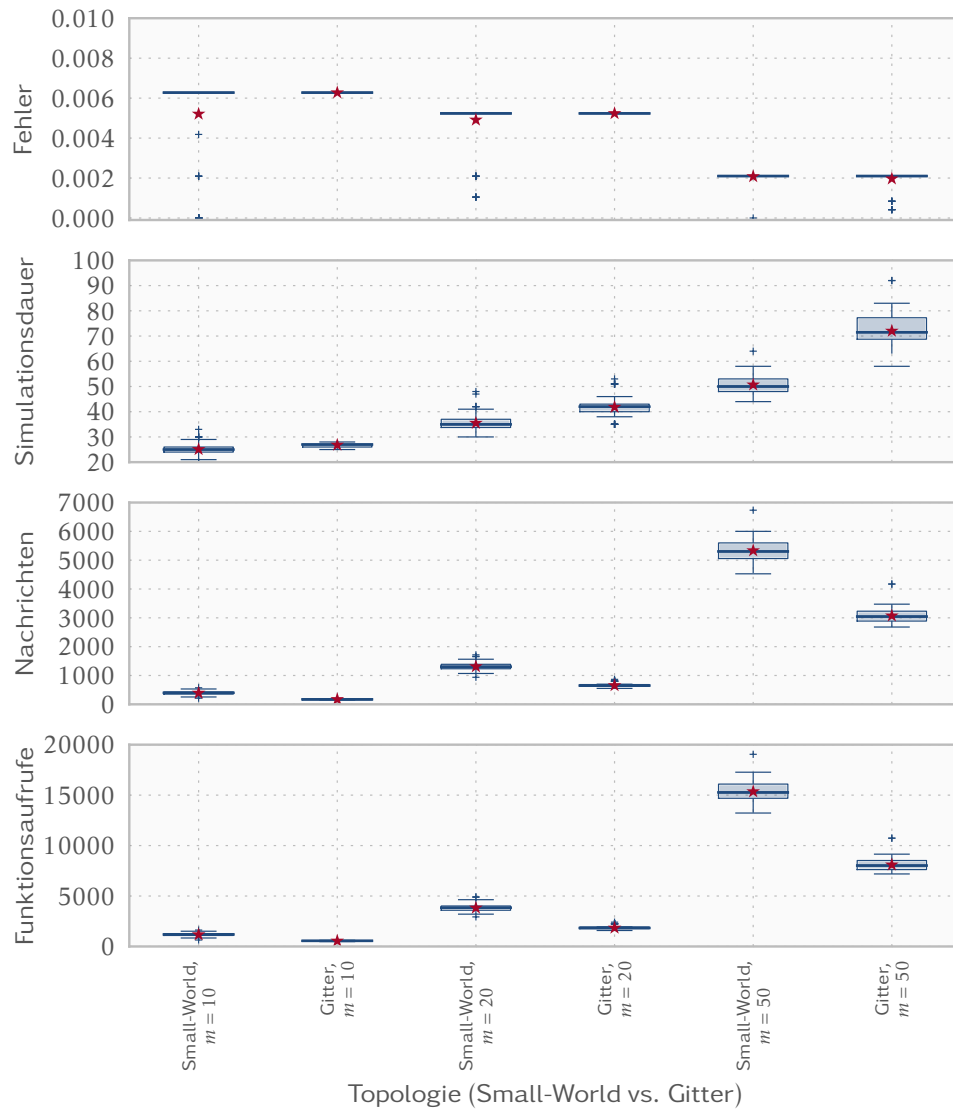


Abbildung A.4: Verbleibender Fehler, Simulationsdauer, Kommunikationsaufwand und Rechenaufwand im Vergleich zwischen *Small-World* und Gitter-Topologie für verschiedene Verbundgrößen m . Darstellung jeweils als Boxplot sowie Mittelwert (Sternsymbol) für 100 Ausführungen je Wert.

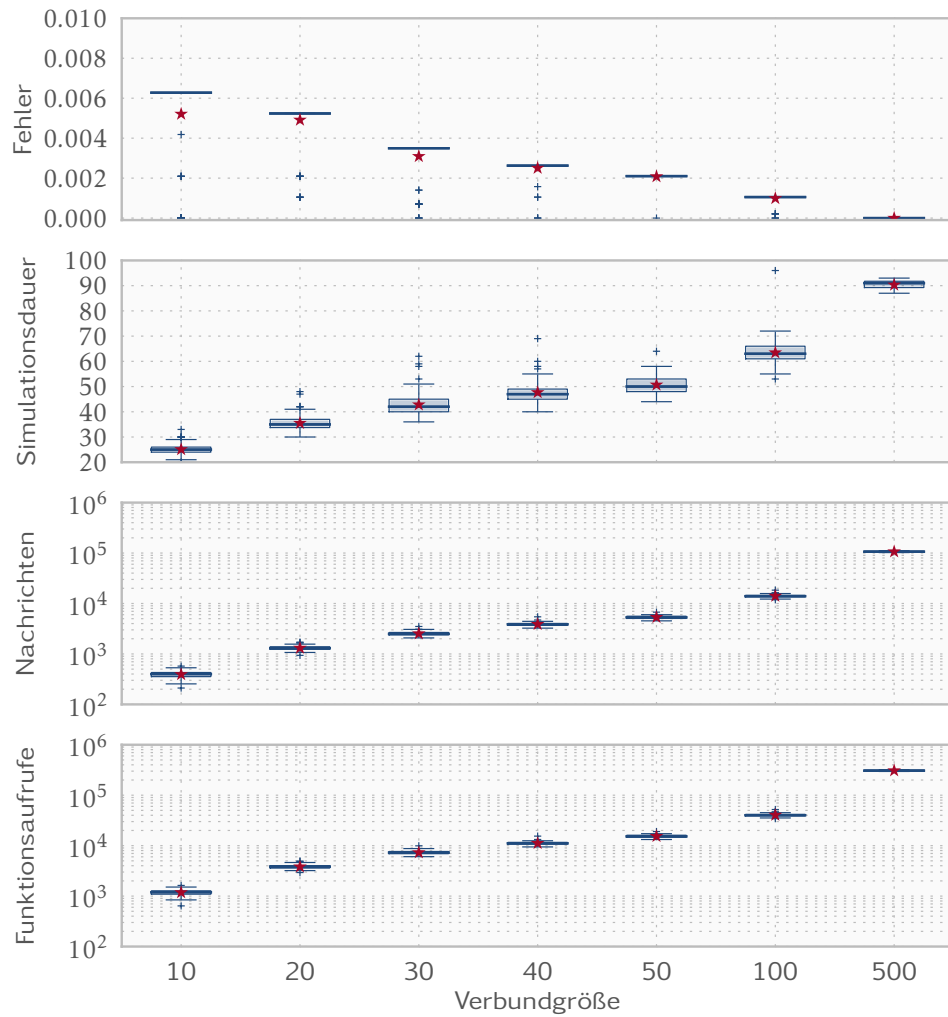


Abbildung A.5: Verbleibender Fehler, Simulationsdauer, Kommunikationsaufwand und Rechenaufwand bei Simulation verschiedener Verbundgrößen m . Darstellung jeweils als Boxplot sowie Mittelwert (Sternsymbol) für 100 Ausführungen je Wert.

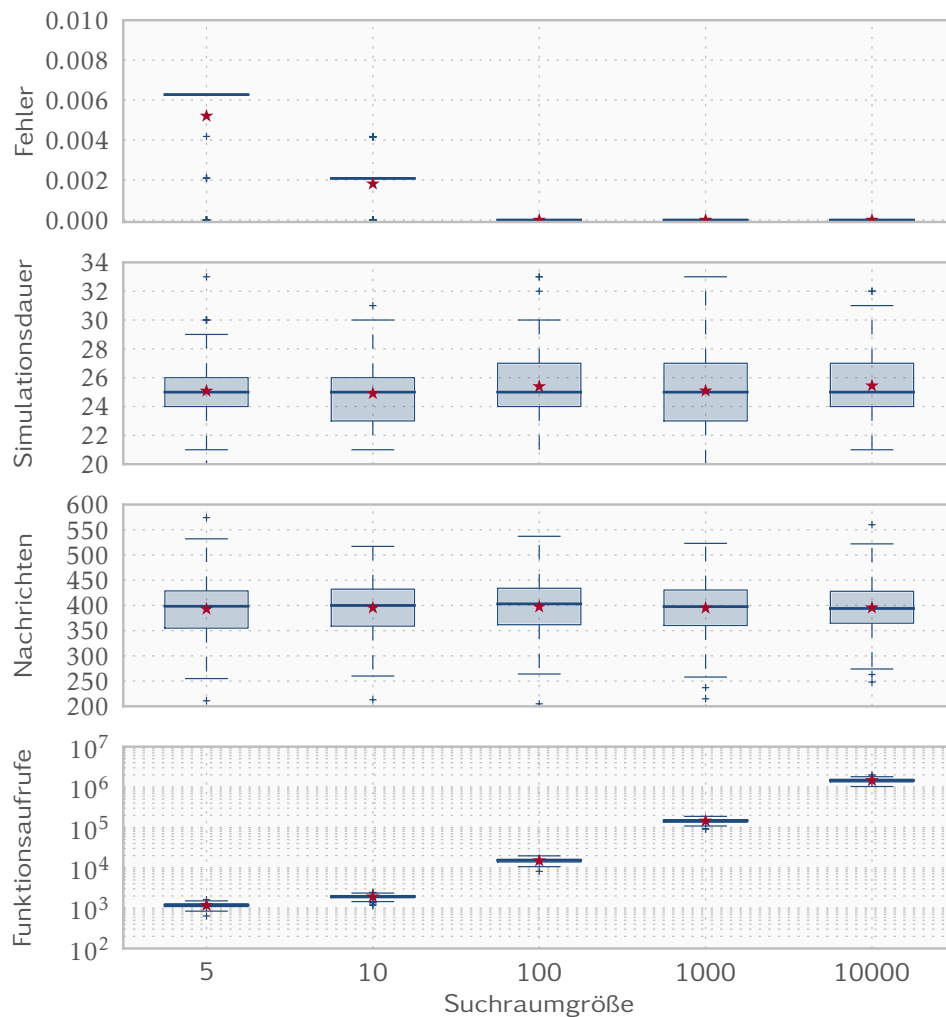


Abbildung A.6: Verbleibender Fehler, Simulationsdauer, Kommunikationsaufwand und Rechenaufwand bei Simulation verschiedener Suchraumgrößen n . Darstellung jeweils als Boxplot sowie Mittelwert (Sternsymbol) für 100 Ausführungen je Wert.

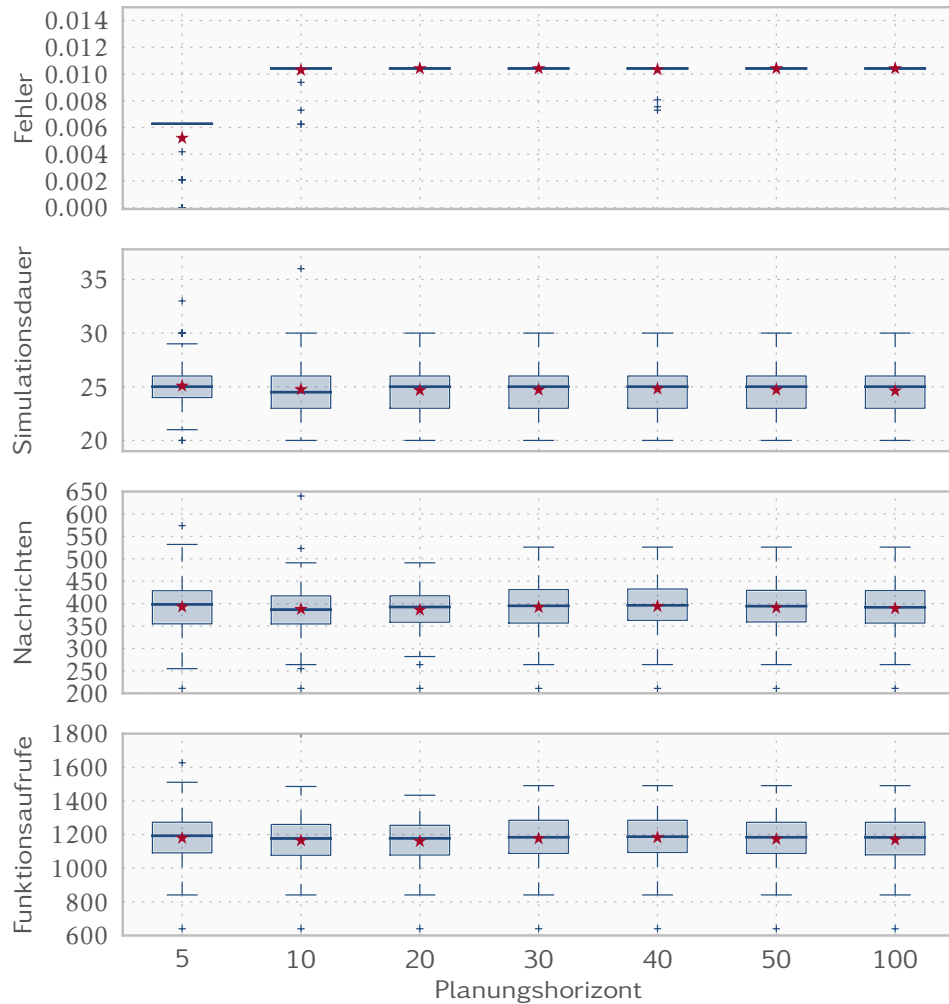


Abbildung A.7: Verbleibender Fehler, Simulationsdauer, Kommunikationsaufwand und Rechenaufwand bei Simulation verschiedener Planungshorizonte q . Darstellung jeweils als Boxplot sowie Mittelwert (Sternsymbol) für 100 Ausführungen je Wert.

ABBILDUNGSVERZEICHNIS

1.1	Topologieebenen im Forschungsverbund Smart Nord	7
1.2	Teilprojekt 1 im Forschungsverbund Smart Nord	9
2.1	Constraint-Graph	26
3.1	Designprozess eines selbstorganisierenden Systems	37
3.2	Prädiktive Einsatzplanung	39
3.3	Overlay-Netz	57
3.4	Informationssynthese	60
3.5	Reaktive Einsatzplanung	73
4.1	Heuristik: COHDA	80
6.1	Prozessmodell	116
6.2	Suchraummodell	120
7.1	$P(10, 5, 5)$ (Gewichtsvektoren, Zielvektoren)	131
7.2	$P(10, 5, 5)$ (Häufigkeitsdichte der Fehlerwerte)	134
7.3	$P(10, 5, 5)$ (<i>Random Walk</i>)	137
7.4	$P(10, 5, 5)$ (COHDA)	139
7.5	$P(10, 5, 5)_{h=5}$ (Laufzeitverhalten)	141
7.6	$P(10, 5, 5)_{h=5}$ (Ergebnis)	143
7.7	Topologie (Performanz, Effizienz)	144
7.8	Vernetzungsgrad (Performanz, Effizienz)	146
7.9	Nachrichtenverzögerung (Performanz, Effizienz)	148
7.10	Reaktionszeit (Performanz, Effizienz)	150
7.11	Statische Gesinnungswahl (Performanz, Effizienz)	153
7.12	Dynamische Gesinnungswahl (Performanz, Effizienz)	154
7.13	Dynamische Gesinnungswahl (Performanz Detailansicht)	155

7.14 Verbundgröße (Effizienzkennzahlen)	160
7.15 Suchraumgröße (Effizienzkennzahlen)	162
7.16 Planungshorizont (Effizienzkennzahlen)	163
7.17 Regression: Vernetzungsgrad	169
7.18 Regression: Nachrichtenverzögerung	170
7.19 Regression: Reaktionszeit	172
7.20 Regression: Verbundgröße	173
7.21 Regression: Suchraumgröße	175
7.22 30 BHKW (Laufzeitverhalten)	178
7.23 30 BHKW (Ergebnis)	178
7.24 30 BHKW	180
7.25 Regression: Verbundgröße (BHKW)	181
7.26 Suchraummodell (Performanz, Effizienz)	183
7.27 Suchraummodell (Effizienzkennzahlen)	185
7.28 EPEX SPOT (Handelstag)	189
7.29 EPEX SPOT V_{10}^+	193
7.30 EPEX SPOT V_{111}^-	194
7.31 EPEX SPOT $V_{123}^\pm$	196
7.32 Spreizungsreduktion V_{12}^+	200
7.33 Spreizungsreduktion V_{111}^-	202
7.34 Spreizungsreduktion $V_{123}^\pm$	203
7.35 <i>Stigspace</i> (30 BHKW)	207
A.1 $P(10, 5, 5)$ (Gewichtsvektoren)	222
A.2 $P(10, 5, 5)$ (Zielvektoren)	223
A.3 $P(10, 5, 5)_{h=5}$ (Simulationsprotokoll)	224
A.4 <i>Small-World</i> vs. Gitter (Performanz, Effizienz)	225
A.5 Verbundgröße (Performanz, Effizienz)	226
A.6 Suchraumgröße (Performanz, Effizienz)	227
A.7 Planungshorizont (Performanz, Effizienz)	228

LITERATUR

- [AA13] Eric Ahlers und Yvonne Aniol. *Realistische Schritte zur Umsetzung von Smart Grids in Deutschland*. Techn. Ber. Bundesverband der Energie- und Wasserwirtschaft (BDEW), 2013 (siehe S. 2)
- [Aga+11] Jérôme Agater u. a. „eCarUs – A Tool for Optimal Placement of Stations for Electric Vehicle Batteries“. In: *25th International Conference on Informatics for Environmental Protection – Innovation in Sharing Environmental Observations and Information*. Hrsg. von Werner Pillmann, Sven Schade und Paul Smits. Environmental Informatics Prize for Students. Shaker Verlag, 2011 (siehe S. 21)
- [AKM12] Hans-Jürgen Appelrath, Henning Kagermann und Christoph Mayer, Hrsg. *Future Energy Grid. Migrationspfade ins Internet der Energie*. aca-tech STUDIE. Springer, 2012. DOI: [10.1007/978-3-642-27864-8](https://doi.org/10.1007/978-3-642-27864-8) (siehe S. 4)
- [AL10] Layla AlAbdulkarim und Zofia Lukszo. „Integrating information security requirements in critical infrastructures: smart metering case“. In: *International Journal of Critical Infrastructures* 6.2 (2010), S. 187–209. DOI: [10.1504/IJCIS.2010.031074](https://doi.org/10.1504/IJCIS.2010.031074) (siehe S. 5)
- [And+10] Gerrit Anders u. a. „Structuring and Controlling Distributed Power Sources by Autonomous Virtual Power Plants“. In: *IEEE Power and Energy Student Summit (PESS 2010)*. IEEE Power & Energy Society, 2010 (siehe S. 15)
- [And+12] Gerrit Anders u. a. „On the Influence of Inter-Agent Variation on Multi-Agent Algorithms Solving a Dynamic Task Allocation Problem under Uncertainty“. In: *Sixth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2012)*. IEEE Computer Society,

- 2012, S. 29–38. DOI: [10.1109/SASO.2012.16](https://doi.org/10.1109/SASO.2012.16) (siehe S. [20](#), [34](#), [149](#), [197](#))
- [And+13a] Gerrit Anders u. a. „A Trust- and Cooperation-Based Solution of a Dynamic Resource Allocation Problem“. In: *7th International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2013)*. IEEE Computer Society, 2013. DOI: [10.1109/SASO.2013.33](https://doi.org/10.1109/SASO.2013.33) (siehe S. [15](#))
- [And+13b] Gerrit Anders u. a. „Trust-Based Scenarios – Predicting Future Agent Behavior in Open Self-Organizing Systems“. In: *Proc. of the 7th IFIP Int'l Workshop on Self-Organizing Systems (IWSOS 2013)*. Lecture Notes in Computer Science (LNCS). Springer, 2013. Im Druck (siehe S. [216](#))
- [AO83] Krzysztof R. Apt und Ernst-Rüdiger Olderog. „Proof rules and transformations dealing with fairness“. In: *Science of Computer Programming* 3.1 (1983), S. 65–100. DOI: [10.1016/0167-6423\(83\)90004-7](https://doi.org/10.1016/0167-6423(83)90004-7) (siehe S. [90](#))
- [Ash47] William Ross Ashby. „The Nervous System as Physical Machine: With Special Reference to the Origin of Adaptive Behavior“. In: *Mind* 56.221 (1947), S. 44–59 (siehe S. [72](#))
- [Ash62] William Ross Ashby. „Principles of the self-organizing system“. In: *Principles of Self-organization* 37.2 (1962). Nachdruck in: *Emergence: Complexity & Organization (E:CO)* 6.1–2 (2004). DOI: [10.1080/00221309.1947.9918144](https://doi.org/10.1080/00221309.1947.9918144) (siehe S. [31](#))
- [AW04] Hagit Attiya und Jennifer Welch. *Distributed Computing: Fundamentals, Simulations, and Advanced Topics*. 2. Aufl. Bd. 19. Wiley Series on Parallel and Distributed Computing. John Wiley & Sons, 2004 (siehe S. [30](#), [94](#))
- [BD10] Deutsche Bundesregierung (Bundeskabinett). *Energiekonzept für eine umweltschonende, zuverlässige und bezahlbare Energieversorgung*. 28. Sep. 2010 (siehe S. [1](#))
- [BD12] Deutsche Bundesregierung (Bundeskabinett). *Gesetz für den Vorrang Erneuerbarer Energien. Erneuerbare-Energien-Gesetz vom 25. Oktober 2008 (BGBl. I S. 2074), das zuletzt durch Artikel 5 des Gesetzes vom 20.*

Dezember 2012 (BGBl. I S. 2730) geändert worden ist. Gesetz. 20. Dez. 2012 (siehe S. 2)

- [BD13] Deutsche Bundesregierung (Bundeskabinett). *Gesetz für die Erhaltung, die Modernisierung und den Ausbau der Kraft-Wärme-Kopplung. Kraft-Wärme-Kopplungsgesetz vom 19. März 2002 (BGBl. I S. 1092), das durch Artikel 4 Absatz 77 des Gesetzes vom 7. August 2013 (BGBl. I S. 3154) geändert worden ist. Gesetz. 7. Aug. 2013 (siehe S. 2)*
- [Bec+13] Andreas Beck u. a. „Energiesysteme und das Paradigma des Agenten“. In: *Agentensysteme in der Automatisierungstechnik*. Hrsg. von Peter Göhner. Xpert.press. Springer, 2013. Kap. 2, S. 21–42. DOI: [10.1007/978-3-642-31768-2](https://doi.org/10.1007/978-3-642-31768-2) (siehe S. 6, 30)
- [BK10] K. El Bakari und W. L. Kling. „Virtual Power Plants: An Answer to Increasing Distributed Generation“. In: *IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*. IEEE Power & Energy Society, 2010, S. 1–6. DOI: [10.1109/ISGTEUROPE.2010.5638984](https://doi.org/10.1109/ISGTEUROPE.2010.5638984) (siehe S. 4, 5)
- [Bla+13] Marita Blank u. a. *Evaluationsszenarien für die Teilprojekte 1 – 4*. Techn. Ber. Version 23.10.2013. Forschungsverbund Smart Nord, 2013. URL: <http://smarnord.de/publikationen> (siehe S. 186, 190, 198)
- [BMU10] Bundesumweltministerium (BMU). *Nationaler Aktionsplan für erneuerbare Energie gemäß der Richtlinie 2009/28/EG zur Förderung der Nutzung von Energie aus erneuerbaren Quellen*. 3. Aug. 2010 (siehe S. 1)
- [Bor01] Francesca Jane Born. „Aiding Renewable Energy Integration through Complimentary Demand-Supply Matching“. Diss. University of Strathclyde, 2001 (siehe S. 46)
- [BR03] Christian Blum und Andrea Roli. „Metaheuristics in combinatorial optimization“. In: *ACM Computing Surveys* 35.3 (2003), S. 268–308. DOI: [10.1145/937503.937505](https://doi.org/10.1145/937503.937505) (siehe S. 214)
- [BRS10] Jörg Bremer, Barbara Rapp und Michael Sonnenschein. „Support Vector based Encoding of Distributed Energy Resources’ Feasible Load Spaces“. In: *IEEE PES Innovative Smart Grid Technologies Conference*

- Europe (ISGT Europe)*. IEEE Power & Energy Society, 2010. DOI: [10.1109/ISGTEUROPE.2010.5638940](#) (siehe S. [44](#), [119](#))
- [BRS11a] Jörg Bremer, Barbara Rapp und Michael Sonnenschein. „Encoding Distributed Search Spaces for Virtual Power Plants“. In: *IEEE Symposium on Computational Intelligence Applications In Smart Grid (CIASG)*. 2011. DOI: [10.1109/CIASG.2011.5953329](#) (siehe S. [40](#), [119](#))
- [BRS11b] Jörg Bremer, Barbara Rapp und Michael Sonnenschein. „Including Environmental Performance Indicators into Kernel based Search Space Representations“. In: *Information Technologies in Environmental Engineering*. Hrsg. von Paulina Golinska, Marek Fertsch und Jorge Marx-Gómez. Environmental Science and Engineering. Springer, 2011. DOI: [10.1007/978-3-642-19536-5_22](#) (siehe S. [119](#))
- [BS12] Jörg Bremer und Michael Sonnenschein. „A Distributed Greedy Algorithm for Constraint-based Scheduling of Energy Resources“. In: *SEN-MAS'2012 Workshop – Proc. of the Federated Conference on Computer Science and Information Systems (FedCSIS 2012)*. Wrocław, Poland, 2012, S. 1285–1292 (siehe S. [10](#), [26](#), [204](#), [206](#))
- [BS13] Jörg Bremer und Michael Sonnenschein. „Constraint-Handling for Optimization with Support Vector Surrogate Models: A Novel Decoder Approach“. In: *International Conference on Agents and Artificial Intelligence (ICAART 2013)*. Hrsg. von Joaquim Filipe und Ana L. N. Fred. Bd. 2. INSTICC Press, 2013. DOI: [10.5220/0004241100910100](#) (siehe S. [120](#))
- [BSA11] Sebastian Beer, Michael Sonnenschein und Hans-Jürgen Appelrath. „Towards a Self-Organization Mechanism for Agent Associations in Electricity Spot Markets“. In: *Informatik 2011 – Proceedings der 41. GI-Jahrestagung*. Hrsg. von Hans-Ulrich Heiß u. a. GI-Edition – Lecture Notes in Informatics (LNI) P-191. Bonner Köllen Verlag, 2011 (siehe S. [79](#), [89](#))
- [Cam+03] Scott Camazine u. a. *Self-Organization in Biological Systems*. 2. Aufl. Princeton University Press, 2003 (siehe S. [33](#), [55](#))

- [Cha+11] Archie C. Chapman u. a. „A unifying framework for iterative approximate best-response algorithms for distributed constraint optimization problems“. In: *The Knowledge Engineering Review* 26.04 (2011), S. 411–444. DOI: [10.1017/S0269888911000178](https://doi.org/10.1017/S0269888911000178) (siehe S. [27](#), [55](#), [59](#), [205](#), [215](#), [216](#))
- [Che+09] Betty H. C. Cheng u. a. „Software Engineering for Self-Adaptive Systems: A Research Roadmap“. In: *Software Engineering for Self-Adaptive Systems*. Hrsg. von Betty H. C. Cheng u. a. Bd. 5525. Lecture Notes in Computer Science. Springer, 2009, S. 1–26. DOI: [10.1007/978-3-642-02161-9_1](https://doi.org/10.1007/978-3-642-02161-9_1) (siehe S. [36](#))
- [Cra04] Valentin Crastan. *Elektrische Energieversorgung* 2. Springer, 2004. DOI: [10.1007/978-3-662-06958-5](https://doi.org/10.1007/978-3-662-06958-5) (siehe S. [8](#), [23](#), [41](#))
- [CRW11] Adam Campbell, Cortney Riggs und Annie S. Wu. „On the impact of variation on self-organizing systems“. In: *Fifth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2011)*. IEEE Computer Society, 2011, S. 119–128. DOI: [10.1109/SASO.2011.23](https://doi.org/10.1109/SASO.2011.23) (siehe S. [33](#))
- [DMW10] Reiner Dumke, Steffen Mencke und Cornelius Wille. *Quality Assurance of Agent-Based and Self-Managed Systems*. CRC Press – An Imprint of Taylor & Francis Group, 2010 (siehe S. [30](#), [40](#))
- [DO03] Diego Doval und Donal O’Mahony. „Overlay Networks: A Scalable Alternative for P2P“. In: *IEEE Internet Computing* 7.4 (2003), S. 79–82. DOI: [10.1109/MIC.2003.1215663](https://doi.org/10.1109/MIC.2003.1215663) (siehe S. [57](#))
- [Dol00] Shlomi Dolev. *Self-Stabilization*. MIT Press, 2000 (siehe S. [93](#), [94](#), [102](#))
- [EG00] Matthias Ehrgott und Xavier Gandibleux. „A survey and annotated bibliography of multiobjective combinatorial optimization“. In: *OR Spectrum* 22.4 (2000), S. 425–460. DOI: [10.1007/s002910000046](https://doi.org/10.1007/s002910000046) (siehe S. [44](#), [67](#)–[69](#))
- [Elm+09] Wilfried Elmenreich u. a. „A Survey of Models and Design Methods for Self-organizing Networked Systems“. In: *Proc. of the 4th IFIP Int’l Workshop on Self-Organizing Systems (IWSOS 2009)*. Bd. 5918. Lecture

- Notes in Computer Science (LNCS). Springer, 2009, S. 37–49. DOI: [10.1007/978-3-642-10865-5_4](https://doi.org/10.1007/978-3-642-10865-5_4) (siehe S. [36](#))
- [ETG12] ETG-Task Force Demand Side Management. *Demand Side Integration - Lastverschiebungspotenziale in Deutschland*. Techn. Ber. Frankfurt am Main, Germany: Energietechnische Gesellschaft im VDE (ETG), 2012 (siehe S. [1](#), [2](#))
- [EU09] Europäisches Parlament und Rat der Europäischen Union. *Richtlinie 2009/28/EG des Europäischen Parlaments und des Rates vom 23. April 2009 zur Förderung der Nutzung von Energie aus erneuerbaren Quellen und zur Änderung und anschließenden Aufhebung der Richtlinien 2001/77/EG und 2003/30/EG*. Richtlinie. 23. Apr. 2009 (siehe S. [1](#))
- [Eys+08] Moritz Eysholdt u. a. „A Tool for Modeling and Optimization of Residential Electricity Consumption“. In: *22th International Conference on Informatics for Environmental Protection – Environmental Informatics and Industrial Ecology*. Hrsg. von A. Möller, B. Page und M. Schreiber. Environmental Informatics Prize for Students. Shaker Verlag, 2008, S. 318–326 (siehe S. [21](#))
- [FD04] Friedrich Forssman und Ralf De Jong. *Detailtypografie. Nachschlagewerk für alle Fragen zu Schrift und Satz*. 3. Aufl. Schmidt, 2004 (siehe S. [III](#))
- [Fer99] Jacques Ferber. *Multi-Agent Systems: Introduction to Distributed Artificial Intelligence*. Addison-Wesley – An Imprint of Pearson Education, 1999 (siehe S. [47](#), [54](#), [55](#), [215](#))
- [FLP85] Michael J. Fischer, Nancy Ann Lynch und Michael S. Paterson. „Impossibility of distributed consensus with one faulty process“. In: *Journal of the ACM* 32.2 (1985), S. 374–382. DOI: [10.1145/3149.214121](https://doi.org/10.1145/3149.214121) (siehe S. [85](#), [218](#))
- [Fra86] Nissim Francez. *Fairness*. Texts and Monographs in Computer Science. Springer, 1986. DOI: [10.1007/978-1-4612-4886-6](https://doi.org/10.1007/978-1-4612-4886-6) (siehe S. [90](#))
- [Ger07] Carlos Gershenson. *Design and Control of Self-organizing Systems*. Hrsg. von Octavio Miramontes. 1. Aufl. CopIt ArXives, 2007 (siehe S. [31–33](#), [35](#), [72](#))

- [HBS13] Christian Hinrichs, Jörg Bremer und Michael Sonnenschein. „Distributed Hybrid Constraint Handling in Large Scale Virtual Power Plants“. In: *IEEE PES Conference on Innovative Smart Grid Technologies Europe (ISGT Europe 2013)*. IEEE Power & Energy Society, 2013 (siehe S. [20](#), [25](#), [121](#), [164](#), [182](#))
- [HDS10] Klaus Heuck, Klaus-Dieter Dettmann und Detlef Schulz. *Elektrische Energieversorgung. Erzeugung, Übertragung und Verteilung elektrischer Energie für Studium und Praxis*. Vieweg+Teubner, 2010. DOI: [10.1007/978-3-8348-9761-9](#) (siehe S. [3](#))
- [Hev+04] Alan R. Hevner u. a. „Design Science in Information Systems Research“. In: *MIS Quarterly* 28.1 (2004), S. 75–105 (siehe S. [209](#), [210](#))
- [Hey01] Francis Heylighen. „The Science of Self-Organization and Adaptivity“. In: *The Encyclopedia of Life Support Systems* 5.3 (2001). Hrsg. von L. Douglas Kiel, S. 253–280 (siehe S. [31](#), [72](#))
- [Hin+12] Christian Hinrichs u. a. „Ergebnisse eines Feldtests zum oszillationsfreien Lastmanagement von Kühl-/Gefriergeräten in Privathaushalten auf Basis von Temperaturvorgaben“. In: *VDE-Kongress 2012 – Intelligente Energieversorgung der Zukunft*. 2012 (siehe S. [21](#))
- [Hin08] Christian Hinrichs. „Untersuchung und Entwicklung von Verfahren zur Desynchronisation adaptiver Kühlgeräte“. Masterarbeit. Carl von Ossietzky Universität Oldenburg, 2008. URN: [urn:nbn:de:gbv:715-oops-14054](#) (siehe S. [21](#), [197](#))
- [Hin11] Christian Hinrichs. „Selbstorganisierte Koordinationsverfahren für ein dezentrales Supply-Demand Matching im elektrischen Verteilnetz“. In: *Energieinformatik 2011 – Tagungsband zum 2. Doktorandenworkshop des GI Arbeitskreises Energieinformationssysteme (GI AK EINS)*. Hrsg. von H.-J. Appelrath u. a. Springer, 2011, S. 15–26 (siehe S. [20](#))
- [HLS] Christian Hinrichs, Sebastian Lehnhoff und Michael Sonnenschein. „Paving the Royal Road for Complex Systems: On the Influence of Memory on Adaptivity“. In: *International Symposium Selforganization in Complex Systems: The Past, Present, and Future of Synergetics*. Hrsg. von Axel Pelster und Günther Wunner. Springer. Im Druck (siehe S. [20](#))

- [HLS10] Bing Han, Jimmy Leblet und Gwendal Simon. „Hard Multidimensional Multiple Choice Knapsack Problems, an Empirical Study“. In: *Computers & Operations Research* 37.1 (2010), S. 172–181. DOI: [10.1016/j.cor.2009.04.006](https://doi.org/10.1016/j.cor.2009.04.006) (siehe S. [128](#), [129](#), [221–223](#))
- [HLS14a] Christian Hinrichs, Sebastian Lehnhoff und Michael Sonnenschein. „A Decentralized Heuristic for Multiple-Choice Combinatorial Optimization Problems“. In: *Operations Research Proceedings 2012. Selected Papers of the International Annual Conference of the German Operations Research Society (GOR)*. Springer, 2014. Kap. XIII, S. 297–302. DOI: [10.1007/978-3-319-00795-3_43](https://doi.org/10.1007/978-3-319-00795-3_43) (siehe S. [20](#), [46](#))
- [HLS14b] Christian Hinrichs, Sebastian Lehnhoff und Michael Sonnenschein. „CO-HDA: A Combinatorial Optimization Heuristic for Distributed Agents“. In: *Agents and Artificial Intelligence. 5th International Conference, ICAART 2013, Barcelona, Spain, February 15-18, 2013. Revised Selected Papers*. Hrsg. von Joaquim Filipe und Ana Fred. Bd. 449. Communications in Computer and Information Science. Springer, 2014 (siehe S. [20](#), [67](#))
- [HMS04] M. Hifi, M. Michrafy und A. Sbihi. „Heuristic algorithms for the multiple-choice multidimensional knapsack problem“. In: *Journal of the Operational Research Society* 55.12 (2004), S. 1323–1332. DOI: [10.1057/palgrave.jors.2601796](https://doi.org/10.1057/palgrave.jors.2601796) (siehe S. [128](#))
- [Hou10] Michiel Houwing. „Smart Heat and Power – Utilizing the Flexibility of Micro Cogeneration“. Diss. Technische Universiteit Delft, 2010 (siehe S. [27](#), [117](#))
- [HRY91] Rodney R. Howell, Louis E. Rosier und Hsu-Chun Yen. „A taxonomy of fairness and temporal logic problems for Petri nets“. In: *Theoretical Computer Science* 82.2 (1991), S. 341–372. DOI: [10.1016/0304-3975\(91\)90228-T](https://doi.org/10.1016/0304-3975(91)90228-T) (siehe S. [90](#))
- [HSL13] Christian Hinrichs, Michael Sonnenschein und Sebastian Lehnhoff. „Evaluation of a Self-Organizing Heuristic for Interdependent Distributed Search Spaces“. In: *International Conference on Agents and Artificial Intelligence (ICAART 2013)*. Hrsg. von Joaquim Filipe und Ana Fred.

- Bd. 1. INSTICC Press, 2013. DOI: [10.5220/0004227000250034](https://doi.org/10.5220/0004227000250034) (siehe S. 20)
- [HVS09] Christian Hinrichs, Ute Vogel und Michael Sonnenschein. „Modelling and Evaluation of Desynchronization Strategies for Controllable Cooling Devices“. In: *6th Vienna International Conference on Mathematical Modelling (Mathmod 2009)*. Hrsg. von Inge Troch und Felix Breitenecker. 2009 (siehe S. 21, 197)
- [HVS11] Christian Hinrichs, Ute Vogel und Michael Sonnenschein. *Approaching Decentralized Demand Side Management via Self-Organizing Agents*. Hrsg. von Alex Rogers, Keith Decker und Koen Kok. 2nd International Workshop on Agent Technologies for Energy Systems (ATES 2011), 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011). 2011 (siehe S. 20)
- [Ili07] Marija D. Ilić. „From Hierarchical to Open Access Electric Power Systems“. In: *Proceedings of the IEEE* 95.5 (2007), S. 1060–1084. DOI: [10.1109/JPROC.2007.894711](https://doi.org/10.1109/JPROC.2007.894711) (siehe S. 3)
- [JV01] Ulrike Jordan und Klaus Vajen. „Influence Of The DHW Load Profile On The Fractional Energy Savings: A Case Study Of A Solar Combi-System With TRNSYS Simulations“. In: *Solar Energy* 69 (2001), S. 197–208. DOI: [10.1016/S0038-092X\(00\)00154-7](https://doi.org/10.1016/S0038-092X(00)00154-7) (siehe S. 177)
- [Kad11] Elsy Kaddoum. „Optimization under Constraints of Distributed Complex Problems using Cooperative Self-Organization“. Diss. Université Paul Sabatier Toulouse, 2011 (siehe S. 5)
- [Kam10] Andreas Kamper. „Dezentrales Lastmanagement zum Ausgleich kurzfristiger Abweichungen im Stromnetz“. Diss. Karlsruher Institut für Technologie, 2010 (siehe S. 13, 14)
- [KAS10] Stephan Kohler, Annegret-Claudine Agricola und Hannes Seidl. *dena-Netzstudie II*. Techn. Ber. Berlin, Germany: Deutsche Energie-Agentur GmbH (dena), 2010 (siehe S. 2)

- [KWK05] J. Koen Kok, Cor J. Warmer und I. G. Kamphuis. „PowerMatcher: Multiagent Control in the Electricity Infrastructure“. In: *Proceedings of the fourth international joint conference on Autonomous agents and multi-agent systems (AAMAS '05)*. ACM Press, 2005. DOI: [10.1145/1082473.1082807](https://doi.org/10.1145/1082473.1082807) (siehe S. 10)
- [Leh10] Sebastian Lehnhoff. *Dezentrales vernetztes Energiemanagement. Ein Ansatz auf Basis eines verteilten adaptiven Realzeit-Multiagentensystems*. Springer, 2010. DOI: [10.1007/978-3-8348-9658-2](https://doi.org/10.1007/978-3-8348-9658-2) (siehe S. 1)
- [LMW10] Mikhail A. Lisovich, Deirdre K. Mulligan und Stephen B. Wicker. „Inferring Personal Information from Demand-Response Systems“. In: *IEEE Security & Privacy Magazine* 8.1 (2010), S. 11–20. DOI: [10.1109/MSP.2010.40](https://doi.org/10.1109/MSP.2010.40) (siehe S. 5)
- [LPJ08] Jiaming Li, Geoff Poulton und Geoffrey James. „Distributed Energy Management“. Patent WO 2008/014562 A1. 2008 (siehe S. 12, 54, 205)
- [LPJ10] Jiaming Li, Geoff Poulton und Geoffrey James. „Coordination of Distributed Energy Resource Agents“. In: *Applied Artificial Intelligence* 24.5 (2010), S. 351–380. DOI: [10.1080/08839514.2010.481483](https://doi.org/10.1080/08839514.2010.481483) (siehe S. 10, 12, 54, 204, 205)
- [Lün12] Ontje Lünsdorf. „Selbstorganisation virtueller Geräte für das Lastmanagement von Kleinverbrauchern“. Diss. Carl von Ossietzky Universität Oldenburg, 2012. URN: [urn:nbn:de:gbv:715-oops-14609](https://nbn-resolving.org/urn:nbn:de:gbv:715-oops-14609) (siehe S. 113)
- [LV90] Gunar E. Liepins und Michael D. Vose. „Representational issues in genetic optimization“. In: *Journal of Experimental & Theoretical Artificial Intelligence* 2.2 (1990). DOI: [10.1080/09528139008953717](https://doi.org/10.1080/09528139008953717) (siehe S. 26)
- [Lyn96] Nancy Ann Lynch. *Distributed Algorithms*. Hrsg. von Jim Gray. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, Inc. – An Imprint of Elsevier, 1996 (siehe S. 41, 51, 93, 96)

- [Mar13] Sönke Martens. „Grundlagen der Selbstorganisationsmechanismen ‚Agent Discovery‘ und ‚Termination Detection‘ im Kontext eines agentenbasierten Einsatzplanungsverfahrens“. Bachelorarbeit. Carl von Ossietzky Universität Oldenburg, 2013. URN: [urn:nbn:de:gbv:715-oops-16860](https://nbn-resolving.org/urn:nbn:de:gbv:715-oops-16860) (siehe S. [83](#), [84](#))
- [Mat89] Friedemann Mattern. *Verteilte Basisalgorithmen*. Informatik-Fachberichte 226. Springer, 1989. DOI: [10.1007/978-3-642-84002-9](https://doi.org/10.1007/978-3-642-84002-9) (siehe S. [41](#), [52](#), [83](#))
- [McA+07] Stephen D. J. McArthur u. a. „Multi-Agent Systems for Power Engineering Applications—Part I: Concepts, Approaches, and Technical Challenges“. In: *IEEE Transactions on Power Systems* 22.4 (2007), S. 1743–1752. DOI: [10.1109/TPWRS.2007.908471](https://doi.org/10.1109/TPWRS.2007.908471) (siehe S. [4](#), [30](#))
- [McN12] Paul McNamara. „Optimisation of Smart Grid Performance using Centralised and Distributed Control Techniques“. Diss. National University of Ireland, 2012 (siehe S. [5](#), [27](#), [117](#))
- [MD07] Nihar R. Mahapatra und Shantanu Dutt. „An efficient delay-optimal distributed termination detection algorithm“. In: *Journal of Parallel and Distributed Computing* 67.10 (2007), S. 1047–1066. DOI: [10.1016/j.jpdc.2007.05.013](https://doi.org/10.1016/j.jpdc.2007.05.013) (siehe S. [83](#))
- [MGK05] Giovanna di Marzo Serugendo, Marie-Pierre Gleizes und Anthony Karageorgos. „Self-organization in multi-agent systems“. In: *The Knowledge Engineering Review* 20.02 (2005), S. 165–189. DOI: [10.1017/S0269888905000494](https://doi.org/10.1017/S0269888905000494) (siehe S. [36](#), [38](#))
- [MN98] Makoto Matsumoto und Takuji Nishimura. „Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator“. In: *ACM Transactions on Modeling and Computer Simulation* 8.1 (1998), S. 3–30. DOI: [10.1145/272991.272995](https://doi.org/10.1145/272991.272995) (siehe S. [127](#))
- [Mos+06] Md. Mostofa Akbar u. a. „Solving the Multidimensional Multiple-choice Knapsack Problem by constructing convex hulls“. In: *Computers & Operations Research* 33.5 (2006), S. 1259–1273. DOI: [10.1016/j.cor.2004.09.016](https://doi.org/10.1016/j.cor.2004.09.016) (siehe S. [128](#))

- [Neg07] Rudy Negenborn. „Multi-Agent Model Predictive Control with Applications to Power Networks“. Diss. Technische Universiteit Delft, 2007 (siehe S. 117)
- [Nes07] David Nestle. „Energiemanagement in der Niederspannungsversorgung mittels dezentraler Entscheidung“. Diss. Universität Kassel, 2007 (siehe S. 4, 5)
- [Nie+12] Astrid Nieße u. a. „Market-Based Self-Organized Provision of Active power and Ancillary services: An Agent-Based Approach for Smart Distribution Grids“. In: *Complexity in Engineering (COMPENG)*, 2012. 2012. DOI: [10.1109/CompEng.2012.6242953](https://doi.org/10.1109/CompEng.2012.6242953) (siehe S. 6)
- [NLH10] Rudy R. Negenborn, Zofia Lukso und Hans Hellendoorn, Hrsg. *Intelligent Infrastructures*. Bd. 42. Intelligent Systems, Control and Automation: Science and Engineering. Springer, 2010. DOI: [10.1007/978-90-481-3598-1](https://doi.org/10.1007/978-90-481-3598-1) (siehe S. 4, 5, 27)
- [NS13] A. Nieße und M. Sonnenschein. „Using Grid Related Cluster Schedule Resemblance for Energy Rescheduling“. In: *SMARTGREENS 2013 – 2nd International Conference on Smart Grids and Green IT Systems*. Hrsg. von Brian Donnellan u. a. INSTICC Press, 2013, S. 22–31 (siehe S. 76)
- [Pad04] Narayana Prasad Padhy. „Unit Commitment—A Bibliographical Survey“. In: *IEEE Transactions on Power Systems* 19.2 (2004), S. 1196–1205. DOI: [10.1109/TPWRS.2003.821611](https://doi.org/10.1109/TPWRS.2003.821611) (siehe S. 8)
- [PB05] Christian Prehofer und Christian Bettstetter. „Self-Organization in Communication Networks: Principles and Design Paradigms“. In: *IEEE Communications Magazine* 43.7 (2005), S. 78–85. DOI: [10.1109/MCOM.2005.1470824](https://doi.org/10.1109/MCOM.2005.1470824) (siehe S. 36–38, 52, 54, 72, 76)
- [Peh+06] Martin Pehnt u. a. *Micro Cogeneration*. Springer, 2006. DOI: [10.1007/3-540-30821-0](https://doi.org/10.1007/3-540-30821-0) (siehe S. 2, 51)
- [Peñ06] Yoseba Koldobika Peña Landaburu. „Optimal Allocation and Scheduling of Demand in Deregulated Energy Markets“. Diss. Technische Universität Wien, 2006 (siehe S. 5, 10, 11)

- [Pis95] David Pisinger. „A minimal algorithm for the Multiple-Choice Knapsack Problem“. In: *European Journal of Operational Research* 83.2 (1995), S. 394–410. DOI: [10.1016/0377-2217\(95\)00015-1](https://doi.org/10.1016/0377-2217(95)00015-1) (siehe S. [128](#))
- [Pis99] David Pisinger. „Linear Time Algorithms for Knapsack Problems with Bounded Weights“. In: *Journal of Algorithms* 33.1 (1999), S. 1–14. DOI: [10.1006/jagm.1999.1034](https://doi.org/10.1006/jagm.1999.1034) (siehe S. [46](#), [47](#), [128](#))
- [Pos80] John Postel. *User Datagram Protocol*. RFC 768 (Internet Standard). The Internet Engineering Task Force (IETF), 1980 (siehe S. [52](#))
- [Pou13] Evangelos Pournaras. „Multi-level Reconfigurable Self-organization in Overlay Services“. Diss. Technische Universiteit Delft, 2013 (siehe S. [5](#), [16](#), [27](#))
- [PS98] Christos H. Papadimitriou und Kenneth Steiglitz. *Combinatorial Optimization. Algorithms and Complexity*. Dover Publications, 1998 (siehe S. [24](#))
- [PWS99] H. Van Dyke Parunak, Allen C. Ward und John A. Sauter. „The Mar-Con algorithm: A systematic market approach to distributed constraint problems“. In: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 13.03 (1999), S. 217–234. DOI: [10.1017/S0890060499133079](https://doi.org/10.1017/S0890060499133079) (siehe S. [27](#))
- [Ram+11] Sarvapali D Ramchurn u. a. „Agent-Based Homeostatic Control for Green Energy in the Smart Grid“. In: *ACM Transactions on Intelligent Systems and Technology* 2.4 (2011), S. 1–28. DOI: [10.1145/1989734.1989739](https://doi.org/10.1145/1989734.1989739) (siehe S. [5](#))
- [Ram+12] Sarvapali D. Ramchurn u. a. „Putting the “Smarts” into the Smart Grid: A Grand Challenge for Artificial Intelligence“. In: *Communications of the ACM* 55.4 (2012). DOI: [10.1145/2133806.2133825](https://doi.org/10.1145/2133806.2133825) (siehe S. [4](#), [5](#))
- [Rau03] Martin Rau. „Nichtlineare modellbasierte prädiktive Regelung auf Basis lernfähiger Zustandsraummodelle“. Diss. Technische Universität München, 2003 (siehe S. [117](#))

- [Rey87] Craig W. Reynolds. „Flocks, Herds and Schools: A Distributed Behavioral Model“. In: *ACM SIGGRAPH Computer Graphics* 21.4 (1987), S. 25–34. DOI: [10.1145/37402.37406](https://doi.org/10.1145/37402.37406) (siehe S. [55](#))
- [RN10] Stuart Jonathan Russell und Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3. Aufl. Prentice Hall Series in Artificial Intelligence. Prentice Hall, 2010 (siehe S. [30](#), [36](#))
- [RS13] Sebastian Rohjans und Michael Specht. „OPC UA: An Automation Standard for Future Smart Grids“. In: *Standardization in Smart Grids*. Power Systems. Springer, 2013. Kap. 12. DOI: [10.1007/978-3-642-34916-4_12](https://doi.org/10.1007/978-3-642-34916-4_12) (siehe S. [123](#))
- [RUS13] Christine Rosinger, Mathias Uslar und Jürgen Sauer. „Threat Scenarios to evaluate Trustworthiness of Multi-agents in the Energy Data Management“. In: *27th International Conference on Environmental Informatics for Environmental Protection, Sustainable Development and Risk Management (EnviroInfo 2013)*. Hrsg. von Bernd Page u. a. Berichte aus der Umweltinformatik. Shaker, 2013, S. 258–264 (siehe S. [216](#))
- [Sch+13] Steffen Schütte u. a. „OPC UA Compliant Coupling of Multi-Agent Systems and Smart Grid Simulations“. In: *39th Annual Conference of the IEEE Industrial Electronics Society (IECON 2013)*. IEEE Industrial Electronics Society, 2013 (siehe S. [123](#))
- [See11] Hella Seebach. „Konstruktion selbst-organisierender Softwaresysteme“. Diss. Universität Augsburg, 2011 (siehe S. [35](#))
- [SF86] Nir Shavit und Nissim Francez. „A new approach to detection of locally indicative stability“. In: *Automata, Languages and Programming*. Hrsg. von Laurent Kott. Bd. 226. Lecture Notes in Computer Science. Springer, 1986, S. 344–358. DOI: [10.1007/3-540-16761-7_84](https://doi.org/10.1007/3-540-16761-7_84) (siehe S. [83](#))
- [Son+12] Michael Sonnenschein u. a. „Dezentrale und selbstorganisierte Koordination in Smart Grids“. In: *VDE-Kongress 2012 – Intelligente Energieversorgung der Zukunft*. 2012 (siehe S. [39](#), [42](#), [76](#), [79](#))

- [SOR07] Hella Seebach, Frank Ortmeier und Wolfgang Reif. „Design and Construction of Organic Computing Systems“. In: *IEEE Congress on Evolutionary Computation 2007*. IEEE, 2007, S. 4215–4221. DOI: [10.1109/CEC.2007.4425021](https://doi.org/10.1109/CEC.2007.4425021) (siehe S. 35)
- [Sta+07] Michael Stadler u. a. „The Adaptive Fridge – Comparing Different Control Schemes for Enhancing Load Shifting of Electricity Demand“. In: *21th International Conference on Environmental Informatics for Environmental Protection, Environmental Informatics and Systems Research (EnviroInfo 2007)*. Hrsg. von Olgierd Hryniewicz, Jan Studzinski und Maciej Romaniuk. Shaker Verlag, 2007, S. 199–206 (siehe S. 197)
- [Sta+09] Michael Stadler u. a. „Modelling and evaluation of control schemes for enhancing load shift of electricity demand for cooling devices“. In: *Environmental Modelling & Software* 24.2 (2009), S. 285–295. DOI: [10.1016/j.envsoft.2008.07.003](https://doi.org/10.1016/j.envsoft.2008.07.003) (siehe S. 197)
- [Ste+13a] Jan-Philipp Steghöfer u. a. „HiSPADA: Self-Organising Hierarchies for Large-Scale Multi-Agent Systems“. In: *Ninth International Conference on Autonomic and Autonomous Systems (ICAS 2013)*. IARIA, 2013, S. 71–76 (siehe S. 15)
- [Ste+13b] Jan-Philipp Steghöfer u. a. „Synthesis of Observers for Autonomic Evolutionary Systems from Requirements Models“. In: *6th Intl. Workshop on Distributed Autonomous Network Management Systems (DANMS 2013) – 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*. IEEE, 2013, S. 1405–1408 (siehe S. 35)
- [Ste13] Michael Horst Emil Steck. „Entwicklung und Bewertung von Algorithmen zur Einsatzplanerstellung virtueller Kraftwerke“. Diss. Technische Universität München, 2013 (siehe S. 5, 8)
- [Str01] Steven Strogatz. „Exploring complex networks“. In: *Nature* 410.6825 (2001), S. 268–76. DOI: [10.1038/35065725](https://doi.org/10.1038/35065725) (siehe S. 81, 126, 138)
- [Tal09] El-Ghazali Talbi. *Metaheuristics: From Design to Implementation*. John Wiley & Sons, Inc., 2009. DOI: [10.1002/9780470496916](https://doi.org/10.1002/9780470496916) (siehe S. 54, 179, 214)

- [Tel00] Gerard Tel. *Introduction to Distributed Algorithms*. 2. Aufl. Cambridge University Press, 2000 (siehe S. 90)
- [Tho07] Bernd Thomas. *Mini-Blockheizkraftwerke*. Vogel Buchverlag, 2007 (siehe S. 43–45)
- [Trö10] Martin Tröschel. „Aktive Einsatzplanung in holonischen Virtuellen Kraftwerken“. Diss. Carl von Ossietzky Universität Oldenburg, 2010 (siehe S. 2, 4, 5, 14, 39, 41, 44, 45, 76, 119)
- [TRY3] *Testreferenzjahre von Deutschland für mittlere und extreme Witterungsverhältnisse (TRY)*. TRY Region 3. Deutscher Wetterdienst, 2010 (siehe S. 187)
- [UM11] Dieter Urban und Jochen Mayerl. *Regressionsanalyse: Theorie, Technik und Anwendung*. VS Verlag für Sozialwissenschaften, 2011. DOI: 10.1007/978-3-531-93114-2 (siehe S. 165)
- [Van06] Koen Vanthournout. „A semantic overlay network based robust data-infrastructure, applied to the electric power grid“. Diss. Katholieke Universiteit Leuven, 2006 (siehe S. 4, 57)
- [VDE10] Ralf Allrutz u. a. *VDE/ITG/GI Positionspapier Organic Computing*. Techn. Ber. Gesellschaft für Informatik (GI) und Informationstechnische Gesellschaft im VDE (ITG), 2010 (siehe S. 72)
- [VDI4655] *Referenzlastprofile von Ein- und Mehrfamilienhäusern für den Einsatz von KWK-Anlagen*. VDI 4655 (VDI Richtlinie). Verein Deutscher Ingenieure, 2008 (siehe S. 187)
- [Wei13] Gerhard Weiss, Hrsg. *Multiagent Systems*. 2. Aufl. MIT Press, 2013 (siehe S. 36, 50)
- [Wim04] Roger Wilfried Wimmer. „Regelung einer Wärmepumpenanlage mit Model Predictive Control“. Diss. Eidgenössische Technische Hochschule Zürich, 2004 (siehe S. 117)
- [Win09] Ludger Winkels. „Referenzmodell für die Tageseinsatzplanung dezentraler heterogener Energieerzeugungsanlagen“. Diss. Carl von Ossietzky Universität Oldenburg, 2009 (siehe S. 45)

- [WMB05] Felix F. Wu, Khosrow Moslehi und Anjan Bose. „Power System Control Centers: Past, Present, and Future“. In: *Proceedings of the IEEE* 93.11 (2005), S. 1890–1908. DOI: [10.1109/JPROC.2005.857499](https://doi.org/10.1109/JPROC.2005.857499) (siehe S. [3](#), [27](#))