

Verteilte Methoden zur Lösung von beschränkten Optimierungsproblemen in Multiagentensystemen

Vom Fachbereich
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des akademischen Grades
eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von

M. Sc. Jan Christian Zimmermann

1. Gutachter/-in: Dr. rer. nat. T. Tatarenko
2. Gutachter/-in: Prof. Dr.-Ing. J. Adamy
3. Gutachter/-in: Prof. Dr.-Ing. habil. J. Deutscher
Tag der Einreichung: 28. März 2023
Tag der mündlichen Prüfung: 23. Juni 2023

D17
Darmstadt 2023

Zimmermann, Jan Christian: Verteilte Methoden zur Lösung von be-
schränkten Optimierungsproblemen in Multiagentensystemen
Darmstadt, Technische Universität Darmstadt,
Jahr der Veröffentlichung der Dissertation auf TUPrints: 2023
URN: [urn:nbn:de:tuda-tuprints-245639](https://nbn-resolving.org/urn:nbn:de:tuda-tuprints-245639)
Tag der mündlichen Prüfung: 23.06.2023

Veröffentlicht unter CC BY-SA 4.0 International
<https://creativecommons.org/licenses/by-sa/4.0/>

Füg auch du
Schaffend dein Scherflein hinzu;
Klein und zag,
Atme, soviel deine Kraft vermag,
Einen Zug in den ewigen Tag!

- Bjørnstjerne Bjørnson

Förderung

Die vorliegende Arbeit wurde durch die Deutsche Forschungsgemeinschaft (DFG) im Rahmen des Schwerpunktprogrammes 1984 "Hybrid and multi-modal energy systems: System theoretical methods for the transformation and operation of complex networks" gefördert.

Danksagung

Die vorliegende Arbeit entstand während meiner Zeit als Wissenschaftlicher Mitarbeiter am Fachgebiet Regelungsmethoden und Intelligente Systeme, ehemals Regelungsmethoden und Robotik, der Technischen Universität Darmstadt.

Zunächst möchte ich mich bei Prof. Jürgen Adamy für die Möglichkeit bedanken, an seinem Fachgebiet zu promovieren. Bei all meinen Fragen und Problemen stand Jürgen mir immer mit Rat, Tat und allerlei hilfreichen Tipps zur Seite.

Prof. Joachim Deutscher danke ich dafür, dass er sich bereit erklärt hat, meine Arbeit zu begutachten.

Tatiana Tatarenko und Prof. Volker Willert danke ich zum einen für die erfolgreiche Einreichung des ursprünglichen Forschungsantrages bei der DFG, wodurch die Finanzierung für mein Forschungsprojekt gesichert wurde. Zum anderen bin ich Tatiana und Volker für die Betreuung meiner Forschung dankbar sowie für die unzähligen Gespräche, sowohl fachlich als auch privat, die entscheidend zum Gelingen meines Projektes beigetragen haben.

Im Sekretariat laufen alle Stränge des Fachgebietes zusammen. Ich bin dankbar, dass ich dort mit Birgit Heid und Susanne Muntermann immer zuverlässige Ansprechpartnerinnen zu den unterschiedlichsten Themen fand.

Danke an alle meine Kolleginnen und Kollegen, die mich während meiner Arbeit am Fachgebiet, zumindest abschnittsweise, begleitet haben. Trotz der hohen Ansprüche der individuellen Forschungsvorhaben, lief die Zusammenarbeit am Fachgebiet immer wie eine ausgezeichnet geregelte Maschine. Auch im Hinblick auf die besonders herausfordernde Zeit

während der Covid-19 Pandemie ist das keine Selbstverständlichkeit. Die nicht immer fachlichen Diskussionen und Kaffeerunden waren immer eine willkommene Ablenkung vom Forschungsalltag. Meinem ehemaligen Bürokollegen Thomas Schmitt möchte ich für die gute Nachbarschaft und die inspirierenden Gespräche danken. Aber höhenverstellbare Tische haben eben auch ihren Reiz. Besonderer Dank gilt Florian Meiners, Nikolas Hohmann und Linus Groß, die trotz ihres hohen Aufgabenpensums die Zeit gefunden haben, die vorliegende Arbeit Korrektur zu lesen. No RIS, no fun!

Ebenfalls entscheidenden Anteil an meiner Promotion haben meine Freundinnen und Freunde in Frankfurt + Umgebung. Danke für die Tage und Nächte, während denen ich nicht an meine Arbeit gedacht habe. Auch möchte ich mich an dieser Stelle bei meiner Kerngruppe während des Studiums, bestehend aus Hikmet Baran, Onur Celik, Lukas Thome und Mirkan Yirik, für die gemeinsame Zeit und die Motivation bedanken.

Meiner Familie danke ich für die unerschütterliche Unterstützung während meiner Doktorandenzeit, während meines Studiums, während meines ganzen Lebens. Ihr seid meine Heimat.

Und schließlich danke ich meiner Partnerin Saskia. Dafür, dass sie mein Leben bereichert und meinen Horizont erweitert hat. Und für die vielen kleinen Dinge.

Hannover, 27. März 2023

Jan Zimmermann

Inhaltsverzeichnis

Notation	XI
Kurzfassung	XIII
Abstract	XV
1 Einleitung	1
1.1 Verteilte Optimierungsmethoden	3
1.2 Einordnung	7
1.3 Beiträge der Arbeit	8
1.4 Struktur der Arbeit	11
2 Grundlagen	13
2.1 Zentrale, dezentrale und verteilte Entscheidungsfindung . .	13
2.2 Mathematische Grundlagen	15
2.2.1 Reelle Abbildungen	15
2.2.2 Mengen	17
2.3 Lösung konvexer Optimierungsprobleme durch Gradienten- abstieg	17
2.4 Nicht-kooperative Spieltheorie	27
2.5 Kommunikation in Multiagentensystemen	32
2.6 Konsensus in Multiagentensystemen	40
2.6.1 Mittelwertkonsensus bei spalten-stochastischer Ge- wichtung: Das Push-Sum Protokoll	43
2.6.2 Mittelwertkonsensus bei reihen-stochastischer Ge- wichtung	44
3 Optimaler Einsatzplan für nicht-kooperierende Microgrids	46
3.1 Allgemeine Beschreibung der Problemstellung	47
3.2 Bestandteile der Microgrids	49
3.2.1 Verbraucher	49
3.2.2 Dezentrale Generatoren	50
3.2.3 Dezentrale Speicher	51

3.2.4	Übertragungsnetzwerk	52
3.2.5	Hauptnetz	55
3.3	Formulierung des Optimierungsproblems	55
3.4	Einordnung in relevante Literatur	59
3.4.1	Kooperative und nicht-kooperative Einsatzplanprobleme in der Literatur	59
3.4.2	Gemeinsamkeiten zwischen Problemformulierung und der Literatur	61
3.4.3	Unterschiede zwischen Problemformulierung und der Literatur	61
3.5	Herausforderungen bezüglich der verteilten Lösung	64
4	Verteilte Lösung von kooperativen Optimierungsproblemen	66
4.1	Einleitung	66
4.2	Problemstellung	67
4.3	Stand der Forschung und Beiträge	70
4.3.1	Stand der Forschung: Gradientenbasierte Verfahren für verteilte Optimierung	73
4.3.2	Stand der Forschung: Zusammenfassung	79
4.3.3	Beiträge	82
4.4	Algorithmus 1: Verteilter Gradientenabstieg mit Straftermen	84
4.4.1	Problemformulierung unter Verwendung von Straftermen	85
4.4.2	Algorithmus	87
4.4.3	Konvergenz	90
4.4.4	Fazit	94
4.5	Algorithmus 2: Verteilter primal-dualer Gradientenabstieg .	95
4.5.1	Formulierung als verteiltes Sattelpunktproblem . . .	96
4.5.2	Algorithmus	97
4.5.3	Konvergenz	100
4.5.4	Fazit	109
4.6	Algorithmus 3 - Projiziertes Gradient-Tracking-Verfahren .	109
4.6.1	Problemformulierung unter Verwendung einer globalen Nebenbedingungsmenge	110
4.6.2	Algorithmus	111
4.6.3	Konvergenz	113
4.6.4	Fazit	116
4.7	Simulationen: Optimaler Generatoreinsatzplan	117
4.7.1	Simulationsmodell	117

4.7.2	Szenarien und Setup	119
4.7.3	Lösung der Optimierungsprobleme	122
4.7.4	Konvergenzverhalten und -geschwindigkeit der Algorithmen 1 - 3	125
4.8	Fazit	141
5	Verteilte Lösung von Multiclusterspielen	143
5.1	Einleitung	143
5.2	Problemstellung	144
5.2.1	Übersicht über die Notation	150
5.2.2	Lösungsbedingungen von Multiclusterspielen	151
5.2.3	Sonderfälle	153
5.3	Stand der Forschung und Beiträge	154
5.3.1	Zeitkontinuierliche Algorithmen	154
5.3.2	Zeitdiskrete Algorithmen	155
5.4	Algorithmus 4: Lösung beschränkter Multiclusterspiele	162
5.4.1	Algorithmus	163
5.4.2	Konvergenz	166
5.4.3	Fazit	170
5.5	Simulationen: Einsatzplanproblem zwischen Microgrids	171
5.5.1	Szenario und Setup	171
5.5.2	Analyse des Nash-Gleichgewichts des Einsatzplanspiels	174
5.5.3	Konvergenz	179
5.6	Fazit	183
6	Fazit	184
6.1	Zusammenfassung	184
6.2	Ausblick	186
A	Anhang zum Kapitel 4	189
A.1	Unterstützende Theoreme für Konvergenzbeweis von Algorithmus 1	189
A.2	Beweis von Lemma 4.4.1	189
A.3	Beweis von Theorem 4.4.1	190
A.4	Beweis von Lemma 4.5.1	193
A.5	Beweis von Lemma 4.5.2	194
A.6	Beweis von Lemma 4.5.3	195
A.7	Beweis von Lemma 4.5.4	195
A.8	Beweis von Satz 4.5.1	198

A.9	Äquivalenz zwischen kooperativen Einsatzplanproblem unter Leistungsgleichung und -ungleichung	201
A.10	Parameterkonfiguration für Simulationen	204
A.11	Kommunikationsgraphen für Busnetzwerke	207
B	Anhang zum Kapitel 5	208
B.1	Beweis von Lemma 5.2.1	208
B.2	Beweis Lemma 5.4.2	209
B.3	Beweis von Satz 5.4.1	214
B.4	Parameterkonfiguration für Simulationen des Einsatzplanproblems zwischen Microgrids	218
B.5	Kommikuationsgraphen für Simulationen aus Abschnitt 5.5	221
	Veröffentlichungen	222
	Wissenschaftliche Vorträge	223
	Literatur	224

Notation

s, v	Skalar, Vektor.
M	Matrix.
$\mathbf{1}$	Vektor, dessen Elemente gleich eins sind.
I	Identitätsmatrix.
$A_{i,:}$	Auswahl der i -ten Zeile der Matrix A .
$x < y$	Alle Einträge von x sind kleiner als die entsprechenden Einträge von y .
$\ x\ $	Euklidische Norm für Vektor x .
$\ X\ _q$	q -Norm der Matrix X .
$\langle x, y \rangle$	Skalarprodukt der Vektoren x, y .
$\text{proj}_{\mathcal{X}}[u]$	Projektion von Vektor u auf Menge \mathcal{X} , sodass $\text{proj}_{\mathcal{X}}[u] = \text{argmin}_{x \in \mathcal{X}} \ x - u\ $.
$\text{col}(x_1, \dots, x_N)$	Erzeugt einen Spaltenvektor mit den Elementen x_1, \dots, x_N .
$\text{diag}(A_1, \dots, A_N)$	Erzeugt eine Diagonalmatrix aus den Matrizen.
$\text{diag}(v)$	Expandiert den Vektor v in eine Matrix, deren Diagonale aus den Elementen des Vektoren besteht und deren restliche Elemente gleich null sind.
$ \mathcal{A} $	Kardinalität der Menge \mathcal{A} .
$f = o(g)$	$\forall C > 0 \exists x_0 > 0 \forall x > x_0 : f(x) < C g(x) $, wenn $x \rightarrow \infty$.
$f = O(g)$	$\exists C > 0 \exists x_0 > 0 \forall x > x_0 : f(x) < C g(x) $, wenn $x \rightarrow \infty$.
$\nabla f(x_0)$	Entspricht dem Gradienten der Funktion $f(x)$, ausgewertet am Punkt x_0 , d. h. $\nabla_x f(x) _{x=x_0}$.
$\nabla_h f(x_0^h, x_0^{-h})$	Entspricht dem Gradienten der Funktion $f(x^1, \dots, x^h, \dots, x^H)$ bezüglich x^h , ausgewertet am Punkt x_0^h , d. h. $\nabla_{x^h} f(x^h, x^{-h}) _{x^h=x_0^h, x^{-h}=x_0^{-h}}$.
$\nabla_x f(x_0, y_0)$	Entspricht dem Gradienten der Funktion $f(x, y)$ bezüglich x , ausgewertet am Punkt x_0 , d. h. $\nabla_x f(x, y) _{x=x_0, y=y_0}$.
$1_{\mathcal{X}}(x)$	Indikatorfunktion der Menge \mathcal{X} , sodass $1_{\mathcal{X}}(x) = 1$, wenn $x \in \mathcal{X}$ und sonst $1_{\mathcal{X}}(x) = 0$.

$\operatorname{sgn}(\cdot)$	Vorzeichenfunktion, entspricht dem Vorzeichen des Arguments.
\otimes	Kroneckerprodukt.
$\rho(A)$	Spektraler Radius der quadratischen Matrix A .
$\mathcal{X} \times \mathcal{V}$	Kartesisches Produkt der Mengen \mathcal{X} und \mathcal{V} .

Kurzfassung

Besteht ein Optimierungsproblem aus einer Menge von einzelnen Kostenfunktionen und Nebenbedingungen, die auf ein Multiagentensystem verteilt sind, so wird von einem verteilten Optimierungsproblem gesprochen. Dabei besitzt jeder Agent eine der Kostenfunktionen und eine gewisse Untermenge der Nebenbedingungen. Je nachdem, ob die Agenten zusammenarbeiten, um das Problem gemeinsam zu lösen, oder gegeneinander bezüglich ihrer verkoppelten Kostenfunktionen konkurrieren, wird die Problemstellung als kooperativ oder nicht-kooperativ bzw. als spieltheoretisches Problem bezeichnet. Um eine robuste, ausfallsichere Lösung der jeweiligen Problemstellung zu erreichen, bei der zusätzlich die Kostenfunktionen bzw. Nebenbedingungen der Agenten privat bleiben, müssen verteilte Optimierungsmethoden entwickelt werden, die auf einer möglichst großen Menge von Kommunikationsarchitekturen konvergieren. Darüber hinaus existieren verteilte Problemstellungen, bei denen eine Untermenge von Agenten des Netzwerkes kooperativ zusammenarbeitet, während mit anderen Agenten ein nicht-kooperatives Verhältnis besteht. Für diese Art von Problemen ist eine entsprechend angepasste, verteilte Lösungsmethode notwendig.

In der vorliegenden Arbeit werden zur Lösung von kooperativen, verteilten Problemstellungen unbeschränkte, gradientenbasierte Verfahren erweitert, sodass Nebenbedingungen berücksichtigt werden können. Dabei kommen die Techniken der Strafterme, der Lagrangeparameter bzw. des dualen Ansatzes und der Projektion in drei verschiedenen Algorithmen zum Einsatz. Neben der Berücksichtigung von Beschränkungen ist ein weiteres Ziel, dass die resultierenden Algorithmen auf einer möglichst großen Menge von Kommunikationsarchitekturen konvergieren. Die Konvergenz der Verfahren wird mathematisch bewiesen und durch Simulationen veranschaulicht.

Zusätzlich werden beschränkte Multiclusterspiele betrachtet, die aus der Kombination einer kooperativen und nicht-kooperativen Problemstellung bestehen. Damit das Problem als gelöst betrachtet werden kann,

muss ein stabiler Zustand, d. h. ein Nash-Gleichgewicht, zwischen den konkurrierenden Parteien und gleichzeitig ein kooperatives Optimum bezüglich der zusammenarbeitenden Agenten erreicht werden. Für diese Problemstellung wird ein gradientenbasiertes Verfahren vorgestellt, das die Problemstellung unter Verwendung eines Gradientenschätzverfahrens und der Projektion der Aktualisierungsgleichungen auf die jeweilige Nebenbedingungsmenge lösen kann. Im theoretischen Teil wird lineare Konvergenz des Verfahrens zum Optimum nachgewiesen, während in Simulationen die Effizienz des Verfahrens gezeigt wird.

Alle im Rahmen der vorliegenden Arbeit durchgeführten Simulationen werden auf eine energietechnische Problemstellung bezogen. Dabei wird ein Einsatzplanproblem zwischen Microgrids betrachtet, bei dem der optimale, kooperative Einsatz von Generatoren und Speichern innerhalb der Microgrids geplant wird, während die Microgrids selbst gegeneinander in einer Marktsituation bezüglich des Strompreises eines Hauptnetzes konkurrieren.

Abstract

A distributed optimization problem is defined as a set of cost functions and constraints that are distributed over a multi-agent system. Each of these agents holds one of the cost functions and a specific subset of the overall constraint set. The distributed problem is denoted as either cooperative or non-cooperative, i. e. game theoretic, depending on whether the agents work together to solve the problem at hand or compete against each other regarding their coupled cost functions. In order to achieve a robust and fail-safe solution of the problem, while keeping the cost functions and constraints private, distributed optimization procedures need to be employed. Additionally, distributed problems exist, wherein a cooperative behaviour is exhibited by a subset of agents, while there is competition with other agents of the network. In order to solve such problems, distributed algorithms need to be accordingly adapted.

In this thesis, unconstrained and gradient-based algorithms are extended, such that constraints can be handled in cooperative environments. For that, penalty functions, Langrange multipliers, i. e. dual approaches, and projections are employed in three distinct algorithms. Another goal next to the consideration of constraints is to make the algorithms applicable to a wide range of communication architectures. For all three algorithms, convergence is mathematically proven and illustrated by simulations.

Furthermore, constrained multi-cluster games are considered, which consist of a combination of cooperative and non-cooperative problems. In order to achieve a solution of such games, a stable state, i. e. a Nash Equilibrium, needs to be reached between the non-cooperating parties while simultaneously achieving local optima inside the cooperating groups. This problem is solved in this work by a gradient-based algorithm that estimates the gradients in the cooperative clusters and projects the gradient steps onto the considered constraint sets. Linear convergence is proven, while simulations show the practical efficiency of the procedure.

All simulations are performed on the basis of a power engineering problem, which is posed as an optimal dispatch problem between microgrids. Herein, the optimal dispatch of cooperating generators and storage devices needs to be set inside the microgrids, while the latter compete against each other in a market situation regarding power provided by the main grid.

1 Einleitung

Wir leben in einer hochgradig vernetzten Welt. Menschen kommunizieren täglich miteinander, sei es über das Internet, den Mobilfunk oder persönlich. Diese Kommunikation erfolgt beispielsweise zur Koordination innerhalb von Gruppen, sodass ein gemeinsames Ziel erreicht werden kann. In Situation, in denen Individuen egoistisch Ziele verfolgen, kann es zu Streit mit einem oder mehreren Konkurrenten kommen. Steht die Schlichtung dieses Streits im Vordergrund, so ist in diesem Fall das Ziel, eine stabile Einigung zu erzielen, von der keiner der Kontrahenten einen Anreiz hat abzuweichen.

Ebenso wie sich menschliche Netzwerke durch den Aufstieg des Internets und der damit einhergehenden Möglichkeit, über weite Strecke miteinander zu kommunizieren, vergrößerten, bilden sich durch die kontinuierliche Weiterentwicklung der Kommunikations- und Sensortechnik auch immer mehr vernetzte Systeme im technischen Bereich. Bei diesen arbeiten die einzelnen Subsysteme entweder zusammen, um ein dem System übergeordnetes Ziel zu erreichen, oder es werden egoistisch eigene Ziele verfolgt, sodass Konkurrenzsituation mit anderen Subsystemen auftreten können.

Ein Beispiel für ein kooperatives System sind vernetzte Fahrzeuge, die über das 5G Mobilfunknetzwerk Informationen austauschen können. Dadurch wird es möglich, gemeinsame Fahrmanöver durchzuführen oder ein besseres Umgebungsbild durch den Austausch von Sensordaten zu generieren. Ziel hierbei ist es, Sicherheit und Effizienz aller Verkehrsteilnehmer zu steigern [1]. Diese Absprachen lassen sich auf weitere Robotersysteme verallgemeinern, siehe beispielsweise [2]. Konkurrenzsituationen treten in technischen Systemen oft dann auf, wenn eine gemeinsam genutzte Ressource verwaltet werden muss. Beispiele hierfür sind die Netzwerkwahl und Ressourcenverwaltung in drahtlosen Zugangsnetzwerken (engl.: wireless access networks) [3] und Marktsituation, wie beispielsweise das klassische Nash-Cournotspiel. Hierbei konkurriert eine Menge von Firmen, die alle das gleiche Gut produzieren, bezüglich des Verkaufspreises des Gutes, wobei dieser von der verfügbaren Menge auf dem Markt abhängt. Simulationen bezüglich dieser Problemstellung finden sich beispielsweise

in [4][5].

Ein weiteres Beispiel, das in der vorliegenden Arbeit auch genauer betrachtet wird, ist die zunehmende Dezentralisierung des Stromnetzwerkes und der damit einhergehenden Entwicklung eines großen verteilten Systems, das die Stromversorgung von Morgen sicher stellen wird. Mit dem technischen Stand der Vergangenheit waren hohe Wirkungsgrade und damit eine preisgünstige Stromproduktion nur bei der Generierung von großen Energiemengen möglich [6]. Folglich wurden zentrale Kraftwerke mit fossilen oder atomaren Brennstoffen gebaut, die über ein entsprechend ausgelegtes, großes Stromnetzwerk mit den Stromabnehmern verbunden waren. Um die Stromversorgung kostengünstig gestalten zu können, sammeln die Systemoperatoren die Informationen bezüglich der überschaubaren Menge an Kraftwerken und erstellen unter Verwendung eines entsprechenden Optimierungsverfahrens einen optimalen Einsatzplan, siehe beispielsweise [7][8]. Dadurch wird über einen bestimmten Zeitraum hinweg festgelegt, wann welches Kraftwerk wie viel Leistung produzieren muss, so dass der prädizierte Leistungsbedarf gedeckt wird und die Betriebskosten der Kraftwerke möglichst gering ausfallen.

Ein Aspekt der Energiewende ist die Dezentralisierung der Stromproduktion und eine zunehmende Notwendigkeit der, ebenfalls dezentralen, Stromspeicherung. Um die Stromversorgung zu modernisieren und zusätzlich weniger abhängig von fossilen Brennstoffen zu machen, wird vermehrt auf kleinere Generatoren, Windkraftwerke und Photovoltaikanlagen gesetzt, die sich in kürzerer Distanz zu den Stromabnehmern befinden [8]. Darüber hinaus werden neue Stromspeicher in das Netzwerk eingebracht, beispielsweise in Form von Plug-in E-Fahrzeugen [9] oder größeren Quartierspeichern [10], deren Speicherstrategien optimal festzulegen sind.

Um der Komplexität des so verteilten Systems entgegenzutreten, können Verbraucher, dezentrale Stromspeicher und Generatoren in sogenannte Microgrids zusammengefasst werden. Ein Microgrid bildet dabei ein eigenes Bilanzierungsgebiet und ist über eine Übertragungsleitung mit dem Hauptnetz und gegebenenfalls anderen Microgrids verbunden [11]. Durch diese Entwicklung entstehen zusätzlich neue Markt- und damit Konkurrenzsituation, bei denen die Microgrids an einem Energiemarkt bezüglich des Energiepreises konkurrieren [12]. Damit eine kostengünstige Stromversorgung weiterhin sichergestellt ist, muss ein optimaler Einsatzplan bezüglich der Generatorproduktion sowie der Stromspeicherstrategie berechnet werden, bei dem gleichzeitig die entstehenden Marktsituationen gelöst werden.

Bei klassischen, zentralen Optimierungsverfahren werden alle Informationen über ein zu lösendes Problem an einer einzigen, zentralen Stelle gesammelt und dort weiterverarbeitet. Diese Vorgehensweise kann bei kleineren, überschaubaren Netzwerken, wie beispielsweise dem klassischen Stromnetzwerk, noch effizient angewendet werden. Bei größeren, verteilten Problemstellungen weist die zentrale Lösung allerdings entscheidende Nachteile auf. Zum einen ist ab einer gewissen Größe des Netzwerkes die zentrale Verarbeitung und der damit verbundene Kommunikationsaufwand schlicht nicht praktikabel [13][14]. Zusätzlich stellt im zentralen Fall der Knoten, an dem die Informationen gesammelt werden, einen sogenannten Single-Point-Of-Failure dar: fällt dieser Knoten aus oder wird dieser Knoten angegriffen, so droht ein Kollaps des gesamten Systems bzw. können sämtliche Informationen des Netzwerkes abgezogen werden [13][14]. Darüber hinaus kann sich bei nicht-kooperativen Problemstellungen eine Entität des Netzwerkes, die sich mit einer anderen Einheiten in Konkurrenz befindet, schlicht weigern, seinen Antagonisten kritische Informationen preiszugeben. Entsprechend schwierig gestaltet sich die Auflösung von nicht-kooperativen Problemen, die, wie bereits angesprochen, in Energienetzen beispielsweise in Marktsituation auftreten, siehe [15][16][17]. Diese Nachteile gilt es zu überwinden, wenn verteilte Probleme effizient und sicher gelöst werden sollen.

1.1 Verteilte Optimierungsmethoden

Mit der verteilten Optimierung kann der Lokalität der verteilten Information in vernetzten Systemen Rechnung getragen werden. Hierbei werden die einzelnen Informationsträger zu eigenständig handelnden Agenten, die sich, durch Kommunikationsleitungen verknüpft, zu einem Multiagentensystem zusammenschließen. Allgemein wird hierbei davon ausgegangen, dass jeder Agent ein Teilproblem des zu lösenden Gesamtproblems besitzt, das aus Kostenfunktionen und Nebenbedingungen besteht. Durch eine fortlaufende Kommunikation untereinander können die Agenten das Gesamtproblem lösen, obwohl keiner der Agenten das gesamte Problem vollständig kennt. Dabei finden Berechnungen zur Identifikation einer optimalen Lösung der Problemstellung innerhalb bzw. bei den Agenten statt, während die Kommunikation für die Koordination bzw. Einigung mit anderen Agenten auf eine gemeinsame Aktion verwendet wird. Ein großer Vorteil dieser Verarbeitung ist, dass kritische Informationen, welche die

Agenten nicht preisgeben wollen, privat bleiben. Ebenso existiert im Vergleich zu zentralen Methoden kein Single-Point-Of-Failure: fällt ein Agent aus, kann zwar dessen Information nicht mehr abgerufen werden, allerdings kann der Optimierungsvorgang zwischen den restlichen Agenten weiter geführt werden. Damit kann unter entsprechend redundant ausgelegten Kommunikationsverbindung die Bearbeitung robust gestaltet werden [8]. Darüber hinaus besteht für verteilte Methoden das Potential, das Gesamtproblem des Agentensystems durch paralleles Berechnen schneller zu lösen, als es in zentralen Systemen möglich ist [8]. Zusätzlich besitzen verteilte Methoden grundsätzlich ein höheres Skalierungspotential als zentrale Ansätze [14].

Um diese Potentiale tatsächlich ausschöpfen zu können, müssen verteilte Algorithmen entwickelt werden, die den entsprechende Anforderungen der verteilten Information unter verschiedenen Kooperations- und Nicht-Kooperationsstrukturen der Agenten gerecht werden. Für unbeschränkte Problemstellungen existieren bereits viele Algorithmen, siehe beispielsweise [18][19][20][21][22], die unter Verwendung unterschiedlicher Herangehensweisen Lösungen finden können. Weniger Arbeiten existieren zur Lösung von beschränkten Problemstellungen. Eine weitere Herausforderung stellt sich, wenn die Nebenbedingungen der Agenten zusätzlich zu den Kostenfunktionen über den gesamten Optimierungsverlauf privat bleiben sollen. Das erste Ziel der vorliegenden Arbeit lässt sich unter diesen Gesichtspunkten wie folgt formulieren.

Ziel 1 (Nebenbedingungen) In der vorliegenden Arbeit soll der Forschungsstand bezüglich der verteilten Optimierung erweitert werden. Dazu werden bestehende Algorithmen für unbeschränkte, verteilte Optimierungsverfahren adaptiert, sodass Nebenbedingungen berücksichtigt werden können und möglichst auch der eingeschränkte Zugang zu diesen Nebenbedingungen erhalten bleibt.

Die Verfolgung dieser Zielsetzung ist relevant, da die oben genannten Vorteile der verteilte Optimierung gegenüber einer zentralen Variante nur dann zum Tragen kommen, wenn vergleichbare Klassen von Optimierungsproblemen gelöst werden können. Die so generierten Algorithmen sind dabei nicht auf bestimmte Optimierungsprobleme, wie beispielsweise das beschriebene Einsatzplanproblem im Stromnetzwerk, festgelegt, sondern können zur Lösung einer ganzen Klasse von Optimierungsproblemen verwendet werden.

Arbeiten alle Agenten zusammen, um ein verteilt formuliertes Optimierungsproblem zu lösen, wird das Agentensystem als kooperativ bezeichnet. Klassischerweise wird hierbei als Gesamtproblem die Optimierung der Summe der einzelnen Teilkostenfunktionen betrachtet. Bei nicht-kooperativen Problemstellungen konkurrieren die Agenten gegeneinander beispielsweise bezüglich einer gemeinsam verwendeten Ressource, siehe Energiemarktproblem [15][16][17] und Nash-Cournotspiel [23]. Entsprechend optimiert hierbei jeder Agent egoistisch seine Strategie. Dabei sind die Kosten eines Agenten sowohl von der eigenen Strategie als auch von den Strategien der restlichen Agenten abhängig.

Solche Problemstellungen können unter Verwendung der nicht-kooperativen Spieltheorie analysiert werden. Ein mögliches Lösungskonzept ist das Nash-Gleichgewicht, welches einen stabilen Zustand darstellt, in dem alle Agenten das Optimum ihrer lokalen Kosten bezüglich ihrer eigenen Strategie erreichen, d. h. sie finden eine beste Antwort bezüglich ihrer Kosten auf die Strategien der restlichen Spieler.

In vielen Problemstellungen koexistiert allerdings auch kooperatives und nicht-kooperatives Verhalten. Das bedeutet, dass gewisse Untermengen der Agenten zusammenarbeiten, während diese mit wiederum anderen Agenten des Systems in Konkurrenz stehen. Ein Beispiel hierfür sind sogenannte Multiclusterspiele [24][25][26][27], bei denen Gruppen zusammenarbeitender Agenten miteinander in Konkurrenz stehen. Um solche Spiele stabil zu lösen, muss das kooperative Optimum innerhalb der Gruppen gefunden und gleichzeitig ein stabiler Zustand zwischen den nicht-kooperierenden Gruppen erreicht werden. Bisher gibt es allerdings wenige Algorithmen, die beschränkte Multiclusterspiele lösen können. Entsprechend ergibt sich das zweite Ziel der vorliegenden Arbeit:

Ziel 2 (Multiclusterspiele) Es soll die Kombination aus kooperativen und nicht-kooperativen Problemstellungen in Form des Multiclusterspiels betrachtet werden. Als explizite Anwendung soll hierbei eine Problemstellung aus dem Kontext der Stromnetzwerke von Morgen gelöst werden, bei der ein Einsatzplanproblem zwischen und innerhalb von Microgrids für dezentrale Generatoren und Speicher in Form eines Multiclusterspiels gelöst wird.

Viele verteilte Algorithmen zur Lösung von kooperativen [18][28][29] und nicht-kooperativen [4][5][17] Optimierungsproblemen verwenden Konsensusprotokolle zur effizienten Informationsverteilung im Netzwerk. Bei diesen werden die Kommunikationsverbindungen als Graph dargestellt, der wiederum durch eine Kommunikationsmatrix repräsentiert werden kann.

Die Elemente der Kommunikationsmatrix entsprechen dabei den gewichteten Kommunikationsverbindungen des Graphen. Die Konsensusbildung bezüglich einer bestimmten Größe, beispielsweise der Schätzung des optimalen Lösungsvektors, lässt sich damit als wiederholte Multiplikation der Schätzungen mit dieser Matrix darstellen, sodass im Grenzwert der Konsensus erreicht wird. Voraussetzung hierfür ist allerdings eine korrekte Wahl der Gewichtung der Kommunikationsmatrix, da nur dann sicher ein Grenzwert erreicht wird, der aus einer bestimmten Mischung der ursprünglichen Schätzungen besteht. Möglich sind hierbei doppelt-stochastische Gewichtungen, bei denen sich alle Zeilen und Spalten der Matrix zu eins summieren, und reihen- bzw. spalten-stochastische Gewichtungen, bei denen dies nur für die jeweiligen Reihen bzw. Spalten gilt. Handelt es sich um einen bidirektionalen Graphen, bei dem Informationen entlang der Kante in beide Richtungen fließen können, so ist die Generierung der drei genannten Gewichtung unter vergleichbarem Aufwand möglich. Allerdings wird in [30] angeführt, dass bidirektionale Kommunikation eine idealisierte Annahme ist, die in praktischen Problemen oft nicht realisierbar ist. Bei gerichteten Graphen können Informationen nur in eine Richtung entlang einer Kante fließen. Bei diesen ist es deutlich einfacher, nur eine reihen- bzw. spalten-stochastische Gewichtung zu erzeugen¹ [31]. Folglich können Optimierungsalgorithmen, die auch unter Annahme von reihen- bzw. spalten-stochastisch gewichteter Kommunikation konvergieren, auf eine allgemeinere Klasse von Kommunikationsgraphen angewendet werden. Durch diese asymmetrische Gewichtung wird allerdings auch eine Asymmetrie in der Kommunikation verursacht, welche die Beweisführung bezüglich der Konvergenz von Optimierungsalgorithmen erschwert bzw. andere Vorgehensweisen notwendig macht.

Ähnliches gilt für die Zeitvarianz von Graphen. Damit ist gemeint, dass während eines betrachteten Zeitraums Verbindungen zwischen den Knoten des Graphen hinzukommen oder wegfallen können. Hiermit kann beispielsweise eine unzuverlässige Kommunikationsarchitektur modelliert werden. Entsprechend ist ein Algorithmus robuster gegenüber Störungen der Kommunikation, wenn dieser auch im Fall von zeitvarianten Graphen konvergiert. Bezüglich der Kommunikation lässt sich entsprechend das nachfolgende Ziel formulieren.

Ziel 3 (Kommunikation) Die in der vorliegenden Arbeit entwickelten Algorithmen sollen auf einer möglichst großen Klasse von Kommunikations-

¹Für eine ausführliche Diskussion bezüglich der Generierung von Gewichtungen für Kommunikationsmatrizen siehe den Grundlagenabschnitt 2.5.

architekturen anwendbar sein. Entsprechend sind reihen- bzw. spaltenstochastische Gewichtungen sowie eine mögliche Zeitvarianz des Graphen zu bevorzugen.

1.2 Einordnung

Bevor auf die Beiträge der vorliegenden Arbeit eingegangen wird, soll zunächst die behandelte Problemklasse von verteilten Optimierungsproblemen eingeschränkt sowie die verwendeten Lösungsansätze charakterisiert werden. Explizit formuliert sind die Annahmen bezüglich der behandelten Problemklasse in Form von Pauschalannahmen in der jeweiligen Problemformulierung in den Abschnitten 4.2 und 5.2 zu finden.

Alle Algorithmen der vorliegenden Arbeit sind für zeitdiskrete Umgebungen entwickelt. Dabei erfolgt sowohl die Kommunikation als auch die Optimierungsaktualisierungen in zeitdiskreten Schritten. Hierdurch ist eine direkte Anwendung auf einem digitalen System möglich.

Zur Beschreibung der behandelten Problemklasse muss zwischen kooperativen und nicht-kooperativen Problemstellungen unterschieden werden. Bezüglich ersteren werden ausschließlich konvexe und kontinuierlich differenzierbare Zielfunktionen betrachtet. Die Beschränkungsmengen sind ebenfalls konvex und liegen, je nach verwendetem Lösungsalgorithmus, als explizite Beschränkungsfunktionen oder generische Menge vor. Diese Annahmen stellen valide Einschränkung der betrachteten Problemklasse dar, da viele relevante verteilte Problemstellungen, unter anderem das in der vorliegenden Arbeit betrachtete energietechnische Problem, konvex sind [14]. Bezüglich nicht-kooperativer Problemstellungen bzw. der Multi-clusterspiele werden konvexe Spiele betrachtet. Das bedeutet, dass die Kostenfunktion jedes Spielers konvex und kontinuierlich differenzierbar bezüglich der eigenen Strategie ist und zusätzlich auch die Nebenbedingungsmengen konvex sind. Darüber hinaus wird vorausgesetzt, dass die jeweilige Beschränkungsmenge der Spieler unabhängig von den Strategien der restlichen Spieler ist und dass ein Nash-Gleichgewicht als stabile Lösung des Spiels existiert. In diesem Zusammenhang wird eine gleichmäßig Monotonie des Spiel mappings vorausgesetzt².

²Die Existenz eines Nash-Gleichgewicht kann anhand des Spiel mappings nachgewiesen werden, siehe hierzu den Grundlagenabschnitt 2.4.

Bei allen in der vorliegenden Arbeit vorgestellten Algorithmen verwenden die Agenten Gradienteninformationen bezüglich ihrer Kostenfunktion, um das Optimum der Problemstellung zu identifizieren. Der Informationsaustausch wird dabei durch ein Konsensusprotokoll gesteuert, mit dem es den Agenten bei korrekter Gewichtung der Kommunikation möglich wird, sich auf eine gemeinsame Strategie zu einigen.

1.3 Beiträge der Arbeit

Die Erweiterungen des Standes der Forschung sind in sechs Beiträge aufgeteilt. Dabei beschäftigen sich die Beiträge 1 bis 4 mit kooperativen Problemstellungen, während bei den Beiträgen 5 und 6 Multiclusterspiele und damit auch nicht-kooperative Spiele betrachtet werden. Für eine detaillierte Einordnung der Beiträge in relevante Literatur siehe die Abschnitte 4.3 und 5.3.

Beitrag 1 besteht aus einem Algorithmus, bei dem ein Optimierungsverfahren zur Lösung von unbeschränkten Problem mit einem Straftermansatz erweitert wird, sodass Nebenbedingungen in Form von expliziten Ungleichungen berücksichtigt werden können, auf die darüber hinaus nur einzelne Agenten Zugriff haben. Der Basisalgorithmus aus [18] sowie dessen durch die Strafterme erweiterte Form benötigen hierbei eine absinkende Schrittweitenfolge in der Aktualisierungsvorschrift zur Verwendung der lokalen Gradienteninformation. Für den Konvergenzbeweis werden spalten-stochastisch gewichtete Kommunikationsmatrizen und zeitvariante Graphen angenommen. Damit erweitert der Algorithmus dieses Beitrages den Stand der Forschung bezüglich der Berücksichtigung von Nebenbedingungen, die Zugriffsbeschränkungen unterliegen, in Kombination mit allgemeineren Kommunikationsannahmen. Entsprechend werden die Ziele 1 und 3 verfolgt. Veröffentlicht ist dieser Algorithmus in der Arbeit [32], bei welcher der Autor der vorliegenden Arbeit Zweitautor ist.

In **Beitrag 2** wird der gleiche Basisalgorithmus wie in Beitrag 1 zur Berücksichtigung von Nebenbedingungen erweitert. Hierfür wird sowohl ein Lagrangemultiplikatoransatz, als auch eine Projektion verwendet. Durch die Verwendung dieser zwei Methoden zur Einbindung von Nebenbedingungen wird es möglich, explizit die Verfügbarkeit von Informationen bezüglich der Nebenbedingungen zu auszunutzen und so

das Konvergenzverhalten potentiell zu verbessern. Die Anforderungen bezüglich der Kommunikationsarchitektur sind die gleichen wie für Beitrag 1. Entsprechend wird auch bei diesem Beitrag die Forschungsfront bezüglich der Einbeziehung von Nebenbedingungen mit expliziten Informationsbeschränkungen bei allgemeiner Kommunikation erweitert und somit die Ziele 1 und 3 verfolgt. Im Vergleich zu dem Algorithmus aus Beitrag 1 kann die Konvergenzzeit sowie der Aufwand zur Festlegung von Optimierungsparametern vor der eigentlich Laufzeit des Algorithmus reduziert werden. Veröffentlicht ist dieser Beitrag in [33].

Dem Algorithmus des **Beitrags 3** liegt im Vergleich zu den vorherigen Beiträgen ein anderes, unbeschränktes Verfahren zugrunde. Dieses erlaubt es, konstante Schrittweiten zu verwenden, was die Konvergenzgeschwindigkeit merklich verbessert. Nebenbedingungen werden hierbei über eine Projektion eingebunden. Im Gegensatz zu den ersten beiden Beiträgen sind die Kommunikationsanforderungen dabei allerdings notwendigerweise weniger allgemein, da von statischen Graphen mit doppelt-stochastisch gewichteten Verbindungen ausgegangen wird. Entsprechend liegt hier der Fokus auf die Verfolgung des ersten Ziels. Im Vergleich zu dem Stand der Forschung bezüglich Algorithmen mit konstanten Schrittweiten wird eine einfachere Struktur und eine freiere Wahl der Schrittweite ermöglicht. Der Algorithmus dieses Beitrages stellt eine Sonderform des Beitrages 5 für rein kooperative Problemstellungen dar. Die Konvergenzargumentation erfolgt in der vorliegenden Arbeit.

Im Rahmen des **Beitrages 4** werden die Algorithmen aus den Beiträgen 1 bis 3 auf eine kooperative Einsatzplanproblemstellung innerhalb von Microgrids angewendet und das Konvergenzverhalten analysiert.

In **Beitrag 5** wird ein Algorithmus vorgestellt, mit dem Multiclusterspiele gelöst werden können. Der Algorithmus besteht hierbei aus einem auf die Problemstellung angepassten, gradientenbasierten Basisverfahren, das zur Berücksichtigung von Nebenbedingungen mit einer Projektion erweitert wird. Entsprechend werden in diesem Beitrag die Ziele 1 und 2 verfolgt. Der bestehende Stand der Forschung wird hierbei durch die Berücksichtigung von Nebenbedingungen in Multiclusterspielen bei weniger stark eingeschränkter Kommunikation zwischen den Clustern erweitert. Die Konvergenzrate dieses Algorithmus ist nachweislich linear. Beschrieben ist der Algorithmus mit Konvergenzbeweis in [34].

In **Beitrag 6** wird der Algorithmus aus Beitrag 5 auf eine erweiterte Einsatzplanproblemstellung, formuliert als Multiclusterspiel, angewendet und das Konvergenzverhalten analysiert.

Weitere Veröffentlichungen In diesem Paragraphen werden kurz weitere Veröffentlichungen des Autors der vorliegenden Arbeit beschrieben, die allerdings nicht explizit in der vorliegenden Arbeit behandelt werden.

In [35] wird der Algorithmus aus Beitrag 1 weiterführend in Anwendung auf eine spezielle Problemstellung analysiert. Im Vordergrund steht hierbei eine optimale Wahl der Parameter des Algorithmus, mit denen eine möglichst kurze Konvergenzzeit erzielt werden kann.

In der Arbeit [36], bei welcher der Autor der vorliegenden Arbeit Zweitautor ist, wird ein payoffbasierter Algorithmus zur Lösung von Multiclusterspielen vorgestellt, bei dem den Agenten keine Gradienteninformationen zu Verfügung stehen. Hierbei erfolgt die Aktualisierung der Optimierungsvariablen unter Schätzung der Gradienten durch Abfrage des aktuellen Wertes der Kostenfunktion von einem Orakel.

In [37], wo der Autor der vorliegenden Arbeit als Zweitautor mitgewirkt hat, werden zentrale, gradienten- und payoffbasierte Verfahren zur Lösung von Multiclusterspielen miteinander bezüglich ihrer Anforderungen und Konvergenzgeschwindigkeiten in Anwendung auf ein erweitertes Nash-Cournotspiel verglichen.

In [38] wird ein Algorithmus präsentiert, mit dem unbeschränkte Multiclusterspiele unter allgemeinen Kommunikationsannahmen gelöst werden können. Die grundlegende Struktur des Algorithmus weist dabei eine gewisse Übereinstimmung mit dem Algorithmus aus Beitrag 5 auf, weswegen auf eine explizite Diskussion der Veröffentlichung in der vorliegenden Arbeit verzichtet wird.

Die Veröffentlichung [39] beschäftigt sich mit der weiterführenden Analyse des Algorithmus aus der Veröffentlichung [38] und vergleicht dessen Konvergenzzeit mit einem weiteren Algorithmus, der für die gleiche Problemklasse entworfen ist.

Wissenschaftliche Vorträge Die genannten Beiträge und Veröffentlichungen wurden im Rahmen von Vorträgen auf wissenschaftlichen Konferenzen [40][41][42][43] und Fachausschüssen der Gesellschaft für Mess- und Automatisierungstechnik (GMA) bezüglich Grundlagen verteilter Netze [44][45] präsentiert.

1.4 Struktur der Arbeit

In der Abbildung 1.1 ist die logische Struktur der Arbeit dargestellt. In den Hauptkapiteln 4 und 5 sind zudem die Beiträge sowie die dazugehörigen Veröffentlichungen vermerkt.

Die Arbeit beginnt mit der Einleitung, welche das Simulationskapitel 3 sowie die Hauptkapitel 4 und 5 motiviert. In Kapitel 2 werden relevante Grundlagen für die Hauptkapitel vermittelt, während in Kapitel 3 das in der vorliegenden Arbeit verwendete Simulationsmodell vorgestellt wird. Bei diesem Modell handelt es sich um ein Einsatzplanproblem zwischen Microgrids, das als Multiclusterspiel formuliert wird. Es schließen sich die beiden Hauptkapitel 4 und 5 der Arbeit an. Diese sind strukturell gleich aufgebaut: Nach einleitenden Bemerkungen bezüglich der behandelten Problemklasse erfolgt zunächst eine explizite Formulierung des zu lösenden Problems. Anschließend wird detailliert auf den Stand der Forschung eingegangen und wie sich die Beiträge in diesen eingliedern. Es folgt die Vorstellung der Algorithmen, deren Formulierung und Konvergenzbeweis die Hauptbeiträge der vorliegenden Arbeit darstellen. Abgeschlossen werden die Hauptkapitel mit einer Diskussion von Simulationsergebnissen der Algorithmen in Anwendung auf ein entsprechendes Szenario des in Kapitel 3 vorgestellten Modells. Im abschließenden Fazit werden alle Ergebnisse der Arbeit zusammengefasst und ein Ausblick über mögliche zukünftige Forschungsaktivitäten gegeben.

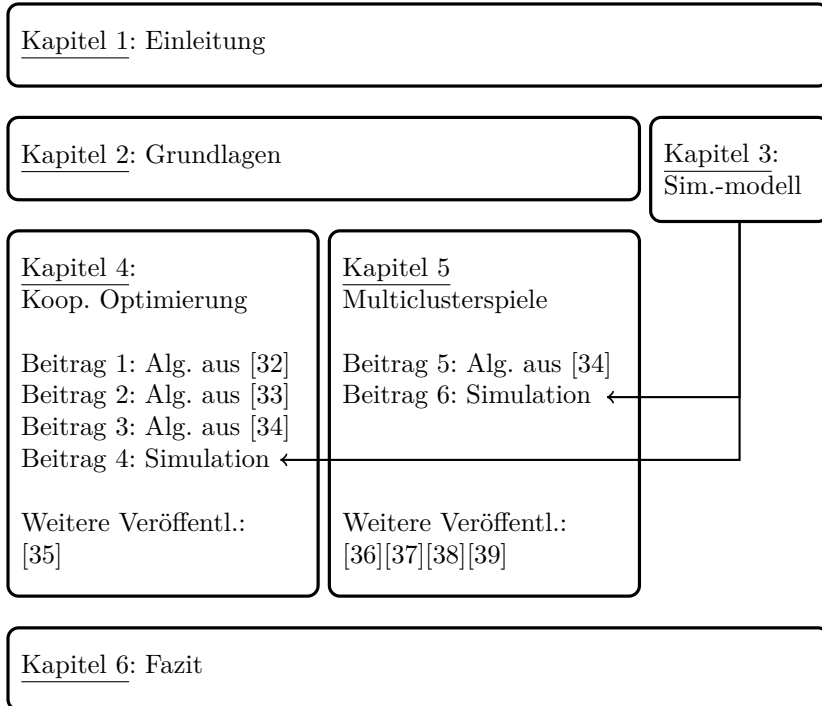


Abbildung 1.1: Logische Struktur der Arbeit mit Beiträgen und Veröffentlichungen.

2 Grundlagen

Dieses Kapitel dient der Vermittlung von Grundlagen, die für das Verständnis des Hauptteils notwendig sind. Dabei werden nicht einzelne Teilgebiete, wie beispielsweise die konvexe Optimierung, allumfassend behandelt, sondern nur die für diese Arbeit relevanten Aspekte darlegt und erklärt. Für weiterführende Informationen sei auf die jeweils angegebenen Quellen verwiesen.

2.1 Zentrale, dezentrale und verteilte Entscheidungsfindung

Es sei ein Agentensystem gegeben, in dem jeweils einzelne Agenten bestimmte Informationen besitzen, über welche die restlichen Agenten nicht verfügen. Ob es sich bei dem System um ein zentrales, dezentrales oder verteiltes System handelt, hängt davon ab, welche Entität im Netzwerk Entscheidungen bezüglich einer zu lösenden Problemstellung fällt und wie die Agenten untereinander vernetzt sind, siehe hierfür auch die Abbildung 2.1. Die nachfolgende Begriffsdefinition für zentrale, dezentrale und verteilte Systeme stützt sich hauptsächlich auf die Arbeit [46].

Bei zentralen Systemen sind alle Agenten mit einer zentralen Berechnungseinheit verbunden. Alle Agenten schicken ihre lokalen Informationen an diese Einheit, bei der dann die Weiterverarbeitung erfolgt und eine Entscheidung getroffen wird. So entsteht eine Hierarchie im System: Während die Agenten keine eigenständigen Entscheidungen treffen können und nur ihre Informationen zu Verfügung stellen, verfügt die zentrale Einheit über die gesamten Informationen und trifft auf Basis dieser Entscheidungen für das gesamte Netzwerk. In einem zentralen System ist der zentrale Knoten im Allgemeinen ein sogenannter Single-Point-Of-Failure: Wird dieser ausgeschaltet, so bricht das gesamte Netzwerk zusammen und die gesetzte Aufgabe kann nicht mehr erfüllt werden.

Bei dezentralen Systemen wird eine weitere, mittlere Hierarchieebene eingeführt. Die Knoten der untersten Hierarchieschicht senden ihre Informationen an Knoten der mittleren Hierarchieebene, welche die Informati-

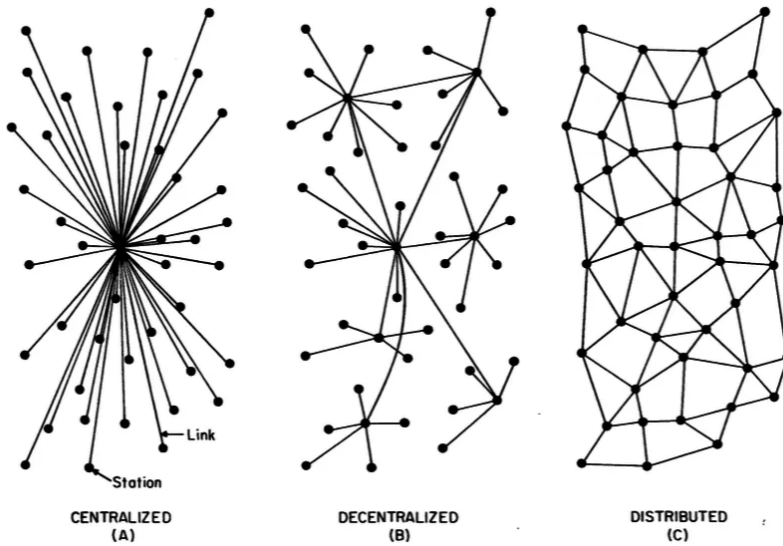


Abbildung 2.1: Darstellung von zentral, dezentral und verteilt organisierten Agentensysteme nach [46].

on sammeln und untereinander verarbeiten. Oft werden auch Systeme als dezentral bezeichnet, bei denen die Kommunikation über einen zentralen Knoten bzw. einige wenige Knoten abläuft, die Entscheidungsfindung allerdings ausschließlich an den lokalen und damit nicht-zentralen/dezentralen Knoten stattfindet.

Vollständig verteilte Systeme besitzen keine Hierarchie. Die Knoten des Netzwerkes kommunizieren mit ihren nächsten Nachbarn, wobei die Berechnungen ausschließlich an den lokalen Knoten erfolgen. In verteilten Netzwerken existiert aufgrund der verteilten Information und Bearbeitung kein Single-Point-of-Failure. Zusätzlich kann dadurch, dass ein Agent im Allgemeinen nicht nur mit einem einzigen Nachbarn kommuniziert, Redundanz in der Kommunikation erzeugt werden.

2.2 Mathematische Grundlagen

Dieser Abschnitt dient zur Einführung mathematischer Grundlagen zu Eigenschaften von Abbildungen und Mengen, die für diese Arbeit von Bedeutung sind.

2.2.1 Reelle Abbildungen

Sei zunächst die Funktion $f : \mathbb{R}^d \rightarrow \mathbb{R}$ gegeben, die den d -dimensionalen Raum der reellen Zahlen auf ein reelles Skalar abbildet. Für diese werden nachfolgend die Eigenschaften pseudo-konvex, konvex und gleichmäßig konvex definiert, die den Grundlagenbüchern [47] und [48] entnommen sind.

Definition 2.2.1 (Pseudo Konvexität) *Die differenzierbare Funktion $f(x)$ wird als pseudo-konvex bezeichnet, wenn*

$$\nabla f(y)^T(x - y) \geq 0 \implies f(x) \geq f(y) \quad (2.1)$$

für alle $x, y \in \mathbb{R}^d$.

Definition 2.2.2 (Konvexität) *Die Funktion $f(x)$ wird als konvex bezeichnet, wenn für $x, y \in \mathbb{R}^d$ und ein $\lambda \in [0, 1]$ die Ungleichung*

$$f((1 - \lambda)x + \lambda y) \leq (1 - \lambda)f(x) + \lambda f(y) \quad (2.2)$$

gilt. Ist die Funktion $f(x)$ kontinuierlich differenzierbar, dann ist die Ungleichung (2.2) äquivalent zu

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) \text{ bzw.} \quad (2.3)$$

$$(\nabla f(x) - \nabla f(y))^T(x - y) \geq 0. \quad (2.4)$$

Anschaulich bedeutet die Ungleichung (2.2), dass der Graph der Funktion $f(x)$ unterhalb der Verbindungslinie der Punkte $(x, f(x))$ und $(y, f(y))$ liegt. Als konkav wird die Funktion $f(x)$ bezeichnet, wenn $-f(x)$ konvex ist. Sind die Ungleichungen strikt für $x \neq y$, so wird von einer strikt konvexen Funktion gesprochen.

Definition 2.2.3 (Gleichmäßige Konvexität) *Die Funktion $f(x)$ wird als gleichmäßig (engl.: strongly) konvex bezeichnet, wenn ein $\mu > 0$ existiert, sodass für beliebige $x, y \in \mathbb{R}^d$*

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{1}{2}\mu \|y - x\|^2 \text{ bzw.} \quad (2.5)$$

$$(\nabla f(x) - \nabla f(y))^T(x - y) \geq \mu \|x - y\|^2. \quad (2.6)$$

Bezüglich der Konvexität einer Funktion $f(x)$ gelten nachweislich die Implikationen

$$\begin{aligned} \text{Gleichm. Konvexität} &\implies \text{strikte Konvexität} \implies \text{Konvexität} \\ &\implies \text{Pseudo Konvexität.} \end{aligned}$$

Die Umkehrungen gelten im Allgemeinen nicht.

Nun wird die Abbildung $F : \mathbb{R}^d \rightarrow \mathbb{R}^d$ betrachtet und Begrifflichkeiten zur Monotonie bzw. die Lipschitzstetigkeit der Funktion definiert.

Definition 2.2.4 (Monotonie) *Die Abbildung $F(x)$ wird als monoton bezeichnet, wenn für beliebige $x, y \in \mathbb{R}^d$*

$$(x - y)^T (F(x) - F(y)) \geq 0 \quad (2.7)$$

gilt.

Äquivalent zur Definition der Konvexität, wird von *striker Monotonie* gesprochen, wenn die Ungleichung der Definition strikt ist.

Definition 2.2.5 (Gleichmäßige Monotonie) *Die Abbildung $F(x)$ wird als gleichmäßig monoton bezeichnet, wenn ein $\mu > 0$ existiert, sodass für beliebige $x, y \in \mathbb{R}^d$*

$$(x - y)^T (F(x) - F(y)) \geq \mu \|x - y\|^2 \quad (2.8)$$

gilt.

Für die Monotonie der Abbildung $F(x)$ gelten die Implikationen

$$\text{Gleichm. Monotonie} \implies \text{strikte Monotonie} \implies \text{Monotonie,}$$

wobei auch hier die Umkehrungen im Allgemeinen nicht gelten.

Definition 2.2.6 (Lipschitzstetigkeit) *Die Abbildung $F(x)$ ist lipschitzstetig mit der Konstanten $L > 0$, wenn für alle $x, y \in \mathbb{R}^d$*

$$\|F(x) - F(y)\| \leq L \|x - y\| \quad (2.9)$$

gilt.

2.2.2 Mengen

Gegeben sei die nichtleere Menge $\mathcal{X} \subset \mathbb{R}^d$. Der Vektor x ist ein innerer Punkt der Menge \mathcal{X} , wenn ein $\epsilon > 0$ existiert, sodass die Kugel $B_\epsilon(x)$ um x mit Radius ϵ ein Teil der Menge \mathcal{X} ist, d. h. $B_\epsilon(x) \subseteq \mathcal{X}$. Die Menge von inneren Punkten von \mathcal{X} wird als der Innenraum von \mathcal{X} betitelt und mit $\text{int}(\mathcal{X})$ bezeichnet [48]. Die nachfolgenden Definitionen sind aus den Grundlagenbüchern [48] und [49] entnommen.

Definition 2.2.7 (Offene und geschlossene Mengen) *Die Menge \mathcal{X} wird als offen bezeichnet, wenn jeder Punkt $x \in \mathcal{X}$ ein innerer Punkt von \mathcal{X} ist, d. h. $x \in \text{int}(\mathcal{X})$.*

Die Menge \mathcal{X} wird als geschlossen bezeichnet, wenn alle Endpunkte der Menge Teil der Menge sind, d. h., wenn für jede konvergierende Teilfolge $\{x_s\}_{s \in \mathbb{N}} \subset \mathcal{X}$ der Grenzwert $\lim_{s \rightarrow \infty} x_s = x \in \mathcal{X}$ gilt.

Alternativ kann für die Definition von offenen und geschlossenen Mengen der Rand betrachtet werden: Sei $\delta\mathcal{X} = \mathcal{X}/\text{int}(\mathcal{X})$ der Rand der Menge \mathcal{X} , d. h. für ein beliebig kleines $\epsilon > 0$ existieren Vektoren $u \in \mathcal{X}$, $v \notin \mathcal{X}$, sodass $\|x - u\| \leq \epsilon$ und $\|x - v\| \leq \epsilon$ für $x \in \delta\mathcal{X}$. Dann ist die Menge geschlossen, wenn $\delta\mathcal{X} \in \mathcal{X}$ und offen, wenn $\delta\mathcal{X} \cap \mathcal{X} = \emptyset$.

Definition 2.2.8 (Beschränkte Mengen) *Die Menge \mathcal{X} ist beschränkt, wenn Vektoren $u, v \in \mathbb{R}^d$ existieren, sodass $v \leq x \leq u$ für alle $x \in \mathcal{X}$. v und u werden als untere und obere Schranke der Menge bezeichnet.*

Definition 2.2.9 (Kompakte Mengen) *Die Menge \mathcal{X} ist kompakt, wenn sie geschlossen und beschränkt ist.*

Definition 2.2.10 (Konvexe Mengen) *Die Menge \mathcal{X} ist konvex, wenn für jedes Paar $x, y \in \mathcal{X}$ die Verbindungslinie $(1 - \lambda)x + \lambda y$ für $\lambda \in [0, 1]$ innerhalb von \mathcal{X} liegt.*

2.3 Lösung konvexer Optimierungsprobleme durch Gradientenabstieg

In diesem Abschnitt werden konvexe Optimierungsprobleme und deren Lösungen durch verschiedene zentrale Lösungsstrategien betrachtet. Zusätzlich werden Bedingungen angeführt, mit denen die Optimalität eines potentiellen Lösungsvektors überprüft werden kann. Zunächst werden hierfür zwei Problemstellungen formuliert, die sich bezüglich der

Struktur ihrer Nebenbedingungen unterscheiden.

Das erste Problem ist ein Minimierungsproblem bezüglich der konvexen Funktion $F : \mathbb{R}^d \rightarrow \mathbb{R}$, bei dem sich alle Lösungen in einer Beschränkungsmenge $\mathcal{X} \subset \mathbb{R}^d$ befinden:

Problem 1

$$\min_x F(x), \tag{2.10a}$$

$$\text{s.t. } x \in \mathcal{X}. \tag{2.10b}$$

Eine Lösung x^* dieses Problems kann mit Hilfe der nachfolgenden Bedingung, zitiert aus [50], auf Optimalität überprüft werden:

Optimalitätsbedingung 1 (Notwendig und hinreichend) *Die Beschränkungsmenge \mathcal{X} sei konvex und die Funktion $F(x)$ sei konvex auf \mathcal{X} . Der Vektor x^* ist dann und nur dann ein Optimum des Problems 1, wenn*

$$\nabla F(x^*)^T (x - x^*) \geq 0, \forall x \in \mathcal{X}. \tag{2.11}$$

Auch im zweiten Problem wird die Funktion $F(x)$ minimiert und eine Beschränkungsmenge \mathcal{X} betrachtet. Allerdings ist die Optimierungsvariable x zusätzlich durch M Ungleichungen beschränkt:

Problem 2

$$\min_x F(x), \tag{2.12a}$$

$$\text{s.t. } g(x) \leq 0, \tag{2.12b}$$

$$x \in \mathcal{X}, \tag{2.12c}$$

wobei $g(x) = \text{col}(g_1(x), \dots, g_i(x), \dots, g_M(x))$ mit den Funktionen $g_i : \mathbb{R}^d \rightarrow \mathbb{R}$, $\forall i \in \{1, \dots, M\}$, womit $g : \mathbb{R}^d \rightarrow \mathbb{R}^M$. Nach möglicher Absorption der Ungleichungsbedingungen in die Menge \mathcal{X} gilt die Optimalitätsbedingung 1 auch für diese Problemstellung. Die Beschreibung eines Teils der Nebenbedingungen als Ungleichung ermöglicht die Verwendung einer größeren Klasse von Analyseverfahren bzw. Lösungsverfahren.

Ist in einem unbeschränkten Optimierungsproblem die konvexe Kostenfunktion $F(x)$ differenzierbar, können gradientenbasierte Verfahren verwendet werden, um Lösungen zu identifizieren. Das klassische Gradientenabstiegsverfahren ist eine Methode, mit der iterativ die Lösung unter

Verwendung des Gradienten der Kostenfunktion ermittelt werden kann. Dafür wird in jeder Iteration k der Gradient der Kostenfunktion am aktuellen Wert der Optimierungsvariablen $x(k)$ ausgewertet und die Optimierungsvariable in Richtung des negativen Gradienten, skaliert über die möglicherweise iterationsabhängige Schrittweite α_k , aktualisiert. Es ergibt sich das bekannte Schema

$$x(k+1) = x(k) - \alpha_k \nabla F(x(k)). \quad (2.13)$$

Im Folgenden wird eine Auswahl von Möglichkeiten vorgestellt, mit denen Nebenbedingungen wie solche der Probleme 1 und 2 in den Gradientenabstieg integriert werden können.

Einbeziehung von Nebenbedingungen in den Gradientenabstieg

Für den Hauptteil der vorliegenden Arbeit sind Projektionen auf die Beschränkungsmengen, Duale Ansätze über die Lagrangefunktion sowie Straftermansätze von besonderer Bedeutung und werden im Folgenden kurz vorgestellt.

Projektion

Die Projektion eines Vektors $u \in \mathbb{R}^d$ auf eine konvexe und kompakte Menge $\mathcal{X} \subset \mathbb{R}^d$ ist definiert durch den Operator

$$\text{proj}_{\mathcal{X}}[u] = \underset{x \in \mathcal{X}}{\text{argmin}} \|x - u\|^2. \quad (2.14)$$

Entsprechend ist die Projektion eines Vektors u auf die Menge \mathcal{X} die Lösung eines Optimierungsproblems, bei dem derjenige Vektor $x \in \mathcal{X}$ gesucht wird, der den geringsten euklidischen Abstand zu dem Vektor u besitzt. Veranschaulicht wird dies beispielhaft in Abbildung 2.2. Ist \mathcal{X} konvex und kompakt, existiert aufgrund der gleichmäßigen Konvexität der Kostenfunktion der Projektion immer eine eindeutige Lösung.

Eine wichtige Eigenschaft des Projektionsoperators ist die Nicht-Ausdehnungseigenschaft (engl.: non-expansiveness): Sei $\mathcal{X} \subset \mathbb{R}^d$ eine konvexe und kompakte Menge. Dann gilt

$$\|\text{proj}_{\mathcal{X}}[x] - \text{proj}_{\mathcal{X}}[y]\| \leq \|x - y\| \quad (2.15)$$

für beliebige $x, y \in \mathbb{R}^d$, siehe beispielsweise [50, Proposition 1.1.4].

Der mathematische Aufwand zur Lösung des Optimierungsproblems der

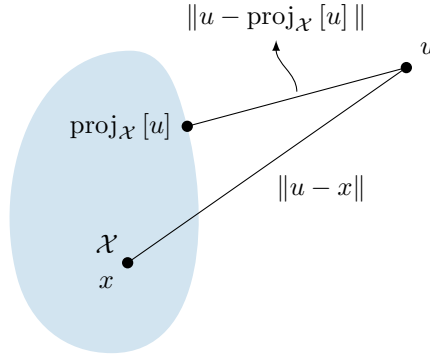


Abbildung 2.2: Beispiel für die Projektion des Vektors u auf die Menge \mathcal{X} nach Gleichung (2.14). Es gilt, dass $\|u - \text{proj}_{\mathcal{X}}[u]\| \leq \|u - x\|$ für alle $x \in \mathcal{X}$.

Projektion aus Gleichung 2.14 ist dabei stark abhängig von der Beschränkungsmenge \mathcal{X} , auf die projiziert wird. Besonders wenig Aufwand ist bei solchen Mengen notwendig, die durch lineare Gleichungen oder Ungleichungen beschrieben werden können. In diesen Fällen liegt ein quadratisches Programm vor, das sich effizient lösen lässt.

Wie in [51, Abschnitt 8.1.1] beschrieben, gibt es in den folgenden Sonderfällen für Beschränkungsmengen sogar analytische Lösungen:

- Hyperebene $\mathcal{X}^{\text{H}} = \{x | a^T x = b\}$:

$$\text{proj}_{\mathcal{X}^{\text{H}}}[u] = \frac{u + (b + a^T u)a}{\|a\|_2^2}. \quad (2.16)$$

- Halbräume $\mathcal{X}^{\text{Hr}} = \{x | a^T x \leq b\}$:

$$\text{proj}_{\mathcal{X}^{\text{Hr}}}[u] = \begin{cases} \frac{u + (b + a^T u)a}{\|a\|_2^2} & a^T u > b, \\ u & a^T u \leq b. \end{cases} \quad (2.17)$$

- Rechtecke $\mathcal{X}^{\text{R}} = \{x | \underline{x} \leq x \leq \bar{x}\}$:

$$\text{proj}_{\mathcal{X}^{\text{R}}}[u]_i = \begin{cases} \underline{x}_i & u_i \leq \underline{x}_i, \\ u_i & \underline{x}_i \leq u_i \leq \bar{x}_i, \\ \bar{x}_i & u_i \geq \bar{x}_i. \end{cases} \quad (2.18)$$

Angewendet auf ein Optimierungsproblem der Form von Problem 1 kann die Projektion dazu verwendet werden, den Gradientenschritt aus Gleichung (2.13) wieder zurück in die Beschränkungsmenge \mathcal{X} zu projizieren, insofern der Gradientenschritt die Optimierungsvariable aus der Beschränkungsmenge hinaus schiebt. Entsprechend ergibt sich die Aktualisierungsvorschrift

$$x(k+1) = \text{proj}_{\mathcal{X}} [x(k) - \alpha_k \nabla F(x(k))]. \quad (2.19)$$

In Gleichung (2.11) ist eine notwendige und hinreichende Optimalitätsbedingung gegeben, mit der überprüft werden kann, ob es sich bei einem Vektor x^* um ein Optimum von Problem 1 handelt. Unter Verwendung des Projektionsoperators lässt sich eine äquivalente Bedingung ableiten, die für Konvergenznachweise von projizierten Gradientenverfahren hilfreich ist:

Optimalitätsbedingung 2 (Notwendig und hinreichend) *Die Funktion $F(x)$ aus Problem 1 sei konvex und differenzierbar, die Menge \mathcal{X} sei geschlossen und konvex. Der Vektor x^* ist dann und nur dann ein Optimum des Problems, wenn für beliebige $\alpha > 0$*

$$x^* = \text{proj}_{\mathcal{X}} [x^* - \alpha \nabla F(x^*)] \quad (2.20)$$

gilt. Folglich muss x^ ein Fixpunkt der Abbildung $H(x) = \text{proj}_{\mathcal{X}} [x - \alpha \nabla F(x)]$ mit $\alpha > 0$ sein.*

Die Äquivalenz zu Bedingung in (2.11) lässt sich unter Verwendung des Projektionstheorems [50, Proposition 1.1.4] herleiten. Demnach ist $x^* \in \mathcal{X}$ dann und nur dann ein Fixpunkt der Abbildung $H(x)$, wenn $(x - x^*)^T (x^* - \nabla F(x^*) - x^*) \leq 0$ und damit $\nabla F(x^*)^T (x - x^*) \geq 0$ für alle $x \in \mathcal{X}$ gilt, was der Gleichung (2.11) entspricht.

Duales Problem und KKT Bedingungen

Eine weitere Möglichkeit zur Einbeziehung von Nebenbedingung in das Gradientenabstiegsverfahren stellt die Betrachtung bzw. Lösung des sogenannten dualen Problems dar. Potentielle Lösungsvektoren können hierbei über die bekannten Karush-Kuhn-Tucker (KKT)-Bedingungen, siehe beispielsweise [51, Abschnitt 5.5.3], auf Optimalität überprüft werden. Dafür wird das Optimierungsproblem 2 mit Ungleichungsnebenbedingungen³ zunächst für den Fall $\mathcal{X} \equiv \mathbb{R}^d$ betrachtet. Ausgangspunkt ist das Hinzufügen

³Die nachfolgende Betrachtung auch auf die Berücksichtigung von Gleichungsnebenbedingungen angewendet werden, siehe hierzu die entsprechenden Quellenverweise.

der Nebenbedingungen zur Zielfunktion über sogenannte Lagrangemultiplikatoren $\mu \in \mathbb{R}^M$. Dadurch ergibt sich die Lagrangefunktion

$$\mathcal{L}(x, \mu) = F(x) + \mu^T g(x). \quad (2.21)$$

Unter deren Verwendung kann die duale Zielfunktion

$$G(\mu) = \inf_x \mathcal{L}(x, \mu) \quad (2.22)$$

des Problems 2 formuliert werden [51, Abschnitt 5.1.2]. Diese Funktion ist immer konkav, auch wenn das ursprüngliche Problem nicht konvex ist. Der obigen Formulierung entsprechend, stellt die duale Funktion $G(\mu)$ eine untere Schranke bezüglich des optimalen Wertes F^* des Problems 2 dar, d. h. $G(\mu) \leq F^*$ für $\mu \geq 0$. Wird daher die duale Funktion bezüglich der Lagrangemultiplikatoren μ maximiert, kann somit die beste untere Schranke gefunden werden. Das entstehende Optimierungsproblem

$$\max_{\mu} G(\mu), \quad (2.23)$$

$$\text{s.t. } \mu \geq 0, \quad (2.24)$$

wird als das duale Problem bezüglich des primalen Problems 2 bezeichnet. Entspricht die optimale Lösung G^* des dualen Problems der Lösung F^* des primalen Problems, wird von *starker Dualität* gesprochen. Diese gilt selbst für konvexe Probleme nur dann, wenn das Problem bzw. dessen Nebenbedingungen zusätzlich sogenannte Constraint Qualifications erfüllt. Eine der bekanntesten Qualifications ist die Slater Bedingung:

Satz 2.3.1 Slater Bedingung für Problem 2:

Die Zielfunktion $F(x)$, die Nebenbedingungsfunktionen $g_i(x)$, $\forall i \in \{1, \dots, M\}$, und die Beschränkungsmenge \mathcal{X} des Problems 2 seien konvex und es existiere ein innerer Punkt $\tilde{x} \in \mathcal{X}$, sodass $g_i(\tilde{x}) < 0$, für alle $i \in \{1, \dots, M\}$. Dann gilt starke Dualität für das Problem 2 und der Punkt \tilde{x} wird als Slatervektor bezeichnet.

Eine allgemeine Formulierung der Slater Bedingung, von der die obige, auf das betrachtete Problem angepasste Bedingung abgeleitet ist, ist beispielsweise in [51, Abschnitt 5.2.3] zu finden.

Ziel eines entsprechenden Optimierungsverfahrens ist es nun, ein sogenanntes *primal-dual optimales Paar* zu identifizieren, das wie folgt definiert ist:

Definition 2.3.1 (Primal-dual optimales Paar) *Es gelte starke Dualität. Es sei μ^* die Lösung des dualen Problems (2.23), sodass $G^* = G(\mu^*)$, zugehörig zur primalen Lösung x^* , sodass $F^* = F(x^*)$. Dann bilden die Vektoren (x^*, μ^*) ein primal-dual optimales Paar.*

Im Folgenden werden Optimalitätsbedingungen betrachtet, mit denen festgestellt werden kann, ob es sich bei einem potentiellen Lösungsvektorpaar um ein primal-dual optimales Paar handelt. Diese Prüfung kann effizient unter Verwendung der bereits genannten KKT Bedingungen erfolgen:

Optimalitätsbedingung 3 (Notwendig und hinreichende KKT Bedingungen, [51] Abschnitt 5.5.3) *Das Problem 2 mit $\mathcal{X} \equiv \mathbb{R}^d$ sei konvex. Es gelte starke Dualität für das Problem, sodass die Dualitätslücke $F^* - G^*$ gleich null ist. Der primale Vektor x^* und die duale Vektor μ^* sind ein primal-dual optimales Paar dann und nur dann, wenn sie die folgenden Bedingungen erfüllen:*

$$\nabla F(x^*) + (\mu^*)^T \nabla g(x^*) = 0 : \text{Lagrangeoptimalität}, \quad (2.25a)$$

$$g(x^*) \leq 0 : \text{Primale Lösbarkeit}, \quad (2.25b)$$

$$\mu^* \geq 0 : \text{Duale Lösbarkeit}, \quad (2.25c)$$

$$(\mu^*)^T g(x^*) = 0 : \text{Komplementärer Schlupf}. \quad (2.25d)$$

Erst die starke Dualität stellt sicher, dass die obige Bedingung nicht nur notwendig, sondern auch hinreichend ist.

Sei \mathcal{X} nun nicht mehr $\mathcal{X} \equiv \mathbb{R}^d$, sondern eine beschränkte, konvexe Untermenge von \mathbb{R}^d . In diesem Fall muss die obige Optimalitätsbedingung angepasst werden, da die Bedingung der Lagrangeoptimalität nun nicht länger gilt. Nach [50, Proposition 6.1.5] kann in diesem Fall die folgende, angepasste Optimalitätsbedingung verwendet werden:

Optimalitätsbedingung 4 (Notwendig und hinreichend, Proposition 6.1.5 [50]) *Das Problem 2 mit $\mathcal{X} \subseteq \mathbb{R}^d$ sei konvex. Es gelte starke Dualität für das Problem, sodass die Dualitätslücke gleich null ist. Der primale Vektor x^* und der duale Vektor μ^* sind ein primal-dual optimales Paar dann und nur dann, wenn die folgenden Bedingungen erfüllt sind:*

$$x^* \in \underset{x \in \mathcal{X}}{\operatorname{argmin}} \mathcal{L}(x, \mu^*) : \text{Lagrangeoptimalität}, \quad (2.26)$$

$$x^* \in \mathcal{X}, g(x^*) \leq 0 : \text{Primale Lösbarkeit}, \quad (2.27)$$

$$\mu^* \geq 0 : \text{Duale Lösbarkeit}, \quad (2.28)$$

$$(\mu^*)^T g(x^*) = 0 : \text{Komplementärer Schlupf}. \quad (2.29)$$

Eine zu dieser Bedingung äquivalente Formulierung ist das sogenannte Sattelpunkttheorem, das den Fokus auf die Sattelpunkteigenschaft des primal-dualen Paares bezüglich der Lagrangefunktion legt.

Satz 2.3.2 (Sattelpunkttheorem, Proposition 6.1.6 [50]) *Das Problem 2 mit $\mathcal{X} \subseteq \mathbb{R}^d$ sei konvex. Es gelte starke Dualität für das Problem, sodass die Dualitätslücke gleich null ist. Das Paar $(x^* \in \mathcal{X}, \mu^* \geq 0)$ ist dann und nur dann ein primal-dual optimales Paar, wenn*

$$\mathcal{L}(x^*, \mu) \leq \mathcal{L}(x^*, \mu^*) \leq \mathcal{L}(x, \mu^*), \forall x \in \mathcal{X}, \mu \geq 0. \quad (2.30)$$

Die Äquivalenz zwischen der Optimalitätsbedingung 4 und Satz 2.3.2 lässt sich folgendermaßen argumentieren: Das Paar (x^*, μ^*) sei nach Optimalitätsbedingung 4 primal-dual optimal. Die rechte Seite der Ungleichung folgt aus der *Lagrangeoptimalität* aus Gleichung (2.26). Gilt die *Primale Lösbarkeit* aus Gleichung (2.27), sodass $g(x^*) \leq 0$, folgt für alle $\mu \geq 0$, dass $\mathcal{L}(x^*, \mu) \leq F(x^*)$, wobei $F(x^*) = \mathcal{L}(x^*, \mu^*)$ gilt, aufgrund der *Komplementären Schlupfbedingung* aus Gleichung (2.29). Entsprechend gilt die linke Seite der Ungleichung. Ebenso kann gezeigt werden, dass die Optimalitätsbedingung 4 erfüllt ist, wenn für das Paar (x^*, μ^*) die Sattelpunktbedingung (2.30) gilt, siehe Beweis von Proposition 6.1.6 in [50].

Demzufolge lässt sich eine Lösung des Problems 2 unter starker Dualität bestimmen, indem der Sattelpunkt der Lagrangefunktion bezüglich der primalen und dualen Variablen gefunden wird, d. h. die Lagrangefunktion bezüglich der Variablen x minimiert und bezüglich der Lagrangeparameter μ maximiert wird. Es existieren entsprechende Algorithmen, die diese Sattelpunkteigenschaft ausnutzen. Beispielsweise wird bei der dualen, projizierten (Sub-)Gradienten Methode aus [52, Abschnitt 8.5.2] zwischen einer Minimierung bezüglich der Variablen x und eines Gradientenaufstieges bezüglich μ iteriert, um den Sattelpunkt des Problems zu finden. In [53] wird ein zentrales Verfahren vorgestellt, bei dem sowohl für die Aktualisierung der primalen Variable, als auch für die Aktualisierung der dualen Variable der Gradient der Lagrangefunktion verwendet wird. Die entsprechenden Aktualisierungsvorschriften lauten

$$x(k+1) = \text{proj}_{x \in \mathcal{X}} [x(k) - \alpha \nabla_x \mathcal{L}(x(k), \mu(k))], \quad (2.31a)$$

$$\mu(k+1) = \text{proj}_{\mu \in \mathcal{D}} [\mu(k) + \alpha \nabla_\mu \mathcal{L}(x(k), \mu(k))], \quad (2.31b)$$

wobei \mathcal{D} die kompakte und konvexe Menge der dualen Lösungen ist und $\alpha > 0$. Gilt die Slater Bedingung aus Satz 2.3.1 bezüglich des Problems 2,

dann ist die dual optimale Menge \mathcal{M}^* nichtleer und beschränkt. Zusätzlich kann in diesem Fall mit dem Slater Vector \tilde{x} , dem Wert der dualen Funktion $\bar{G} = G(\bar{\mu})$ an einem Punkt $\bar{\mu} \geq 0$ und einem Skalar $r > 0$ die Wahl

$$\mathcal{D} = \left\{ \mu \geq 0 \mid \|\mu\| \leq \frac{F(\tilde{x}) - \bar{G}}{\gamma} + r \right\} \quad (2.32)$$

getroffen werden [53], wobei $\gamma = \min_{i \in \{1, \dots, M\}} \{-g_i(x)\}$.

Strafterme

Ähnlich dem vorausgegangenen Abschnitt werden auch bei der Straftermmethode Nebenbedingungen in Ungleichungsform zur Zielfunktion addiert, allerdings nicht über Lagrangemultiplikatoren, sondern unter Verwendung eines Strafparameters $r \in \mathbb{R}^+$ und einer sogenannten Straffunktion $\Psi : \mathbb{R}^d \rightarrow \mathbb{R}$ (englisch: penalty function). Diese Methode erfordert, dass die Nebenbedingungen explizit als (Un-)Gleichungen vorliegen, weswegen in diesem Abschnitt das Problem 2 mit $\mathcal{X} \equiv \mathbb{R}^d$ betrachtet wird.

Die Funktion $\Psi(x)$ wird als Straffunktion eines Problems bezeichnet, wenn sie die folgende, aus [54] auf die Problemstellung angepasste, Definition erfüllt:

Definition 2.3.2 *Eine kontinuierliche Funktion $\Psi : \mathbb{R}^d \rightarrow \mathbb{R}$ ist eine Straffunktion einer skalaren Nebenbedingungsungleichung $g(x) \leq 0$, wenn*

- $\Psi(x) = 0$ für $g(x) \leq 0$ und
- $\Psi(x) > 0$ für $g(x) > 0$

gilt.

Es existieren Funktionen mit unterschiedlichen Eigenschaften, die obige Definition erfüllen. In Bezug auf die Problemstellung 2 mit $\mathcal{X} \equiv \mathbb{R}^d$ gibt es beispielsweise die Optionen

- Nicht-glatte Strafterme [54]: $\Psi(x) = \sum_{i=1}^M \max\{0, g_i(x)\}$,
- Quadratische Strafterme [54]: $\Psi(x) = \sum_{i=1}^M \max\{0, g_i(x)^2\}$ und

- Strafterme mit beschränkten Gradienten [55]:

$$\Psi(x) = \sum_{i=1}^M p(g_i(x)) \text{ mit } p(u) = \begin{cases} \ln\left(\frac{e^u + e^{-u}}{2}\right), & \text{wenn } u > 0, \\ 0, & \text{wenn } u \leq 0. \end{cases} \quad (2.33)$$

In Abbildung 2.3 ist die letzte Option aus obiger Aufzählung zusammen mit dem entsprechenden Gradienten bei einer beispielhaften unteren und oberen Schranke einer Variablen $x \in \mathbb{R}$ unter verschiedenen Strafparametern r dargestellt.

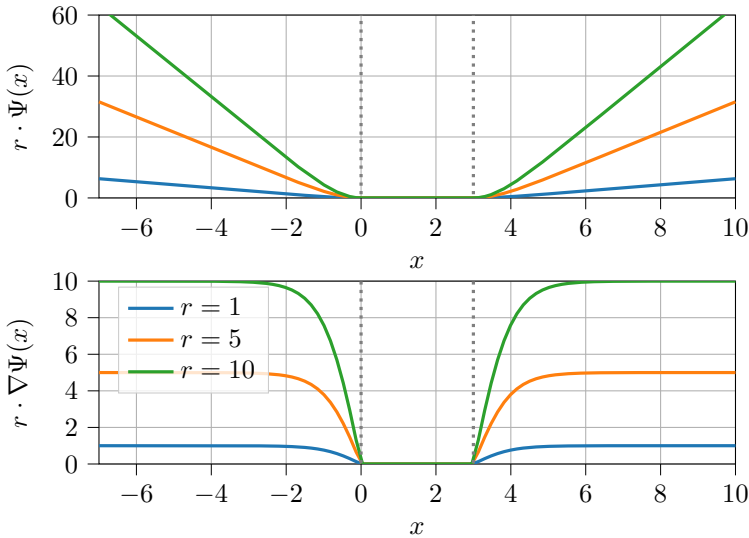


Abbildung 2.3: Veranschaulichung der Strafterme (2.33) für untere und obere Schranke $g_1(x) = -x$, $g_2(x) = x - 3$ und verschiedene Strafparameter r .

Unter Verwendung der Straffunktion und eines iterationsabhängigen Strafparameters r_k kann nun das ungleichungsbeschränkte Optimierungsproblem in ein unbeschränktes überführt werden:

$$\text{Problem 2} \rightarrow \min_x F(x) + r_k \Psi(x). \quad (2.34)$$

Eine Lösungsmethode für dieses Problem ist beispielsweise in [54, Abschnitt 1.3.3] zu finden. Für das dort vorgestellte Verfahren muss zunächst

eine Sequenz für den Strafparameter festgelegt werden, die $0 < r_k < r_{k+1}, r_k \rightarrow \infty$ erfüllt. Anschließend wird wiederholt das unbeschränkte Suboptimierungsproblem

$$x(k+1) \in \operatorname{argmin}_x [F(x(k)) + r_k \Psi(x(k))] \quad (2.35)$$

gelöst, wobei das Resultat $x(k)$ der vorausgegangenen Iteration jeweils als Startwert verwendet wird. Nach Theorem 1.3.1 aus [54] gilt unter obiger Vorschrift

$$\lim_{k \rightarrow \infty} F(x(k)) = F^*, \quad \lim_{k \rightarrow \infty} \Psi(x(k)) = 0, \quad (2.36)$$

wobei F^* das Optimum des ungleichungsbeschränkten Problems 2 repräsentiert. So einfach und universell dieses Verfahren ist, gibt es doch wenig theoretische Anhaltspunkte für die Wahl der Straffunktion $\Psi(x(k))$. Ebenso entscheidend wie obskur ist die optimale Wahl der Strafparametersequenz r_k . Letztlich verbleibt auch die Frage der Präzision der Lösung des Suboptimierungsproblems in Gleichung (2.35) für endliche Iterationsanzahlen in der Praxis zu beantworten.

2.4 Nicht-kooperative Spieltheorie

Mit Hilfe der nicht-kooperativen Spieltheorie können unterschiedliche Konkurrenzsituationen zwischen nicht miteinander kooperierenden Entitäten, den Spielern, modelliert werden. Im Folgenden werden zunächst die Bestandteile eines nicht-kooperativen, strategischen Spiels erklärt, bevor genauer auf die für diese Arbeit relevante Klasse von Spielen eingegangen und ein Lösungskonzept in Form des Nash-Gleichgewichts präsentiert wird. Anschließend wird die bedeutende Äquivalenz von spieltheoretischen Problemstellungen und Variationsungleichungen erklärt und anhand derer herausgearbeitet, welche Eigenschaften Spiele besitzen müssen, damit Nash-Gleichgewichte existieren.

Nach [47] besteht ein strategisches Spiel aus folgenden Elementen:

- einer Menge $\mathcal{I} := \{1, \dots, N\}$ von endlich vielen Spielern,
- Strategiemengen $\mathcal{X}^i \subseteq \mathbb{R}^{d^i}$ eines jeden Spielers $i \in \mathcal{I}$ und
- einer Kostenfunktion $F^i : \mathcal{X} \rightarrow \mathbb{R}^d$ für alle Spieler $i \in \mathcal{I}$, mit dem kartesischen Produkt $\mathcal{X} := \mathcal{X}^1 \times \dots \times \mathcal{X}^N$ aller Strategiemengen.

Durch diese Elemente kann ein strategisches Spiel vollständig beschrieben und verkürzt durch die Notation $\Gamma(N, \mathcal{X}, \{F^i\}_{i \in \mathcal{I}})$ repräsentiert werden.

Je nach zu modellierender Problemstellung hat ein solches Spiel unterschiedliche Eigenschaften. Aufbauend auf der Klassifikation aus [47, Kapitel 1] kann die für diese Arbeit relevante Klasse von Spielen definiert werden: Alle in der vorliegenden Arbeit betrachteten Spiele sind strategische Spiele und

- *nicht-kooperativ*: Es existiert keine Kooperation zwischen den Spielern, d. h. es werden keine verbindlichen Absprachen zwischen den Spielern getroffen. Darüber hinaus optimiert jeder Spieler egoistisch seine eigene Kostenfunktion, indem die eigene Strategie $x^i \in \mathcal{X}^i$ in Reaktion auf die Strategien der anderen Spieler x^{-i} angepasst wird. Nur Spieler i kann die Strategie x^i verändern.
- *kontinuierlich*: Alle Strategiemengen \mathcal{X}^i sind kontinuierlich.
- *simultan*: Alle Spieler wählen ihre Strategien gleichzeitig aus. Keinem der Spieler ist im Vorfeld die (unveränderliche) Strategie eines anderen Spielers bekannt.
- *einmalig*: Das Spiel wird nur ein einziges Mal gespielt. Es kann somit kein Lerneffekt durch eine wiederholte Ausführung des Spiels auftreten.
- besitzen eine *unvollständige Informationsstruktur*: Die Spieler besitzen nur unvollständige Informationen über die anderen Spieler. Insbesondere sind durch den verteilten Aufbau der Spiele, der in vorliegender Arbeit im besonderen Fokus steht, die Kostenfunktionen und Strategiemengen der Spieler privat. Ebenso haben die Spieler keinen direkten Zugriff auf die Strategien der jeweils anderen Spieler.
- *konvex*: Für fixierte Strategien x^{-i} der anderen Spieler ist die eigene Kostenfunktion des Spielers i konvex für alle $i \in \mathcal{I}$. Zusätzlich werden die Strategiemengen \mathcal{X}^i , $i \in \mathcal{I}$, als konvex angenommen.

Für obig definierte Spiele existieren unterschiedliche Lösungskonzepte [47, Kapitel 1]. Eine besondere Rolle spielt das Nash-Gleichgewicht, da dieses ein effizientes Lösungskonzept für die in der vorliegenden Arbeit behandelten Problemstellungen im Rahmen von Einsatzplanproblemen in Microgrids darstellt. Darüber hinaus existiert für Nash-Gleichgewichtsprobleme eine wichtige Äquivalenz zu Variationsungleichungen, auf die im späteren

Verlauf dieses Abschnittes detaillierter eingegangen wird. Zunächst wird jedoch der Begriff des Nash-Gleichgewichts definiert:

Definition 2.4.1 (Nash-Gleichgewicht) [47, Definition 1.5] *Der Strategievektor $x^* = \text{col}(x^{i,*}, x^{-i,*})$ ist ein Nash-Gleichgewicht des Spiels $\Gamma(N, \mathcal{X}, \{F^i\}_{i \in \mathcal{I}})$, wenn $x^* \in \mathcal{X}$ und*

$$F^i(x^{i,*}, x^{-i,*}) \leq F^i(x^i, x^{-i,*}), \forall x^i \in \mathcal{X}^i, \forall i \in \mathcal{I}. \quad (2.37)$$

Hierbei wird die übliche Notation verwendet, dass x^i die Strategie des Spielers i bezeichnet und x^{-i} die Strategien aller Spieler mit Ausnahme des Spielers i . Dieser Formulierung entsprechend hat keiner der Spieler einen Anreiz sich aus dem Nash-Gleichgewicht zu entfernen, wenn alle anderen Spieler an der Strategie $x^{-i,*}$ festhalten, da in diesem Fall sich nach Definition 2.4.1 die Kosten des Spielers i nicht verringern lassen. Damit stellt das Nash-Gleichgewicht einen stabilen Zustand des Spiels dar und $x^{i,*}$ kann als beste Antwort auf $x^{-i,*}$ verstanden werden.

Ein Spiel, das den obigen Definitionen entspricht und bei dem das Nash-Gleichgewicht als Lösungsprinzip verwendet wird, wird als Nash-Gleichgewichtsproblem bezeichnet. Mathematisch lässt sich ein Nash-Gleichgewichtsproblem mit N Spielern als die folgenden N verkoppelten Optimierungsprobleme formulieren:

$$\min_{x^i} F^i(x^i, x^{-i}), \quad (2.38a)$$

$$\text{s.t. } x^i \in \mathcal{X}^i, \quad (2.38b)$$

für $i \in \mathcal{I}$.

Nun stellt sich die Frage, unter welchen Voraussetzungen Nash-Gleichgewichte für ein Spiel $\Gamma(N, \mathcal{X}, \{F^i\}_{i \in \mathcal{I}})$ existieren und, falls ja, ob das Nash-Gleichgewicht eindeutig ist. Für die Lösung dieser Problemstellung können Variationsungleichungen (engl.: variational inequality) verwendet werden. Im Folgenden werden diese kurz vorgestellt und deren Verbindung zu Nash-Gleichgewichtsproblemen gezeigt. Für eine weiterführende, anschauliche Interpretation sei auf [56] verwiesen.

Nach Definition 3.1 aus [47] ist ein Variationsungleichungsproblem folgendermaßen definiert: Es sei eine konvexe, nichtleere und abgeschlossene Menge $\mathcal{X} \subseteq \mathbb{R}^d$ und eine Abbildung $M : \mathcal{X} \rightarrow \mathbb{R}^d$ gegeben. Eine finite Variationsungleichung VI $(\mathcal{X}, M(x))$ beschreibt nun die Aufgabe, einen

Vektor $x^* \in \mathcal{X}$ zu finden, sodass

$$M(x^*)^T(x - x^*) \geq 0, \forall x \in \mathcal{X} \quad (2.39)$$

gilt, d. h. $x^* \in \text{sol}(\mathcal{X}, M(x))$. Diese Bedingung kann als Verallgemeinerung des Minimumprinzips aus der Optimalitätsbedingung 1 für mehrdimensionale Abbildungen $M(x)$ interpretiert werden [56].

Die Existenz von Lösungen einer Variationsungleichung $\text{VI}(\mathcal{X}, M(x))$ kann nach [56] mit der Überprüfung der Monotonieeigenschaft der Abbildung $M(x)$ nachgewiesen werden: Sei \mathcal{X} abgeschlossen und konvex und $M(x)$ kontinuierlich auf \mathcal{X} . Dann gelten die Implikationen

- $M(x)$ ist monoton auf $\mathcal{X} \implies$ Die Lösungsmenge der $\text{VI}(\mathcal{X}, M(x))$ ist abgeschlossen und konvex.
- $M(x)$ ist strikt monoton auf $\mathcal{X} \implies$ Die $\text{VI}(\mathcal{X}, M(x))$ hat höchstens eine Lösung.
- $M(x)$ ist gleichmäßig monoton auf $\mathcal{X} \implies$ Die $\text{VI}(\mathcal{X}, M(x))$ hat eine eindeutige Lösung.

Nun da definiert ist, unter welchen Voraussetzungen Lösung für eine Variationsungleichung existieren, kann die Verbindung zu Nash-Gleichgewichtsproblemen, d. h. den verkoppelten Optimierungsproblemen aus Gleichung (2.38), hergestellt werden. Hierzu wird folgender Satz aus [57] zitiert:

Satz 2.4.1 [57, Proposition 1.4.2] *Seien $\mathcal{X}^i \subset \mathbb{R}^{d_i}$ abgeschlossene, konvexe Mengen und das Spiel $\Gamma(N, \mathcal{X}, \{F^i\}_{i \in \mathcal{I}})$ mit $\mathcal{X} = \mathcal{X}^1 \times \dots \times \mathcal{X}^N$ gegeben. Für jedes fixierte x^{-i} sind die Funktionen $F^i(x^i, x^{-i})$ konvex und kontinuierlich differenzierbar für alle $i \in \mathcal{I}$.*

Der Vektor x^ ist dann und nur dann ein Nash-Gleichgewicht des Spiels, wenn x^* eine Lösung der Variationsungleichung $\text{VI}(\mathcal{X}, M(x))$ ist, wobei*

$$M(x) = \text{col}((\nabla_i F^i(x^i, x^{-i}))_{i \in \mathcal{I}}). \quad (2.40)$$

Die multidimensionale Abbildung $M(x)$ wird nach der englischen Begriffsprägung im Folgenden als *Mapping* des Spiels bezeichnet. Folglich ist es möglich, die Existenz von Nash-Gleichgewichten eines konvexen und differenzierbaren Spiels $\Gamma(N, \mathcal{X}, \{F^i\}_{i \in \mathcal{I}})$ mit $\mathcal{X} = \mathcal{X}^1 \times \dots \times \mathcal{X}^N$ nachzuweisen, indem die Monotonie des Mappings $M(x)$ der entsprechenden Variationsungleichung untersucht wird.

Unter Berufung auf die entsprechende Variationsungleichung ist es außerdem möglich, eine Optimalitätsbedingungen für Nash-Gleichgewichtsprobleme der Form (2.38) anzugeben.

Optimalitätsbedingung 5 (Notwendig und hinreichend) Sei $\mathcal{X} = \mathcal{X}^1 \times \dots \times \mathcal{X}^N$ nichtleer, abgeschlossen und konvex. $M : \mathcal{X} \rightarrow \mathbb{R}^d$ sei das Mapping des konvexen und kontinuierlich differenzierbaren⁴ Spiels $\Gamma(N, \mathcal{X}, \{F^i\}_{i \in \mathcal{I}})$, definiert in Gleichung (2.40). Dann ist $x^* \in \mathbb{R}^d$ ein Nash-Gleichgewicht des Spiels genau dann, wenn x^* ein Fixpunkt der Abbildung

$$H(x) := \text{proj}_{\mathcal{X}} [x - \gamma M(x)] \quad (2.41)$$

ist, d. h. $x^* = H(x^*)$, wobei $\gamma > 0$ ein beliebiger, positiver Faktor ist.

Diese Optimalitätsbedingung folgt aus dem Zusammenhang der Lösungen von Nash-Gleichgewichtsproblemen und Variationsungleichungen aus Satz 2.4.1 und dem Satz 3.6 aus [47] über die Optimalitätsbedingungen von Variationsungleichungen.

Verallgemeinerte Nash-Gleichgewichtsprobleme Verallgemeinerte Nash-Gleichgewichtsprobleme unterscheiden sich von Nash-Gleichgewichtsproblemen bezüglich der Strategiemengen \mathcal{X}^i , $i \in \mathcal{I}$. Während diese bei Letzteren nach Problemstellung (2.38) konstant, d. h. unabhängig von den Strategien der Spieler, sind, besteht bei Ersteren eine Abhängigkeit dieser Mengen von den Strategien x^{-i} der restlichen Spieler im System. Entsprechen lässt sich das Verallgemeinerte Nash-Gleichgewichtsproblem nach [47, Kapitel 5] durch die verkoppelten Optimierungsprobleme

$$\min_{x^i} F^i(x^i, x^{-i}), \quad (2.42a)$$

$$\text{s.t. } x^i \in \mathcal{X}^i(x^{-i}), \quad (2.42b)$$

für alle $i \in \mathcal{I}$ beschreiben. Die Verkopplung der Probleme tritt hierbei sowohl in den Kostenfunktionen als auch in den Nebenbedingungen auf. Das Verallgemeinerte Nash-Gleichgewicht ist analog zum Nash-Gleichgewicht definiert: Der Vektor x^* ist ein Verallgemeinertes Nash-Gleichgewicht des Spiels $\Gamma(N, \mathcal{X}(x), \{F^i\}_{i \in \mathcal{I}})$, wenn

$$F^i(x^{i,*}, x^{-i,*}) \leq F^i(x^i, x^{-i,*}), \forall x^i \in \mathcal{X}^i(x^{-i,*}), \forall i \in \mathcal{I}. \quad (2.43)$$

⁴Das Spiel $\Gamma(N, \mathcal{X}, \{F^i\}_{i \in \mathcal{I}})$ wird als kontinuierlich differenzierbar bezeichnet, wenn die Kostenfunktionen aller Spieler $F^i, \forall i \in \mathcal{I}$, kontinuierlich differenzierbar sind.

Die Untersuchung von Verallgemeinerten Nash-Gleichwichten ist durch die zusätzliche Verkopplung der Strategiemengen der Spieler deutlich schwieriger im Vergleich zu der Untersuchung von Nash-Gleichgewichten. Eine Ausnahme stellen die sogenannten normalisierten bzw. variationalen Verallgemeinerten Nash-Gleichgewichte dar, die sich durch Variationsungleichungen untersuchen lassen, siehe hierzu Satz 5.6 aus [47, Kapitel 5].

Im Hauptteil werden ausschließlich Nash-Gleichgewichtsprobleme untersucht, jedoch werden Verallgemeinerte Nash-Gleichgewichte in der Literaturübersicht bezüglich Multiclusterspielen in Kapitel 5 erwähnt.

2.5 Kommunikation in Multiagentensystemen

Müssen sich Agenten in einem Multiagentensystem untereinander koordinieren, um eine bestimmte Aufgabe zu erfüllen, beispielsweise um eine gemeinsame Entscheidung zu treffen oder ein verteiltes Optimierungsproblem zu lösen, ist eine Kommunikationsarchitektur notwendig. Diese verbindet die Agenten untereinander und ermöglicht einen Informationsaustausch. Modelliert wird die Architektur in der vorliegenden Arbeit generisch als eine Menge von Kommunikationskanälen zwischen den Agenten. Die folgenden Definitionen und Darstellungen sind Standardbeschreibungen, die beispielsweise in [58] nachgeschlagen werden können.

Repräsentation als Graph Effizient lassen sich diese Kommunikationskanäle zusammen mit den Agenten als Graph \mathcal{G} beschreiben, wobei die Agenten die Knoten des Graphen darstellen und die Kanten zwischen den Knoten, insofern sie existieren, die Kommunikationskanäle. Dieser Graph ist definiert durch seine Knotenmenge \mathcal{V} , seine Kantenmenge \mathcal{E} und die Gewichtung seiner Kanten entsprechend der Gewichtungsmatrix $W \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, die im Verlauf dieses Abschnittes noch weiter diskutiert wird. Entsprechend ist ein Graph durch $\mathcal{G} \{ \mathcal{V}, \mathcal{E}, W \}$ vollständig beschrieben. Ist der Graph ungewichtet, wird in der vorliegenden Arbeit auf die explizite Aufführung von W verzichtet. Ist \mathcal{I} die Menge der Agenten des Netzwerkes, so gilt $\mathcal{V} \equiv \mathcal{I}$ für die Knotenmenge. Somit entspricht die Anzahl der Agenten N der Kardinalität der Menge \mathcal{V} , d. h. $|\mathcal{V}| = N$. Existiert ein Kommunikationskanal zwischen den Agenten i und j , sodass Agent i an Agenten j Informationen schicken kann, dann ist das Paar (i, j) Teil der Kantenmenge \mathcal{E} .

Gilt für jede Kante $(i, j) \in \mathcal{E}$ auch $(j, i) \in \mathcal{E}$, d. h. kann zwischen jedem Agentenpaar $i, j \in \mathcal{V}$, die über eine Kante miteinander verbunden sind, ein Informationsaustausch in beide Richtungen stattfinden, wird der Graph als *ungerichtet* bezeichnet und die Kanten ohne Richtungsmarkierung dargestellt, siehe Abbildung 2.4 (a). Gilt dies für mindestens eine Verbindung $i, j \in \mathcal{E}$ nicht, so wird der Graph als *gerichtet* bezeichnet und die Kanten entsprechend mit einer Richtungsmarkierung versehen, siehe Abbildung 2.4 (b).

Es wird davon ausgegangen, dass jeder Agent sogenannte Eigenschleifen (engl.: selfloops) besitzt, d. h. Informationen an sich selbst schicken. Es gilt folglich, dass $(i, i) \in \mathcal{E}$, $\forall i \in \{1, \dots, |\mathcal{V}|\}$. Praktisch bedeutet dies, dass jeder Agent Zugriff auf seine eigenen Informationen hat. In den in der vorliegenden Arbeit verwendeten Darstellungen von Kommunikationsgraphen wird auf die explizite Darstellung der Eigenschleifen aus Gründen der Übersichtlichkeit verzichtet und nur die Kommunikationskanäle zwischen zwei unterschiedlichen Agenten werden dargestellt.

Nachbarschaft Ist ein Graph gerichtet, wird für jeden Agenten i zwischen Eingangsnachbarn und Ausgangsnachbarn, d. h. direkten Nachbarn von Agent i , die diesem Agenten Informationen schicken, und Nachbarn, die von Agent i Informationen empfangen können, unterschieden. Alle Eingangsnachbarn von i befinden sich in der Menge \mathcal{N}_i^+ , während sich alle Ausgangsnachbarn in \mathcal{N}_i^- befinden. Formal gilt, dass $j \in \mathcal{N}_i^+$, wenn $(j, i) \in \mathcal{E}$ und $j \in \mathcal{N}_i^-$, wenn $(i, j) \in \mathcal{E}$. In der vorliegenden Arbeit wird davon ausgegangen, dass jeder Agent i sein eigener Ausgangs- und Eingangsnachbar ist, d. h. $i \in \mathcal{N}_i^+$ und $i \in \mathcal{N}_i^-$. Der Anzahl der Eingangs- bzw. Ausgangsnachbarn entsprechend, wird der Eingangsgrad δ_i^+ bzw. Ausgangsgrad δ_i^- des Agenten i definiert. Es gilt $\delta_i^+ = |\mathcal{N}_i^+|$ und $\delta_i^- = |\mathcal{N}_i^-|$.

Verbundenheit Die Eigenschaft der Verbundenheit eines Graphen gibt an, ob Informationen von einem beliebigen Knoten ausgehend mit dem gesamten Netzwerk geteilt werden können. Entsprechend lässt sich die Verbundenheit für ungerichtete Graphen formal wie folgt formulieren:

Definition 2.5.1 (Verbundenheit ungerichteter Graphen) *Ein ungerichteter Graph $\mathcal{G} \{\mathcal{V}, \mathcal{E}\}$ wird als verbunden bezeichnet, wenn ein ungerichteter Pfad von jedem Knoten i zu jedem anderen Knoten j für alle Paare $i, j \in \mathcal{V}$ existiert.*

Der Graph in Abbildung 2.4 (a) ist beispielsweise verbunden. Analog

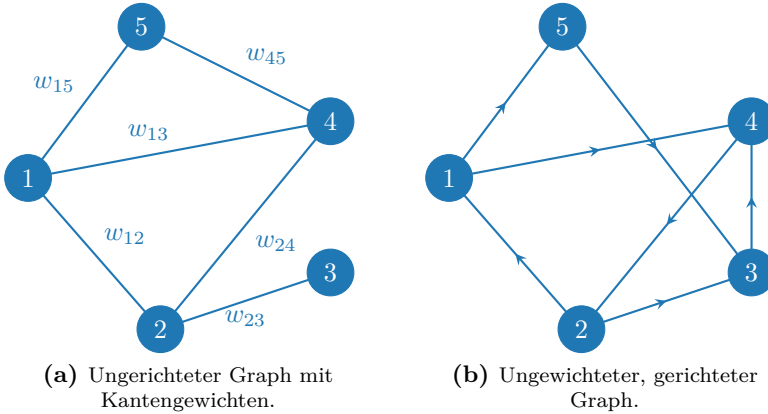


Abbildung 2.4: Agentensystem mit gerichteter und ungerichteter Kommunikationsstruktur.

lässt sich eine wichtige Verbundenheitseigenschaft für gerichtete Graphen definieren:

Definition 2.5.2 (Starke Verbundenheit gerichteter Graphen) *Ein gerichteter Graph $\mathcal{G} \{\mathcal{V}, \mathcal{E}\}$ wird als stark verbunden bezeichnet, wenn ein gerichteter Pfad von jedem Knoten i zu jedem anderen Knoten j für alle $i, j \in \mathcal{V}$ existiert.*

Ein Beispiel für einen stark verbundenen, gerichteten Graphen ist in Abbildung 2.4 (b) zu finden.

Taktung der Kommunikation In der vorliegenden Arbeit wird der Informationssaustausch als synchron und damit getaktet betrachtet. Das bedeutet, dass alle Agenten über eine Uhr verfügen, die synchron mit den Uhren der anderen Agenten tickt. Bei jedem Tick der Uhr versendet der Agent über seine ausgehenden Kanten Informationen an seine Nachbarn und empfängt gleichzeitig Informationen über seine Eingangskanäle. Es gibt verschiedene Arbeiten, bei denen die Uhr der Agent asynchron, beispielsweise nach einem zufälligen Poissonprozess tickt, siehe beispielsweise [28][59]. Entsprechend senden und empfangen die Agenten Informationen asynchron. Dieser Fall wird allerdings im Folgenden nicht weiter betrachtet.

Adjazenz- und Gewichtungsmatrix Die Adjazenzmatrix A eines Graphen ist eine effiziente Möglichkeit, die Knoten und Kanten zu repräsentieren. Diese Matrix der Dimension $|\mathcal{V}| \times |\mathcal{V}|$ enthält in ihren Einträgen A_{ij} den Wert 1, wenn $(j, i) \in \mathcal{E}$, d. h. wenn der Agent j an den Agenten i Informationen schicken kann. In Folge der Existenz von Eigenschleifen gilt für alle in der vorliegenden Arbeit behandelten Graphen $A_{ii} = 1, \forall i \in \mathcal{I}$.

Von der Adjazenzmatrix lassen sich weitere Matrizen ableiten, die sich durch die Gewichtung ihrer nicht-nullwertigen Einträge unterscheiden. Es gilt für die Einträge einer solchen gewichteten Matrix $W \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, dass

$$W_{ij} = \begin{cases} w_{ij}, & \text{wenn } A_{ij} > 0, \\ 0, & \text{wenn } A_{ij} = 0, \end{cases} \quad (2.44)$$

wobei $w_{ij} > 0$ die entsprechende Gewichtung der Kante (j, i) repräsentiert. Folglich beschreibt die Matrix W einen Graphen, bei dem die Kanten mit Gewichten versehen sind, siehe hierzu die Abbildung 2.4 (a). Dieser gewichtete Graph kann als das Tupel $\mathcal{G} \{\mathcal{V}, \mathcal{E}, W\}$ beschrieben werden. Für die nicht-null Gewichte der Matrix gilt $w_{ij} > \eta$ mit $0 < \eta$. Damit existiert eine positive untere Schranke η für alle Gewichte, die ungleich null ist. Somit wird verhindert, dass das Gewicht einer Kante beliebig nah an null gewählt wird.

Zeitvariante Graphen Als zeitvariant werden in der vorliegenden Arbeit Graphen bezeichnet, deren Kantenmenge zeitvariant ist, d. h., dass $\mathcal{E}(k)$ in jeder Iteration k unterschiedlich sein kann. Damit können Störungen der Kommunikation modelliert werden, wenn beispielsweise der Kommunikationskanal zwischen zwei Agenten gestört wird, oder wenn ein Agent aus dem Kommunikationsradius eines sendenden Agenten austritt und in den eines anderen eintritt. Explizit werden hierbei Änderungen der Knotenmenge \mathcal{V} ausgeschlossen, d. h. kein Agent kann das betrachtete System verlassen und kein zusätzlicher Agent kann das System betreten. Interessant ist hierbei, dass nicht unbedingt jeder einzelne gerichtete bzw. ungerichtete $\mathcal{G}(k) = \mathcal{G} \{\mathcal{V}, \mathcal{E}(k)\}$ stark verbunden bzw. verbunden sein muss. Beispielsweise reicht für die Konsensusbildung bezüglich eines bestimmten Wertes über einen zeitvarianten Kommunikationsgraphen die schwächere, B -(starke) Verbundenheit aus:

Definition 2.5.3 (B -(starke) Verbundenheit) *Eine Sequenz aus (gerichteten) Graphen $\{\mathcal{G}(k)\}$ ist B -(stark) verbunden, wenn für jeden Zeitschritt*

$k \geq 0$ der Graph

$$\mathcal{G}(k : k + B) = \{\mathcal{V}, \mathcal{E}(k) \cup \mathcal{E}(k + 1) \cup \dots \cup \mathcal{E}(k + B - 1)\} \quad (2.45)$$

(stark) verbunden ist. Das Intervall B ist endlich.

Diese Definition folgt entsprechenden Formulierungen für gerichtete Graphen aus beispielsweise [18] und [33]. Ist somit ein Graph B -(stark) verbunden, existiert ein Zeitintervall der endlichen Länge B , sodass die Vereinigung der Graphen $\mathcal{G}(k : k + B)$ in diesem Intervall (stark) verbunden ist. Mit $k \rightarrow \infty$ existieren unendlich viele (stark) verbundene Vereinigung von Graphen.

Gewichtung der Kommunikation Es existieren drei besondere Arten die Gewichtung der Matrix und damit der Kanten des Graphes zu wählen, die für diese Arbeit von besonderem Interesse sind. Diese sind in der folgenden Definition zusammengefasst:

Definition 2.5.4 [Reihen-, spalten- und doppelt-stochastische Matrizen]
Eine Matrix $W \in \mathbb{R}^{N \times N}$

- ist reihen-stochastisch, wenn alle Reihensummen gleich eins sind, d. h. $W\mathbf{1} = \mathbf{1}$,
- ist spalten-stochastisch, wenn alle Spaltensummen gleich eins sind, d. h. $\mathbf{1}^T W = \mathbf{1}^T$,
- ist doppelt-stochastisch, wenn sowohl alle Reihensummen als auch alle Spaltensummen gleich eins sind, d. h. $W\mathbf{1} = \mathbf{1}$ und $\mathbf{1}^T W = \mathbf{1}^T$.

Die Gewichtung der Matrix ist entscheidend für die Kommunikation im Agentensystem. Wie in Abschnitt 2.6 deutlich werden wird, kann beispielsweise nur bei bestimmten Gewichtungen ein Konsensus bezüglich bestimmter Vektoren im System erreicht werden.

Nun stellt sich die Frage, wie entsprechende Gewichtungen in einem Agentennetzwerk erzeugt werden können, ohne dass die Agenten die gesamte Struktur des Graphen kennen und ohne dass eine zentrale Einheit die Gewichtung vornimmt. Es wird davon ausgegangen, dass die Agenten die Gewichtung der jeweiligen Kante selbst vornehmen können, indem sie alle Informationen, die sie über eine bestimmte Kante verschicken mit einem entsprechenden Faktor versehen oder die erhaltenen Nachrichten

entsprechend gewichten, bevor diese weiterverarbeitet werden. Will somit beispielsweise Agent i die Information über einen reellen Wert $a \in \mathbb{R}$ an den Agenten j über die Kante (i, j) verschicken, die mit der Gewichtung W_{ji} versehen sein soll, dann entspricht dies der Übertragung der Information $W_{ji}a$ von Agent i zu Agent j über eine ungewichtete Kante.

Es sei der restriktivste Informationsfall angenommen, bei dem kein Agent Informationen über seine Nachbarschaft bzw. über die Kanten hat, die ihn mit anderen Agenten verbinden. Eigene Informationen werden als Broadcast versendet, wobei bei dem Abschicken der Sendung nicht klar ist, welche und wie viele Agenten die Informationen empfangen. Bei Erhalt einer Nachricht weiß der empfangende Agent nicht, von welchem sendenden Agenten die Informationen stammen, ausschließlich die Anzahl der synchron eingehenden Nachrichten kann registriert werden.

Reihen-stochastische Gewichtung Wenn davon ausgegangen wird, dass pro synchronem Zeitschritt jeder Agent genau eine Nachricht broadcastet, kann jeder Agent i seinen eigenen Eingangsgrad δ_i^+ , d. h. die Anzahl seiner Eingangsnachbarn, bestimmen, indem die Anzahl der empfangenen Nachrichten gezählt wird. Ist jedem Agenten $i \in \mathcal{I}$ der eigene Eingangsgrad bekannt, kann eine reihen-stochastische Gewichtung der Kommunikationsmatrix dadurch erreicht werden, dass jeder Agent i die empfangenen Nachrichten, einschließlich der Nachricht der Eigenschleife, mit dem Faktor $1/\delta_i^+$ gewichtet, d. h.

$$W_{ij \in \mathcal{I}} = \begin{cases} 1/\delta_i^+, & \text{wenn } A_{ij} > 0, \\ 0, & \text{sonst.} \end{cases} \quad (2.46)$$

Für die entsprechenden Reihensummen der Kommunikationsmatrix ergibt sich die reihen-stochastische Gewichtung

$$\sum_{j=1}^N W_{ij} = \sum_{j=1}^{|\mathcal{N}_i^+|} 1/\delta_i^+ = \delta_i^+/\delta_i^+ = 1 \quad (2.47)$$

für alle $i \in \mathcal{I}$.

Spalten-stochastische Gewichtung Die spalten-stochastische Gewichtung lässt sich ähnlich erzeugen. Hierfür muss jedem Agenten sein Ausgangsgrad δ_i^- bekannt sein. Wenn dann jeder Agent seine ausgehen-

den Nachrichten mit seinem Ausgangsgrad δ_i^- gewichtet, ergibt sich

$$W_{i \in \mathcal{I}j} = \begin{cases} 1/\delta_j^-, & \text{wenn } A_{ij} > 0, \\ 0, & \text{sonst.} \end{cases} \quad (2.48)$$

Für die Spaltensummen folgt somit

$$\sum_{i=1}^N 1/\delta_j^- = \sum_{i=1}^{|\mathcal{N}_j^-|} 1/\delta_j^- = \delta_j^-/\delta_j^- = 1 \text{ für alle } j \in \mathcal{I}. \quad (2.49)$$

Die spalten-stochastische Gewichtung ist allerdings aufwendiger herzustellen als die reihen-stochastische, da der Ausgangsgrad eines Agenten nicht direkt, wie zum Beispiel beim Eingangsgrad durch ein einfaches Verschieben von Nachrichten, festgestellt werden kann. Es gibt verteilte Algorithmen, wie zum Beispiel der in [60] vorgestellte, mit denen der Ausgangsgrad ohne vollständige Graphinformation berechnet werden kann. Ist der Graph allerdings zeitvariant, d. h. die Kantenmenge $\mathcal{E}(k)$ und damit auch die Gewichtungsmatrix $W(k)$ sind iterationsabhängig, müssen die Ausgangsgrade der Agenten vor jedem Kommunikationsschritt über die zu gewichtende Matrix $W(k)$ neu berechnet werden.

Doppelt-stochastische Gewichtung Die Aufwendigkeit der Generierung einer doppelt-stochastischen Gewichtung eines Graphen ist stark davon abhängig, ob der Graph gerichtet oder ungerichtet ist. Bei ungerichteten Graphen ist aufgrund der symmetrischen Kommunikation für jeden Agenten $i \in \mathcal{I}$ der Ausgangsgrad gleich dem Eingangsgrad, $\delta_i^- = \delta_i^+$, und die Gewichtungsmatrix W ist symmetrisch. Wird nun beispielsweise nach Gleichung (2.46) eine reihen-stochastische Gewichtung hergestellt, ist die Matrix aufgrund der Symmetrie auch spalten-stochastisch. Entsprechend ist es aufgrund der Kommunikationssymmetrie bei ungerichteten Graphen immer möglich, eine doppelt-stochastische Gewichtung der Gewichtungsmatrix W zu erzeugen, siehe hierzu auch [61, Korollar 3.4].

Eine doppelt-stochastische Gewichtung bei gerichteten Graphen zu erstellen, gestaltet sich schwieriger, da im Allgemeinen die Kommunikation asymmetrisch ist. Ein Sonderfall stellen hier die *balancierten* gerichteten Graphen dar. Ein ungewichteter, gerichteter Graph $\mathcal{G} \{\mathcal{V}, \mathcal{E}, A\}$ wird in der vorliegenden Arbeit als *balanciert* bezeichnet, wenn für die Adjazenzmatrix $\sum_{j=1}^N A_{ij} = \sum_{j=1}^N A_{ji}$, $\forall i \in \{1, \dots, N\}$ gilt⁵. Damit ist ein gerichteter

⁵In der vorliegenden Arbeit wird zwischen der *ungewichteten* Balance eines Graphen,

Graph balanciert, wenn für alle Knoten $i \in \{1, \dots, N\}$ der Ausgangsgrad gleich dem Eingangsgrad ist, d. h. $\delta_i^+ = \delta_i^- = \delta_i$. Jeder ungerichtete Graph ist aufgrund seiner Symmetrie balanciert, die Umkehrung dieser Aussage gilt allerdings nicht, da gerichtete, balancierte Graphen nicht zwangsweise über bidirektionale Kommunikationsverbindungen verfügen müssen. Für balancierte, gerichtete Graphen ist die Erzeugung einer doppelt-stochastischen Gewichtung wie auch im Fall der ungerichteten Graphen einfach: Es reicht aus, jede Zeile i wie in (2.46) mit dem Grad δ_i des jeweiligen Knotens i zu gewichten um eine doppelt-stochastische Gewichtung zu erzeugen. Der Grad kann hierbei wie beschrieben entsprechend einfach verteilt ermittelt werden.

Für allgemeine stark verbundene und gerichtete Graphen lässt sich folgender Satz beweisen:

Satz 2.5.1 *Für jeden stark verbundenen, gerichteten Graphen mit Eigenschleifen lässt sich eine Gewichtung finden, sodass dessen Gewichtsmatrix doppelt-stochastisch wird.*

Beweis. Sei $A \in \mathbb{R}^{N \times N}$ die ungewichtete Adjazenzmatrix eines stark verbundenen, gerichteten Graphen. Nach [61] ist die starke Verbundenheit des Graphen eine hinreichende und notwendige Bedingung dafür, dass eine gewichtete Matrix W^{bal} gefunden werden kann, sodass

$$\sum_{i=1}^N W_{ij}^{\text{bal}} = \sum_{i=1}^N W_{ji}^{\text{bal}} \quad (2.50)$$

für alle $j \in \mathcal{I}$ gilt. Nach Theorem 3.2 aus [61] kann durch die Neugewichtung der Elemente der Matrix W^{bal} durch die Zuweisung $\phi : (W_{ij}^{\text{bal}}) \rightarrow \frac{W_{ij}^{\text{bal}}}{\sum_{k=1}^N W_{ik}^{\text{bal}}}$, also durch die Gewichtung der Matrixelement mit der Summe der jeweiligen Zeile, eine doppelt-stochastische Matrix $D = \phi(W^{\text{bal}})$ dann und nur dann erzeugt werden, wenn sich alle Zeilensummen zur gleichen Konstante summieren, also $\sum_{k=1}^N W_{ik}^{\text{bal}} = C, \forall i \in \{1, \dots, N\}$ gilt. Wie in

wie oben definiert, und einer *gewichteten* Balance unterschieden, wobei für letztere für die Gewichtsmatrizen $\sum_{i=1}^N W_{ij} = \sum_{i=1}^N W_{ji}$ gelten muss, siehe beispielsweise [61]. Die Definition der ungewichteten Balance ist dabei deutlich restriktiver, da hier die Anzahl der Eingangs- und Ausgangsnachbarn für jeden Knoten gleich sein muss. Es kann leicht überprüft werden, dass damit die ungewichtete Balance ein Sonderfall ist und nicht für jeden stark verbundenen gerichteten Graphen gilt, während in [61] angeführt wird, dass für jeden stark verbundenen Graph eine Gewichtung gefunden werden kann, sodass dieser gewichtet balanciert ist.

Korollar 3.3 aus [61] festgestellt, können die Gewichte der Eigenschleifen so festgelegt werden, dass sich die Konstante C ergibt. Grund hierfür ist, dass die Gewichte der Eigenschleifen als Diagonalelemente keinen Einfluss auf die Gleichung (2.50) haben. \square

Die Identifikation einer solchen Gewichtung erweist sich in der Praxis als deutlich komplexer als in den bisher betrachteten Fällen. Nichtsdestotrotz gibt es verteilte Verfahren, wie zum Beispiel vorgestellt in [61], mit denen es möglich ist, doppelt-stochastische Gewichtungen für beliebige, stark verbundene, gerichtete Graphen zu erzeugen. Wie auch in [31] vermerkt, ist ein solches Verfahren aufwendig und müsste bei zeitvarianten Graphen bei jeder Änderung der Graphstruktur von neuem ausgeführt werden. Bei balancierten gerichteten Graphen und ungerichteten Graphen hingegen kann die doppelt-stochastische Gewichtung deutlicher einfacher über die Gewichtung der empfangenen Nachrichten durch den eigenen Eingangsgrad festgelegt werden.

2.6 Konsensus in Multiagentensystemen

Damit autonom agierende Agenten eines Multiagentensystems ein gemeinsames Ziel erreichen bzw. eine gemeinsame Entscheidung treffen können, müssen sich die Agenten zunächst auf eine Aktion, die optimal bezüglich des zu erreichenden Zieles ist, einigen. Das Ergebnis dieser Einigung wird als Konsens oder Konsensus bezeichnet. Um effizient einen Konsens bezüglich einer auszuführenden Aktion erreichen zu können, folgt der Prozess der Einigung einem fest vorgegebenen Protokoll, das als *Konsensusprotokoll* bezeichnet wird.

Sei ein Agentensystem mit insgesamt N Agenten gegeben, die sich in der Menge $\mathcal{I} = \{1, \dots, N\}$ befinden. In einem Konsensusverfahren wird davon ausgegangen, dass jeder Agent $i \in \mathcal{I}$ eine eigene (Ein-)Schätzung $x_i \in \mathbb{R}^d$, initialisiert mit $x_i(0) = x_{i,0}$, bezüglich der vom Agentensystem auszuführenden Aktion $x \in \mathbb{R}^d$ hat. Für eine vereinfachte Darstellung werden im Folgenden skalare Aktionsvektoren mit der Dimension $d = 1$ angenommen, die Überlegungen lassen sich allerdings ohne Weiteres auf den mehrdimensionalen Fall erweitern. Ziel der Konsensusbildung ist es, sich asymptotisch auf einen Wert x_K zu einigen, der sich innerhalb der konvexen Hülle der Menge $\mathcal{X} = \{x_{1,0}, \dots, x_{N,0}\}$ befindet, d. h. $x_K = \sum_{i=1}^N \gamma_i x_{i,0}$, wobei für die Faktoren $\gamma_i > 0$, $\forall i \in \mathcal{I}$, und $\sum_{i=1}^N \gamma_i = 1$ gilt. Von besonderem Interesse ist hierbei oft die Gewichtung $\gamma_i = 1/N$, $\forall i \in \{1, \dots, N\}$, bei welcher der

Konsensuswert x_K den Mittelwert der Initialisierung annimmt und somit $x_K = 1/N \sum_{i=1}^N x_{i,0}$ gilt. Ein Protokoll, das diesen Konsensuswert erzeugen kann, wird *Mittelwertkonsensusprotokoll* genannt.

In kommunikationsbasierten Konsensusprozessen, die für diese Arbeit von Relevanz sind, tauschen Agenten ihre Schätzungen x_i untereinander über einen Kommunikationsgraphen aus. In jeder Iteration des Konsensusprotokolls wird dabei einmal und synchron die eigene Schätzung kommuniziert, sodass jedem Agenten für die Aktualisierung der eigenen Schätzung ausschließlich die Schätzungen der direkten Eingangsnachbarn zur Verfügung stehen. Sei $k \in \mathbb{N}^0$ der diskrete Zeitschritt der synchronen Taktung der Kommunikation. Während der Aktualisierung wird die neue eigene Schätzung $x_i(k+1)$ aus einer gewichteten Summe der eigenen Schätzung der vorherigen Iteration $x_i(k)$ und den Schätzungen der Agenten $j \in \mathcal{N}_i^+$ gebildet, wobei die Gewichtung der einzelnen Summanden aus der Gewichtungsmatrix W des Kommunikationsgraphen hervorgeht. Das einfachste Protokoll dieser Form kann folglich durch die Vorschrift

$$x_i(k+1) = \sum_{j=1}^N W_{ij} x_j(k) \quad (2.51)$$

für den Agenten i und entsprechend vektorisiert mit $x(k) = \text{col}(x_1(k), \dots, x_N(k)) \in \mathbb{R}^N$ als

$$x(k+1) = Wx(k) \quad (2.52)$$

beschrieben werden. Unter Annahme von ungerichteten Graphen und einer doppelt-stochastischen Gewichtung der Kanten wird in [62] bewiesen, dass unter dieser Vorschrift $\lim_{k \rightarrow \infty} x_i(k) = \frac{1}{N} \sum_{j=1}^N x_j(0)$, $\forall i \in \mathcal{I}$, gilt und somit der Mittelwertkonsensus asymptotisch erreicht wird.

Der Konvergenzbeweis kann über das Perron-Frobenius-Theorem für nicht-negative, primitive Matrizen, siehe [63], erfolgen und soll im Folgenden kurz skizziert werden. Seien $w \in \mathbb{R}^N$ und $v \in \mathbb{R}^N$ die positiven linken und rechten Eigenvektoren zum betragsmäßig größten Eigenwert λ_1 der Matrix W . Dann gilt nach dem genannten Theorem $\lim_{k \rightarrow \infty} W^k = \lambda_1 v w^T$ mit normierten Vektoren $v^T w = 1$. Es sei W nun doppelt-stochastisch gewichtet, wodurch $\lambda_1 = 1$ gilt. Für die linken und rechten Eigenvektoren folgt, dass deren Elemente jeweils gleich sein müssen und somit, unter Berücksichtigung der Normierung, $v_1 = v_2 = \dots = v_N = 1/\sqrt{N}$ und $w_1 = w_2 = \dots = w_N = 1/\sqrt{N}$ gilt. Für die Aktualisierungsvorschrift aus

(2.51) folgt demnach

$$\lim_{k \rightarrow \infty} x(k) = vw^T x(0) = \mathbf{1} \frac{1}{N} \sum_{i=1}^N x_i(0), \quad (2.53)$$

wodurch asymptotisch der Mittelwert der Anfangszustände der Agenten erreicht wird. Im Folgenden soll darauf eingegangen werden, wie sich der Grenzwert für $k \rightarrow \infty$ verändert, wenn die Kommunikationsmatrix nicht mehr doppelt-, sondern nur noch reihen- oder spalten-stochastisch ist.

Kommunikation bei spalten-stochastischer Gewichtung Bei Kommunikation über spalten-stochastisch gewichtete Graphen bleibt die „Masse“ im Netzwerk erhalten. Das bedeutet, dass zu jedem diskreten Zeitpunkt $k \in \mathbb{N}^0$ die Summe der Schätzungen x_i der Agenten im Netzwerk gleich ist, d. h. $\sum_{i=1}^N x_i(0) = \sum_{i=1}^N x_i(k)$, $\forall k$. Das Perron-Frobenius-Theorem lässt sich auch auf spalten-stochastische Matrizen unter Verwendung der Normierung $v^T w = 1$ der linken und rechten Eigenvektoren der Kommunikationsmatrix anwenden. Aufgrund der Gewichtung muss für den linken Eigenvektor gelten, dass alle Einträge den gleichen Wert besitzen. Ohne Einschränkung der Allgemeinheit kann daher die Normierung $w^T = \mathbf{1}^T$ angenommen werden. Bei spalten-stochastischer Gewichtung gilt für die Iterationsvorschrift aus (2.52) der Grenzwert

$$\lim_{k \rightarrow \infty} x(k) = vw^T x(0) = v \sum_{i=1}^N x_i(0). \quad (2.54)$$

Aufgrund der Normierung $v^T w = v^T \mathbf{1} = 1$ entspricht die Summe der Endwerte der Summe der Anfangswerte. Da der Vektor v nur dann gleiche Einträge besitzt, wenn die Kommunikationsmatrix doppelt-stochastisch ist, wird kein Konsensus der Agenten erreicht. Daher sind spalten-stochastische Matrizen zwar masseerhaltend, aber nicht konsensusbildend.

Kommunikation bei reihen-stochastischer Gewichtung Reihen-stochastisch gewichtete Graphen sind im Gegensatz zu spalten-stochastisch gewichteten Graphen nicht masseerhaltend. Da sich allerdings alle Reihen der Kommunikationsmatrix zu eins summieren, liegen die Schätzungen aller Zeitschritte $k \in \mathbb{N}^0$ innerhalb der konvexen Hülle, die von den Anfangsschätzungen $x(0) = \text{col}(x_1(0), \dots, x_N(0))$ aufgespannt wird. Zusätzlich hat der rechte Eigenvektor v aufgrund der reihen-stochastischen Gewichtung identische Einträge in allen Dimensionen und

kann zu $v = \mathbf{1}$ normiert werden. Für den Grenzwert der Aktualisierungsvorschrift (2.52) folgt

$$\lim_{k \rightarrow \infty} x(k) = v w^T x(0) = \mathbf{1} \sum_{i=1}^N w_i x_i(0). \quad (2.55)$$

Entsprechend wird ein Konsensus unter den Agenten des Systems erreicht. Dieser Konsensuswert entspricht allerdings nicht wie im doppeltstochastischen Fall dem Mittelwert der Anfangszustände, sondern einer gewichteten Summe, die von dem linken Eigenvektor w abhängt.

Damit ist es unter spalten- bzw. reihen-stochastisch gewichteten Kommunikationsmatrizen unter Verwendung der Vorschrift (2.52) nicht möglich einen Grenzwert zu erreichen, der unabhängig von rechtem bzw. linkem Eigenvektor ist.

2.6.1 Mittelwertkonsensus bei spalten-stochastischer Gewichtung: Das Push-Sum Protokoll

Wie bereits festgestellt, ist es zum einen nicht praktikabel, bei unbalancierten, gerichteten Graphen eine doppeltstochastische Gewichtung verteilt zu erzeugen. Zum anderen ist es nicht möglich, bei nicht vorhandener doppeltstochastischer Gewichtung mit dem Konsensusprotokoll aus Gleichung (2.52) einen von der Graphstruktur unabhängigen Konsensus zu erzeugen.

In [64][65] ist daher das sogenannte Push-Sum Protokoll veröffentlicht, mit dem unter Verwendung von ausschließlich spalten-stochastisch gewichteter Kommunikation der Mittelwert bezüglich der verteilten Schätzungen erreicht werden kann. In diesem Protokoll wird ein zusätzlicher Kommunikationsschritt in jeder Iteration eingeführt, mit welchem der rechte Eigenvektor v der spalten-stochastisch gewichteten Kommunikationsmatrix W , gespeichert in einer Variablen $y \in \mathbb{R}^N$, ermittelt werden kann. Unter Verwendung der vektorisierten Schätzungen $x(k) = \text{col}(x_1(k), \dots, x_N(k)) \in \mathbb{R}^N$ erfolgen die Aktualisierungsschritte des Push-Sum Protokolls nach

$$y(k+1) = W y(k), \quad (2.56a)$$

$$x(k+1) = W x(k). \quad (2.56b)$$

Wird die Initialisierung $y(0) = \mathbf{1}$ verwendet, gilt nach dem Perron-

Frobenius-Theorem für den Grenzwert

$$\lim_{k \rightarrow \infty} y(k) = v w^T y(0) = v \sum_{i=1}^N y_i(0) = N v. \quad (2.57)$$

Wird nun in einer neuen Variablen $z(k) \in \mathbb{R}^N$ der Grenzwert $x(k)$ aus Gleichung (2.54) mit dem Grenzwert für $y(k)$ ins Verhältnis gesetzt, ergibt sich für jeden Agenten $i \in \mathcal{I}$

$$\lim_{k \rightarrow \infty} z_i(k) = \lim_{k \rightarrow \infty} \frac{x_i(k)}{y_i(k)} = \frac{v_i \sum_{i=1}^N x_i(0)}{N v_i} = \frac{1}{N} \sum_{i=1}^N x_i(0). \quad (2.58)$$

Damit kann durch die Gewichtung mit $y(k)$ die Asymmetrie des Grenzwertes von $x(k)$ in der Variablen $z(k)$ ausgeglichen werden, sodass die Agenten einen Konsensus erreichen und dieser Konsensus dem Mittelwert der Anfangszustände der Agenten $x_i(0)$ für $i \in \mathcal{I}$ entspricht. Folglich ist der Grenzwert unabhängig von der Kommunikationsstruktur. Zusätzlich konvergiert das Push-Sum Verfahren nach [65] auch auf zeitvarianten Kommunikationsstrukturen, vorausgesetzt der zeitvariante Graph ist B -stark verbunden, siehe Definition 2.5.3.

2.6.2 Mittelwertkonsensus bei reihen-stochastischer Gewichtung

Im Gegensatz zur spalten-stochastischen Gewichtung gestaltet es sich für reihen-stochastische Matrizen schwieriger, ein Konsensusprotokoll zu erstellen, das gegen den Mittelwert der Anfangszustände konvergiert. Grund hierfür ist, dass die fehlende Symmetrie in der Kommunikation sich in einer Gewichtung der Summe niederschlägt, die nicht so einfach heraus dividiert werden kann, wie es für spalten-stochastische Gewichtungen der Fall ist. Die Problematik soll im Folgenden kurz motiviert werden. Sei die Kommunikationsmatrix W reihen-stochastisch gewichtet. Es ist möglich, den linken Eigenvektor über die Vorschrift

$$Y(k+1) = WY(k) \quad (2.59)$$

zu berechnen, wobei die Matrix Y als Identitätsmatrix $I_{N \times N}$ initialisiert wird. Nach dem Perron-Frobenius-Theorem und der reihen-stochastischen Gewichtung entsprechend, ergibt sich für den Grenzwert

$$\lim_{k \rightarrow \infty} Y(k) = \mathbf{1} w^T Y(0) = \mathbf{1} w^T. \quad (2.60)$$

Entsprechend ist es mit dieser Vorschrift für jeden Agenten möglich, den linken Eigenvektor des Systems zu schätzen. Es ist allerdings nicht möglich, wie es für das Push-Sum Verfahren der Fall ist, die Gewichtung am Ende des Verfahrens auszugleichen und damit parallel den Eigenvektor und den Konsensus zu schätzen. Ist allerdings der linke Eigenvektor bereits vor dem Start der Konsensusiteration

$$x(k+1) = Wx(k) \quad (2.61)$$

bekannt, können die Startwerte mit dem Gewicht $\tilde{x}_i(0) = \frac{1}{Nw_i}x_i(0)$ versehen werden, damit der Mittelwertkonsensus erreicht wird. Denn unter dieser Vorgewichtung gilt für den Grenzwert

$$\lim_{k \rightarrow \infty} x(k) = \mathbf{1} \sum_{i=1}^N w_i \tilde{x}_i(0) = \mathbf{1} \frac{1}{N} \sum_{i=1}^N x_i(0). \quad (2.62)$$

Problematisch wird diese Strategie allerdings bei zeitvarianten Kommunikationsgraphen, da sich auch der Eigenvektor der Kommunikationsmatrix entsprechend ändert.

3 Optimaler Einsatzplan für nicht-kooperierende Microgrids

Das primäre Ziel von Einsatzplanproblemen (engl.: economic dispatch problems) in Stromnetzwerken ist es, die Stromversorgung der Verbraucher im Netz über einen bestimmten Planungszeitraum hinweg sicherzustellen. Um dieses Ziel zu erreichen, muss ein Einsatzplan für die aktiven Bestandteile des Netzwerkes, z. B. der Generatoren und Energiespeicher, erstellt werden. Dieser Einsatzplan legt fest, wie viel Leistung die Generatoren zu welchem Zeitpunkt produzieren bzw. wie die Speicherstrategien im betrachteten Zeitraum festzulegen sind, sodass ein fest oder variabel definierter Verbrauch versorgt wird. Um gleichzeitig die Kosten der Stromversorgung zu minimieren, wird das Erstellen des Einsatzplans als Optimierungsproblem formuliert.

Nachfolgend wird ein Simulationsmodell vorgestellt, welches ein Einsatzplanproblem zwischen Microgrids über einen bestimmten Planungshorizont abbildet. Als Microgrid, wie beispielhaft in Abbildung 3.1 abgebildet, wird ein Subnetzwerk des elektrischen Netzes bezeichnet, welches bei entsprechender Kapazität autark betrieben werden kann, aber dennoch eine Verbindung zum Hauptnetzwerk besitzt, siehe auch [11].

Das Modell wird als agentenbasiertes, verteiltes Optimierungsproblem formuliert, bei dem die Agenten die Bestandteile des Netzwerkes repräsentieren. Diese Bestandteile treffen, je nach Zugehörigkeit zu einem Microgrid als Organisationseinheit, kooperativ oder nicht-kooperativ mit anderen Agenten eine optimale Entscheidung bezüglich ihres Einsatzes. Hierbei wird zusätzlich die Zugehörigkeit der Optimierungsteilnehmer zu bestimmten kooperativen Gruppen berücksichtigt.

Das nachfolgend beschriebene Modell basiert weitestgehend auf bekannten Problemstellungen aus relevanter Literatur. Erweiterungen bezüglich bestehender Problemstellung werden entsprechend hervorgehoben und motiviert.

3.1 Allgemeine Beschreibung der Problemstellung

In Abbildung 3.1 ist das betrachtete Netzwerk, für welches das Einsatzplanproblem formuliert werden soll, beispielhaft dargestellt. Die farblich voneinander unterschiedenen Ellipsen stehen für die jeweiligen Microgrids. Das Hauptnetzwerk wird durch das Gitter in der Mitte der Microgrids repräsentiert. Die Verbindungslinien innerhalb der Microgrids bzw. zwischen den Microgrids und dem Hauptnetzwerk stellen Übertragungsleitungen dar. Das System besteht aus zwei Ebenen: Der unteren Ebene,

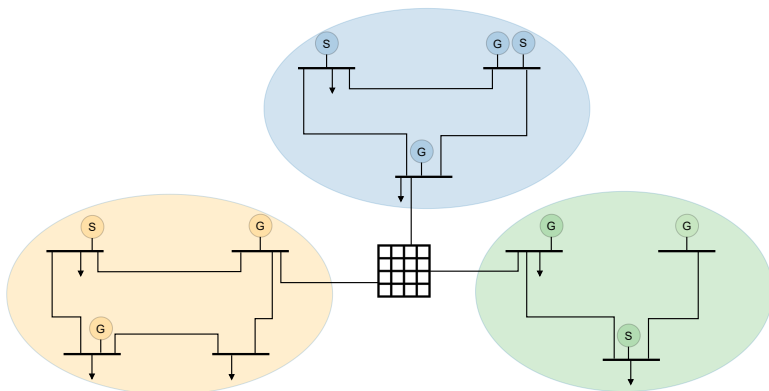


Abbildung 3.1: Gesamtsystem bestehend aus einer Schnittstelle zum Hauptnetzwerk und drei Microgrids mit Generatoren (G), Speichern (S) und Lasten (Pfeile) sowie dem Busnetzwerk, welches die einzelnen Bestandteil des Microgrids verbindet.

welche das Innere eines Microgrids abbildet und der oberen Ebene, auf der die Microgrids mit einer Schnittstelle zum Hauptnetz verbunden sind. Die Microgrids bestehen jeweils aus Generatoren, Speichern und Verbrauchern, die an Busse angeschlossen sind. Diese stellen die Bilanzknoten im Netzwerk dar. Die Bussystemdarstellung ist üblich für Leistungsflussberechnungen, siehe beispielsweise [66][67].

Die Microgrids werden von unterschiedlichen Unternehmen betrieben, welche die Speicher und Generatoren des jeweiligen Subnetzwerkes besitzen. Ziel jedes dieser Unternehmen ist es, die Leistungsnachfrage im eigenen Microgrid zu decken, indem Leistung mit den eigenen Generatoren erzeugt wird, die eigenen Speicher ent- bzw. geladen werden und/oder Leis-

tung vom Hauptnetz gekauft wird. Somit entspricht das Microgrid einem abgegrenzten Leistungsbilanzgebiet im Netzwerk.

Die Optimierung erfolgt über die Zeitslots $t \in \mathcal{T} = \{1, \dots, T\}$ bis zu einem festgelegten, endlichen Zeithorizont T . Soll beispielsweise der Einsatzplan für den nächsten Tag festgelegt werden, kann ein Zeitslot t einer Stunde entsprechen und der Horizont $T = 24$ festgelegt werden. Die Leistungsnachfrage gilt genau dann als gedeckt, wenn die Leistungsbilanzen, d. h. produzierte, gespeicherte oder die über das Hauptnetz in das Microgrid eingespeiste Leistung für alle Busse des eigenen Microgrids in jedem betrachteten Zeitslot t ausgeglichen sind. Zusätzlich sollen die Kosten für die Bedienung der Leistungsnachfrage minimiert werden. Um diese Minimierung zu erreichen, arbeiten alle aktiven Bestandteile des jeweiligen Microgrids zusammen. Explizit sollen dabei die gemeinsamen Kosten, die von dem das Microgrid betreibende Unternehmen getragen werden, minimiert werden. Es ergibt sich somit ein kooperatives Optimierungsproblem innerhalb eines jeden Microgrids.

Auf der oberen Ebene sind die Microgrids über ein gemeinsames Verbindungsterminal zum Hauptnetz gekoppelt. Das Hauptnetz verkauft Leistung an die jeweiligen Microgrids, wobei angenommen wird, dass der hierfür anfallende Preis pro Kilowattstunde mit der Nachfrage aller Microgrids ansteigt. Damit hat die Menge der nachgefragten Leistung von Microgrid i Einfluss auf die Kosten der restlichen Microgrids im betrachteten Netzwerk. Wird davon ausgegangen, dass die Microgrids bzw. deren Betreiber nicht miteinander kooperieren, ergibt sich somit ein nicht-kooperatives, spieltheoretisches Problem zwischen den Microgrids. Dieses Spiel wird als gelöst betrachtet, wenn die Strategien der Spieler bzw. der Microgrids, sich zueinander in einem Nash-Gleichgewicht befinden (siehe Abschnitt 2.4), von dem kein Spieler, d. h. kein Microgridbetreiber, einen Anreiz hat abzuweichen.

Die beschriebene Problemstellung soll als verteiltes Optimierungsproblem formuliert und gelöst werden. Jeder aktive Optimierungsteilnehmer, d. h. jeder Agent des Multiagentensystem, besitzt eine eigene Kostenfunktion, die nur diesem Agenten bekannt ist und die darüber hinaus privat bleiben soll. Zusätzlich sind die Agenten in der Wahl ihrer Strategie durch lokale und globale Nebenbedingungen eingeschränkt. Als lokale Nebenbedingungen werden diejenigen Restriktionen bezeichnet, die nur den jeweiligen Agenten betreffen, nur diesem bekannt sind und auch während des Optimierungsverlaufes privat bleiben sollen. Globale Nebenbedingungen des Optimierungsproblems sind allen Agenten des jeweiligen Microgrids

bekannt, allerdings nicht den Agenten der anderen Microgrids. Durch die verteilte Lösung des Optimierungsproblems entfällt die Notwendigkeit eines zentralen Koordinators, der die gesamte Information des Netzwerkes sammelt. Würde ein solcher zentraler Knoten existieren, könnte bei einem Angriff auf diesen Knoten die gesamte Information des Netzwerkes abgegriffen und eine Versorgung der Verbraucher verhindert werden. Bei verteilter Lösung des Optimierungsproblems hingegen kann der Ausfall eines Agenten und damit einer Recheneinheit erkannt und mit dem Ausschluss des defekten Agenten reagiert werden. Welche Agenten welche Komponenten der Kostenfunktion berücksichtigen, wird in Abschnitt 3.3 beschrieben.

Im Folgenden werden zunächst alle Bestandteile des Systems vorgestellt und modelliert, bevor das Gesamtproblem mathematisch formuliert wird. Anschließend erfolgt eine Einordnung des Modells in relevante Literatur und abschließend werden die Herausforderungen bezüglich der Lösung des Problems hervorgehoben.

3.2 Bestandteile der Microgrids

3.2.1 Verbraucher

Verbraucher werden in der Abbildung 3.1 als Lasten an den jeweiligen Bussen des Netzwerkes repräsentiert. Es wird angenommen, dass der Verbrauch $v(t)$ an einem beliebigen Bus in jedem Zeitslot t zu Beginn des Optimierungsprozesses bekannt ist und auch während des Prozesses nicht verändert werden kann. Es existieren Varianten des Einsatzplanproblems, siehe zum Beispiel [55][68], bei denen der Verbrauch in einem gewissen Rahmen variabel gestaltet ist. Das bedeutet, dass Verbraucher beispielsweise Lasten gezielt abschalten bzw. deren Einsatz in einen anderen Zeitslot verschieben können, um Flexibilität zu erzeugen und so teure Strompreisspitzen zu vermeiden. Für jeden Verbraucher wird hierbei eine konkave Nutzenfunktion definiert, die ihren maximalen Wert annimmt, wenn der aktuelle Bedarf komplett gedeckt wird und absinkt, sobald eine Verschiebung oder ein Abschalten einer Last getätigt wird und somit der Verbrauch $v(t)$ im aktuellen Zeitslot sinkt. Ziel der Verbraucher ist es hierbei ihren Nutzen zu maximieren. In der vorliegenden Arbeit wird die Flexibilität bezüglich der Lastverschiebung durch die verwendeten Speicher, siehe Unterabschnitt 3.2.3, gewährleistet.

3.2.2 Dezentrale Generatoren

Es wird zwischen regulierbaren Generatoren, deren Leistungsproduktion steuerbar ist, und nicht regulierbaren Generatoren mit konstantem Leistungsgenerierung unterschieden. Zu ersteren werden brennstoffbasierte Generatoren, wie beispielsweise Gaskraftwerke gezählt, deren Produktion sich vereinfacht betrachtet durch die Zugabe oder Reduzierung des Brennstoffmaterials regulieren lässt. Der Verbrauch von Brennstoff geht mit Kosten für die Beschaffung einher, die im Einsatzplanproblem berücksichtigt sind. Klassischerweise werden quadratische Kostenfunktionen der Form

$$f^G(g(t)) = a_G g(t)^2 + b_G g(t) + c_G \quad (3.1)$$

verwendet, wobei $g(t)$ die Leistungsproduktion des Generators zum Zeitpunkt t ist, siehe beispielsweise [68][69]. Die konstanten Koeffizienten $a_G, b_G, c_G > 0$ der Kostenfunktion unterscheiden sich je nach Brennstoff, Bauart und Größe des Kraftwerkes. Zusätzlich unterliegt die Produktion der regulierbaren Generatoren unteren und oberen Schranken $\underline{g}(t)$ und $\bar{g}(t)$. Untere Schranken sind beispielsweise dann vonnöten, wenn der Betrieb unter einer bestimmten Grenze nicht wirtschaftlich ist oder eine lange Hochfahrzeit des Kraftwerkes nach komplettem Stillstand vermieden werden soll. Die obere Schranke ergibt sich zum einen aus den physikalischen Kapazitäten des Generators und ebenfalls einer möglichen wirtschaftlichen Beschränkung. Die Beschränkungsmenge eines regulierbaren Generators kann folglich durch

$$\mathcal{X}^G = \{g(t) | \underline{g}(t) \leq g(t) \leq \bar{g}(t), \forall t \in \mathcal{T}\} \quad (3.2)$$

beschrieben werden.

Wird der Einsatzplan über mehrere Zeitslots geplant, können zusätzlich Rampenbeschränkungen (engl.: ramp rate constraints) berücksichtigt werden, die sicherstellen, dass sich die produzierte Generatorleistung zwischen zwei Zeitslots nur in einem bestimmten Rahmen ändern kann. Folglich gilt die Beschränkung $|g(t-1) - g(t)| \leq g^R$. Möglich sind auch unterschiedliche Beschränkungen bezüglich der Abnahme und des Anstieges der Generatorleistung [69].

Im Gegensatz zu regulierbaren Generatoren haben beispielsweise Windkraft- und Solaranlagen keine Brennstoffkosten. Deren abgegebene Leistung ist von den Witterungsverhältnissen abhängig und kann somit,

wenn überhaupt, nur geringfügig reguliert werden⁶. Folglich fallen ausschließlich konstante Kosten an, die durch eine Optimierung nicht beeinflusst werden können. Daher wird in der vorliegenden Arbeit auf eine explizite Modellierung von nicht regulierbaren Kraftwerken verzichtet. Zusätzlich wird die Wahrscheinlichkeitsverteilung der Prädiktion der Leistungsproduktion von beispielsweise wetterabhängigen Kraftwerken nicht berücksichtigt und ausschließlich mit deterministischen Werten gerechnet⁷.

3.2.3 Dezentrale Speicher

Energiespeicher oder Batterien, im Folgenden als Synonym verwendet, ermöglichen eine Verschiebung von Leistung unter Verwendung der Speicherkapazitäten von einem Zeitslot in einen anderen und sind somit ein beliebtes Mittel, siehe [16][17][34], um Flexibilität in das Optimierungsproblem zu integrieren. Dadurch können teure Strompreisspitzen vermieden werden. Obwohl die Speicher selbst keine speicherstrategieabhängige Betriebskostenfunktion, wie beispielsweise die regulierbaren, brennstoffbasierten Generatoren, besitzen, können dennoch quadratische Kosten

$$f^S(s(t)) = a_S s(t)^2 \quad (3.3)$$

mit positivem Koeffizienten $a_S > 0$ in Abhängigkeit der Speicherstrategie $s(t)$ eingeführt werden, durch die eine hohe Beanspruchung des Speichers mit virtuellen Kosten bestraft wird. Damit kann indirekt die Vermeidung des vorzeitigen Alterns einer Batterie durch hohe Beanspruchung in das Optimierungsproblem integriert werden [17].

Im Folgenden wird ein vereinfachtes Speichermodell eingeführt, das als Nebenbedingungsmenge in dem Optimierungsproblem berücksichtigt wird. Ähnliche Modellierungen finden sich auch in [16] und [17]. Gilt $s(t) > 0$, wird der Speicher um $|s(t)|$ entladen, wenn $s(t) < 0$ wird die entsprechende

⁶Bei Windkraftanlagen kann beispielsweise die Ausrichtung der Rotorblätter bezüglich der Windrichtung angepasst werden, um die Energieerzeugung einer Windfarm zu maximieren oder reduzieren [70]. In der optimalen Einsatzplanung kann davon ausgegangen werden, dass alle Windkraftanlagen in ihrer optimalen Stellung betrieben werden, um die CO₂ Gesamtproduktion zu verringern und Kosten zu sparen.

⁷Eine Möglichkeit der Inkorporation von Unsicherheiten bezüglich der Prädiktion in die Leistungsbalance des Microgrids stellen sogenannte Chance Constraints dar, mit denen garantiert werden kann, dass eine Beschränkung mit einer spezifizierten Wahrscheinlichkeit eingehalten wird. Diese Beschränkungen können unter Annahme bestimmter Wahrscheinlichkeitsverteilungen, wie z. B. einer Normalverteilung, konvex deterministisch approximiert werden [71].

Leistung in den Speicher eingespeist. Die Leistungsmenge, die dem Speicher entnommen werden kann, ist abhängig von der aktuellen Ladung $q(t)$ des Speichers, die physikalisch durch die Kapazität des Speichers beschränkt ist, sodass $0 \leq q(t) \leq \bar{q}$. Folglich muss für die Speicherstrategie $-q(t) \leq -s(t) \leq \bar{q} - q(t)$ in jedem Zeitslot t gelten. Zusätzlich wird davon ausgegangen, dass die Speicher nicht ideal sind und daher zwischen den Zeitslots Ladung verlieren und somit $q(t+1) = \gamma q(t) - s(t)$ mit $0 < \gamma < 1$ gilt. Dieser Effekt wird im Englischen als Leakage bezeichnet, der zugehörige Faktor γ ist dabei von der Art des Speichers abhängig⁸. Unter Berücksichtigung dieser Faktoren gilt für die Ladung eines Speichers zu Beginn des Zeitslots t

$$q(t) = \gamma^{t-1} q(1) - \sum_{r=1}^{t-1} \gamma^{t-r} s(r), \quad (3.4)$$

wobei $q(1)$ der Ladung des Speichers zu Beginn des betrachteten Zeitintervalls \mathcal{T} entspricht. Damit muss für jeden Zeitslot t die Beschränkung $-\gamma^{t-1} q(1) \leq -\sum_{r=1}^t \gamma^{t-r} s(r) \leq \bar{q} - \gamma^{t-1} q(1)$ gelten. Zusätzlich soll sichergestellt werden, dass die Optimierung den Speicher zum Ende des berücksichtigten Zeitraums nicht zu Lasten des nächsten Planungszeitraums komplett entleert. Daher wird die Beschränkung $|q(T) - q(1)| \leq \epsilon$ eingeführt, wobei $\epsilon > 0$ eine entsprechend kleine Konstante ist. Alle beschriebenen Beschränkungen können nun in der Menge

$$\mathcal{X}^S := \left\{ s(t) \mid \begin{array}{l} \underline{s}(t) \leq s(t) \leq \bar{s}(t), |q(T) - q(1)| \leq \epsilon, \\ -\gamma^{t-1} q(1) \leq -\sum_{r=1}^t \gamma^{t-r} s(r) \leq \bar{q} - \gamma^{t-1} q(1), \\ \forall t \in \mathcal{T}, \end{array} \right\} \quad (3.5)$$

zusammengefasst werden. Wie erkannt werden kann, sind die Nebenbedingungen bezüglich der Zeitslots $t \in \mathcal{T}$ miteinander verkoppelt.

3.2.4 Übertragungsnetzwerk

Für die Modellierung von Leistungsflüssen in elektrischen Netzwerken werden Busnetzwerke verwendet, siehe beispielsweise [66]. Solche Netzwerke sind in Abbildung 3.1 innerhalb der Microgrids dargestellt. Zwischen den Bussen, die in der Abbildung als horizontale Balken dargestellt sind, verlaufen die Übertragungslinien, abgebildet als ungerichtete Verbindungen, des Netzwerkes, über die Leistung zwischen den jeweils angrenzenden

⁸In [16] wird der Faktor mit etwa $\gamma = 0.99$ angegeben.

Bussen übertragen werden kann. Die Busse selbst repräsentieren Schaltanlagen, an denen dezentrale Generatoren, Speicher und Verbraucher eines Gebietes zusammengefasst werden. Zusätzlich fungieren sie als Balancierungsknoten, an denen sich die generierte, verbrauchte und über die Übertragungsleitungen eingebrachte oder abgegebene Leistung ausgleichen muss. Das Problem, Leistung zwischen den Bussen zu übertragen und dabei physikalische Randbedingungen der Leitungen zu berücksichtigen, wird als Leistungsflussproblem (engl.: power flow) bezeichnet. Diese Probleme unterscheiden sich je nach Detailgrad der Modellierung der Übertragungsleitungen bzw. der Busse. Die nachfolgende Modellierung basiert auf dem linearisierten (DC) Leistungsflussproblem, siehe beispielsweise [72]. Eine genauere Einordnung der Modellierung in die Literatur erfolgt in Abschnitt 3.4.

Die Übertragungsleitungen zwischen den Bussen lassen sich durch einen ungerichteten Graphen $\mathcal{G}^{\text{B}}\{\mathcal{V}^{\text{B}}, \mathcal{E}^{\text{B}}\}$ beschreiben, dessen Knoten $v_i \in \mathcal{V}^{\text{B}}$ die jeweiligen Busse i repräsentieren und die Kanten (i, j) bzw. (j, i) die Verbindung zwischen den Bussen i, j , insofern diese existiert. Für das grüne Beispielmicrogrid aus Abbildung 3.1 ergibt sich folglich die zum entsprechenden Graphen \mathcal{G}^{B} zugehörige ungewichtete Adjazenzmatrix

$$A^{\text{B}} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \quad (3.6)$$

Die Busse sind hierbei von links nach rechts durchnummeriert und entsprechen den jeweiligen Zeilen und Spalten. Der Knoten v_4 entspricht dem Terminal zum Hauptnetzwerk.

Der Leistungsfluss zwischen zwei Bussen i, j wird durch die Variablen e_{ij} bzw. e_{ji} beschrieben. Da Leistung allerdings nicht gleichzeitig in beide Richtungen fließen kann und somit $e_{ij} = -e_{ji}$ gelten muss, lässt sich der Leistungsfluss auf einer Verbindung auch durch eine einzige Variable und deren Vorzeichen beschreiben und somit die Anzahl der festzulegenden Variablen reduziert. Dafür wird aus dem ungerichteten Graphen \mathcal{G}^{B} ein gerichteter $\bar{\mathcal{G}}^{\text{B}}\{\mathcal{V}^{\text{B}}, \bar{\mathcal{E}}^{\text{B}}\}$ erzeugt, beispielsweise indem die untere Dreiecksmatrix von A^{B} zu null gesetzt wird. Existiert eine gerichtete Verbindung von Bus i zu Bus j , d. h. $e_{ij} \in \bar{\mathcal{E}}^{\text{B}}$, und es fließt Leistung von Bus i zu Bus j entlang der Richtung der Leitung, dann gilt $e_{ij} > 0$. Fließt Leistung entgegen der Richtung der Leitung von Bus j nach Bus i , so gilt $e_{ij} < 0$. Diese Zählrichtung muss bei der Berechnung der Leistungsbalance am jeweiligen

Knoten berücksichtigt werden. So gilt beispielsweise für die Leistungsbalance am Bus i , an den ein Generator und ein Speicher angeschlossen sind, für alle Zeitslots $t \in \mathcal{T}$, dass

$$g_i(t) + s_i(t) - \sum_{j \in \vec{\mathcal{N}}_i^-} e_{ij}(t) + \sum_{j \in \vec{\mathcal{N}}_i^+} e_{ji}(t) = v_i(t), \quad (3.7)$$

wobei hier $\vec{\mathcal{N}}_i^-$ und $\vec{\mathcal{N}}_i^+$ Ausgangs- und Eingangsnachbarschaft des Busses i bezüglich des gerichteten Graphen $\vec{\mathcal{G}}^B$ entsprechen. Der Verbrauch $v_i(t)$ des Busses i wird, wie in Unterabschnitt 3.2.1 beschrieben, in jedem Zeitslot t als konstant und unveränderlich angenommen.

Ist $j \in \vec{\mathcal{N}}_i^-$, d. h. es existiert eine gerichtete Verbindung von i nach j , und e_{ij} ist positiv, so fließt Leistung von Bus i nach Bus j und muss folglich von Bus i bereitgestellt werden. Ist e_{ij} dagegen negativ, fließt Leistung von j nach i und schlägt sich positiv in der Leistungsbilanz von Bus i nieder.

Jede generische Leitung besitzt eine maximale Übertragungskapazität \bar{e} , die nicht überschritten werden darf. Folglich muss für jede Leitung die im Zeitslot t übertragene Leistung $e(t) \in \mathbb{R}$ der Beschränkungsmenge

$$\mathcal{X}^L = \{e(t) \mid -\bar{e}(t) \leq e(t) \leq \bar{e}(t), \forall t \in \mathcal{T}\} \quad (3.8)$$

entsprechen.

Für die Übertragung der Leistung $e(t)$ über eine gerichtete Verbindung zwischen zwei Bussen müssen die an die Leitung angrenzenden Busse gemeinsam die Kosten

$$f^L(e(t)) = a_L e(t)^2 \quad (3.9)$$

tragen, wobei $a_L > 0$ gilt. Durch diese Kosten können zum einen Übertragungsentgelte für die Netzbetreiber realisiert werden. Zum anderen kann mit diesen Kosten eine Reduzierung der übertragenen Leistung erreicht werden, insofern es die Nebenbedingungen des Optimierungsproblems erlauben. Hierdurch reduzieren sich indirekt ebenfalls Leistungsverluste, die durch die Übertragung auf den Leitungen entstehen.

Die Übertragungsleitungen zwischen dem Hauptnetzterminal und den jeweiligen Bussen der Microgrids werden so wie die Verbindungen innerhalb der Microgrids berücksichtigt. Dazu kann das Terminal als Knoten in den gerichteten Graphen $\vec{\mathcal{G}}^B$ und eine gerichtete Verbindung vom Terminal zum jeweiligen Bus im Microgrid etabliert werden. Auch die Kosten für die Übertragung werden der Kostenfunktion (3.9) entsprechend verarbeitet.

3.2.5 Hauptnetz

Die Schnittstelle zum Hauptnetz stellt neben den dezentralen Generatoren eine weitere Möglichkeit für die Microgrids dar, Leistung zu beziehen. Der Preis, der für die bezogene Leistung bezahlt werden muss, ist dabei nachfrageabhängig und richtet sich nach der vom gesamten Netzwerk nachgefragten Leistungsmenge $p(t)$. Hierfür wird, wie auch in [17], die affine Preisfunktion

$$P(p(t)) = b_P p(t) \quad (3.10)$$

mit positiver Konstante $b_P > 0$ verwendet, wobei $p(t) = \sum_{h=1}^H p^h(t)$ der Summe der von den Microgrids $h \in \mathcal{H} = \{1, \dots, H\}$ nachgefragten Leistungsmenge $p^h(t)$ entspricht. Diese Leistungsmenge $p^h(t)$ ist beschränkt durch $0 \leq p^h(t) \leq \bar{p}^h(t)$, $\forall t \in \mathcal{T}$. Hierbei stellt $\bar{p}^h(t)$ die Kapazität der betreffenden Leitung dar. Da es sich hierbei im Grunde um eine Leitungsbeschränkung handelt, ist diese in die Beschränkungsmenge \mathcal{X}^B aus Gleichung (3.8) integriert und wird daher nicht als eigenständige Beschränkungsmenge geführt. Die Beschränkung $0 \leq p^h(t)$ legt fest, dass Leistung vom Hauptnetzwerk nur gekauft, aber nicht Leistung an das Hauptnetzwerk verkauft werden kann.

3.3 Formulierung des Optimierungsproblems

Es wird ein System bestehend aus insgesamt N Agenten betrachtet, die sich in der Menge $\mathcal{I} = \{1, \dots, N\}$ befinden. Jeder Agent i repräsentiert einen Bus der Subsysteme innerhalb der Microgrids, wie in Unterabschnitt 3.2.4 beschrieben. Die Menge der Agenten ist auf insgesamt H Microgrids aufgeteilt, die alle Teil der Menge $\mathcal{H} = \{1, \dots, H\}$ sind, sodass jedes Microgrid insgesamt aus N^h Agenten besteht. Die Menge der Agenten, die zum Microgrid h gehören, wird durch \mathcal{I}^h beschrieben. Kein Agent kann Teil von mehreren Microgrids sein, $\mathcal{I}^h \cap \mathcal{I}^l = \emptyset$, $\forall h, l \in \mathcal{H}, h \neq l$ und jedes Microgrid besteht aus mindestens einem Agenten, sodass $\mathcal{I}^h \neq \emptyset$. Die Busnetzwerke innerhalb der Microgrids $h \in \mathcal{H}$ werden durch die gerichteten Graphen $\vec{\mathcal{G}}_B^h(\mathcal{V}_B^h, \vec{\mathcal{E}}_B^h)$ repräsentiert, wobei $\mathcal{V}_B^h \equiv \{\mathcal{I}^h, \text{HN}\}$ mit HN als Entität des Hauptnetzes.

Struktur und Anzahl der lokalen Kostenfunktionen und Nebenbedingungen der Agenten sind davon abhängig, ob Generatoren, Speicher und Lasten an den jeweiligen Bus angeschlossen sind. Auch die Anzahl der an den Bus angeschlossenen Übertragungsleitungen bestimmt Anzahl und

Form der lokalen Funktionen. Es wird davon ausgegangen, dass der ungerichtete Graph \mathcal{G}_B^h , der das Übertragungsnetz in Microgrid h repräsentiert, verbunden ist, sodass keine isolierten Busse existieren⁹.

Für die nachfolgende Beschreibung des Optimierungsproblems wird aus Gründen der Übersichtlichkeit davon ausgegangen, dass sich an jedem Bus ein Generator, ein Speicher und eine Last befinden. Zusätzlich wird angenommen, dass die Kosten

$$f^{h,P}(\mathbf{p}(t)) = P(p(t))p^h(t) = \left(b_P \sum_{h=1}^H p^h(t) \right) p^h(t), \quad (3.11)$$

die durch den Bezug der Leistung $p^h(t)$ vom Hauptnetz anfallen, gleichmäßig auf alle Busse des jeweiligen Microgrids h aufgeteilt werden¹⁰. Damit besitzt der Bus $i \in \mathcal{I}^h$ des Microgrids h in Zeitslot t die lokale Kostenfunktion

$$\begin{aligned} f_i^h(\mathbf{p}(t), g_i^h(t), s_i^h(t), \mathbf{e}_i^h(t)) &= \frac{1}{N^h} f^{h,P}(\mathbf{p}(t)) + f_i^G(g_i^h(t)) + f_i^S(s_i^h(t)) \\ &+ \frac{1}{2} \sum_{j \in \bar{\mathcal{N}}_i^-} f_i^L(e_{ij}^h(t)) + \frac{1}{2} \sum_{j \in \bar{\mathcal{N}}_i^+} f_i^L(e_{ji}^h(t)) \end{aligned} \quad (3.12)$$

wobei $\mathbf{p}(t) = \text{col}((p^h(t))_{h \in \mathcal{H}})$. Der Vektor

$$\mathbf{e}_i^h(t) = \text{col} \left(\text{col} \left((e_{ij}^h(t))_{j \in \bar{\mathcal{N}}_i^-} \right), \text{col} \left((e_{ji}^h(t))_{j \in \bar{\mathcal{N}}_i^+} \right) \right)$$

enthält die ein- und ausgehenden Leistungsflüsse am Bus i , $g_i^h(t)$ und $s_i^h(t)$ repräsentieren die generierte Leistung bzw. die Speicherstrategie. Die quadratischen Kostenfunktionen f_i^G , f_i^S und f_i^L sind in den Gleichungen (3.1), (3.3) und (3.9) definiert. Durch den Index i können die Konstanten $a_{\cdot,i}$, $b_{\cdot,i}$, $c_{\cdot,i}$ der Kostenfunktionen an den verschiedenen Bussen voneinander unterschieden werden. Der Faktor $\frac{1}{2}$ vor den Kostenfunktionen f_i^L wird eingeführt, da immer jeweils zwei Busse durch eine Leitung verbunden sind und sich kooperativ die Kosten teilen.

⁹Isolierte Busse können beispielsweise auftreten, wenn eine Übertragungsleitung defekt ist. Da diese Busse allerdings nicht an der Lösung des verteilten Optimierungsproblems mitwirken können, werden sie durch die Annahme der Verbundenheit der Graphen \mathcal{G}_B^h , $h \in \{1, \dots, H\}$, ausgeschlossen.

¹⁰Eine weitere Möglichkeit ist, dass nur der Bus mit Verbindung zum Hauptnetz die Kosten trägt. Da bei kooperativer Lösung des Microgridproblems über die Kosten aller Busse summiert wird, erzeugen beide Varianten äquivalente Optimierungsprobleme.

Diese Kosten müssen von dem jeweiligen Microgrid h in jedem Zeitslot $t \in \mathcal{T}$ getragen werden. Die Gesamtkosten über einen Zeithorizont von T ergeben sich somit zu

$$f_i^h(\mathbf{p}, \mathbf{g}_i^h, \mathbf{s}_i^h, \mathbf{e}_i^h) = \sum_{t=1}^T f_i^h(\mathbf{p}(t), g_i^h(t), s_i^h(t), \mathbf{e}_i^h(t)) \quad (3.13)$$

mit $\mathbf{p} = \text{col}(\mathbf{p}^h, \mathbf{p}^{-h})$, $\mathbf{p}^h = \text{col}((p^h(t))_{t \in \mathcal{T}})$, $\mathbf{g}_i^h = \text{col}((g_i^h(t))_{t \in \mathcal{T}})$, $\mathbf{s}_i^h = \text{col}((s_i^h(t))_{t \in \mathcal{T}})$, $\mathbf{e}_i^h = \text{col}((\mathbf{e}_i^h(t))_{t \in \mathcal{T}})$.

Die Beschränkungsmengen der Busse $i \in \mathcal{I}^h$ in Microgrid $h \in \mathcal{H}$ ergeben sich entsprechend den in den Unterabschnitten 3.2.1 - 3.2.5 formulierten Beschränkungen der einzelnen Bestandteile des Microgrids sowie der Balancebedingung

$$\mathcal{X}_i^{\text{B}} = \left\{ \mathbf{g}_i^h, \mathbf{s}_i^h, \mathbf{e}_i^h \mid g_i^h(t) + s_i^h(t) - \sum_{j \in \bar{\mathcal{N}}_i^-} e_{ij}^h(t) + \sum_{j \in \bar{\mathcal{N}}_i^+} e_{ji}^h(t) = v_i^h(t), \forall t \in \mathcal{T} \right\} \quad (3.14)$$

nach Gleichung (3.7) zu

$$\mathcal{X}_i^h = \mathcal{X}_i^{\text{G}} \cup \mathcal{X}_i^{\text{S}} \cup \mathcal{X}_i^{\text{L}} \cup \mathcal{X}_i^{\text{B}}. \quad (3.15)$$

Das Gesamtoptimierungsproblem des Betreibers des Microgrids h bildet sich aus der Summierung der Kosten für die einzelnen Kostenfunktionen der Busse und aus der Vereinigung der Nebenbedingungsmengen. Da die Kosten für die Befriedigung der Leistungsnachfrage minimiert werden sol-

len, ergibt für das Microgrid h :

$$\min_{x^h} F^h(x^h, x^{-h}) = \min_{x^h} \sum_{i=1}^{N^h} f_i^h(\mathbf{p}, \mathbf{g}_i^h, \mathbf{s}_i^h, \mathbf{e}_i^h), \quad (3.16a)$$

$$\text{s.t. } g_i^h(t) + s_i^h(t) - \sum_{j \in \mathcal{N}_i^-} e_{ij}^h(t) + \sum_{j \in \mathcal{N}_i^+} e_{ji}^h(t) = v_i(t), \quad (3.16b)$$

$$\underline{g}_i^h(t) \leq g_i^h(t) \leq \bar{g}_i^h(t), \quad (3.16c)$$

$$0 \leq p^h(t) \leq \bar{p}^h(t), \quad (3.16d)$$

$$\underline{s}_i^h(t) \leq s_i^h(t) \leq \bar{s}_i^h(t), \quad (3.16e)$$

$$|q_i^h(T) - q_i^h(1)| \leq \epsilon_{i,h}, \quad (3.16f)$$

$$-\gamma_{i,h}^{t-1} q_i^h(1) \leq -\sum_{r=1}^t \gamma_{i,h}^{t-r} s_i^h(r) \leq \bar{q}_i^h - \gamma_{i,h}^{t-1} q_i^h(1), \quad (3.16g)$$

$$-\bar{e}_{ij}(t) \leq e(t)_{ij} \leq \bar{e}_{ij}(t), \forall i, j \in \bar{\mathcal{E}}_B^h, \quad (3.16h)$$

$$\forall t \in \mathcal{T}, \forall i \in \mathcal{I}^h.$$

Die Variable x^h beschreibt den gesamten Aktionsvektor von Microgrid h und beinhaltet die Komponenten $x^h = \text{col}(\mathbf{p}^h, \mathbf{g}^h, \mathbf{s}^h, \mathbf{e}^h)$ mit $\mathbf{g}^h = \text{col}((g_i^h)_{i \in \mathcal{I}^h})$, $\mathbf{s}^h = \text{col}((s_i^h)_{i \in \mathcal{I}^h})$ und $\mathbf{e}^h = \text{col}((e_i^h)_{i \in \mathcal{I}^h})$. Die Variable x^{-h} beschreibt entsprechend die Aktionen aller anderen Microgrids. Wie anhand der Kostenfunktionen $F^h(x^h, x^{-h})$ erkannt werden kann, hängen die Kosten von Microgrid h von Aktionen x^{-h} der anderen Microgrids ab. Im Speziellen stellt die Preisfunktion (3.10) das Kopplungselement des Problems dar, da diese von der Nachfrage aller Microgrids $\{\mathbf{p}^h\}_{h \in \mathcal{H}}$ abhängt.

Da, wie eingangs beschrieben, davon ausgegangen wird, dass die Microgrids nicht miteinander kooperieren, entsteht ein nicht-kooperatives, spieltheoretisches Problem zwischen den Microgrids, während das Problem innerhalb der jeweiligen Microgrids kooperativ gelöst werden soll. Beide Probleme sollen unter der Annahme von verteilter Information, d. h. unter der Annahme, dass die Kostenfunktion der einzelnen Agenten bzw. Busse privat sind, gelöst werden. Darüber hinaus können, je nach verwendetem Lösungsalgorithmus, ebenfalls die Beschränkungsmengen privat sein.

3.4 Einordnung in relevante Literatur

Im vorausgegangenen Abschnitt ist die Modellierung der einzelnen Komponenten des Problems in Bezug auf relevante Literatur erklärt. Hier soll nun eine Einordnung des Gesamtproblems erfolgen und Gemeinsamkeiten und Unterschiede hervorgehoben werden. Bei allen betrachteten Modellierungen wird davon ausgegangen, dass die Probleme verteilt auf einem Agentensystem definiert sind, wobei kein Agent das Gesamtproblem kennt.

3.4.1 Kooperative und nicht-kooperative Einsatzplanprobleme in der Literatur

Das klassische Einsatzplanproblem (engl. economic dispatch problem) beschreibt die Aufgabe, kooperativ einen optimalen Einsatzplan für eine Menge von N Generatoren zu finden, um einen festgelegten Bedarf D zu decken [73]. Es ergibt sich das Optimierungsproblem

$$\sum_{i=1}^N \min_{g_i} f_i^G(g_i), \quad (3.17a)$$

$$\text{s.t. } \underline{g}_i \leq g_i \leq \bar{g}_i, \forall i \in \{1, \dots, N\}, \quad (3.17b)$$

$$\sum_{i=1}^N g_i = D, \quad (3.17c)$$

siehe beispielsweise [74][75]. Jeder Generator i wird hierbei durch einen Agenten i repräsentiert, der die lokale Kostenfunktion f_i^G und die entsprechenden oberen und unteren Schranken für die Produktion als lokale Nebenbedingungen besitzt. Die Betrachtung beschränkt sich auf die Wirkleistung von Generatoren und Verbrauchern. Zusätzlich wird das System als ein Einbussystem modelliert, siehe Abbildung 3.2. Dabei wird davon ausgegangen, dass alle Generatoren sich am selben Bus wie die Last befinden, weswegen Übertragungsleitungen nicht berücksichtigt werden.

Leistungsflussprobleme (engl.: Optimal Power Flow), siehe beispielsweise [72][76], stellen eine Erweiterung der Einsatzplanprobleme dar, bei der die obige Problemstellung durch ein Modell des Übertragungsnetzwerkes als Nebenbedingung ergänzt wird. In verteilten Formulierungen repräsentieren hierbei oft die Agenten die Busse des Systems. Es wird zwischen AC/Wechselstrom und DC/Gleichstrom Leistungsflussproblemen unterschieden. Das AC-Flussproblem ermöglicht eine genauere Modellierung der

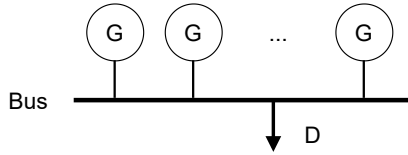


Abbildung 3.2: Einbusmodellierung für das Einsatzplanproblem aus (3.17) nach [73] mit Generatoren G und Last D .

Übertragungsleitung unter Einbeziehung der Blindleistung und physikalischen Randbedingungen der Leitungen. Die resultierenden Nebenbedingungen des Optimierungsproblems sind dabei nicht-linear und je nach Modellierung ebenfalls nicht konvex. Das DC-Modell stellt eine linearisierte Form der AC-Modellierung dar, bei der nur Wirkleistung und Spannungswinkel der Generatoren als Variablen betrachtet werden [77].

Weitere Erweiterungen des kooperativen Einsatzplanproblems umfassen beispielsweise das Berücksichtigen von Verlusten bei den Generatoren [68], Rampenbeschränkungen der Generatoren [69] und die Berücksichtigung von Windanlagen [78], wobei in der Kostenfunktion explizite Kosten für Über- und Unterproduktion bezüglich des Produktionserwartungswertes aufgeführt werden. Zusätzlich existieren kooperative Leistungsflussprobleme, in denen ebenfalls Speicher berücksichtigt werden und deren Formulierung mehrere Zeitslots umfasst [79].

In nicht-kooperativen, spieltheoretischen Modellierungen werden Einsatzplanprobleme oft als Energie- oder Leistungsmanagementprobleme bezeichnet. In diesen Formulierungen werden Zusammenschlüsse von dezentralen Generatoren, Speichern und Verbrauchern betrachtet, die gegeneinander bezüglich einer gemeinsam genutzten Ressource konkurrieren, wodurch eine nicht-kooperative Marktsituation erzeugt wird [16][17]. Diese Zusammenschlüsse repräsentieren die Spieler des nicht-kooperativen Spiels bzw. die Agenten des Systems und können auf verschiedenen Ebenen stattfinden. Beispielsweise können solche Zusammenschlüsse einzelne Häuser, Bezirke, Städte oder auch Regionen sein. In jedem Fall ist das Ziel, den Bedarf der Verbraucher innerhalb des eigenen Zusammenschlusses zu decken, während ein optimaler Einsatz der Generatoren, Speicher und teilweise auch der aktiven Verbraucher gefunden werden soll. Werden Speicher berücksichtigt, wird üblicherweise ein längerer Zeithorizont mit mehreren Zeitslots betrachtet. In [16] wird beispielsweise ein solches Modell vorge-

stellt, bei dem aktive Nutzer, die über Generatoren und Speicher verfügen und einen bestimmten Verbrauch haben, gegeneinander bezüglich des Preises eines zentralen Kraftwerkes konkurrieren. Die Agenten sind hier die aktiven Nutzer, auch Prosumer genannt, deren Modellierung auf Häuserebene stattfindet. In [17] wird ein ähnlicher Ansatz verwendet: Auch hier konkurrieren Zusammenschlüsse von Generatoren, Verbrauchern und Speichern bezüglich der gekoppelten Versorgung über ein Hauptnetz, während ein optimaler Einsatzplan der eigenen Einheiten gefunden werden soll. Zusätzlich können hier die Zusammenschlüsse auch Energie untereinander austauschen, was zu gekoppelten Nebenbedingungen für das nicht-kooperative, spieltheoretische Problem führt. Entsprechend wird in diesem Problem nach einem Verallgemeinerten Nash-Gleichgewicht gesucht, siehe Abschnitt 2.4 des Grundlagenkapitels für die Begriffserklärung.

3.4.2 Gemeinsamkeiten zwischen Problemformulierung und der Literatur

Die grundlegende Problematik ist in der Modellierung der vorliegenden Arbeit wie auch in den beschriebenen Modellen aus der Literatur die gleiche: Ziel ist es, den Bedarf des betrachteten Bilanzgebietes zu decken und die hierbei anfallenden Kosten zu minimieren. Der betrachtete Zeitraum kann dabei in mehrere Zeitslots aufgeteilt sein, vor allem wenn Abhängigkeiten zwischen diesen Zeitslots existieren, siehe beispielsweise die Modellierung der Speicher in Unterabschnitt 3.2.3, oder nur ein einziger Zeitslot betrachtet werden. Die Elemente des Microgrids in Form von Generatoren, Speichern und passiven Verbrauchern folgen den gebräuchlichen Modellierungen der Problemklasse, siehe beispielsweise [16][17][68]. Ebenso folgt die Modellierung der Verbindung zum Hauptnetz bzw. der Preisfunktion für den Bezug von Leistung vom Hauptnetz der spieltheoretischen Modellierung aus [17].

3.4.3 Unterschiede zwischen Problemformulierung und der Literatur

Ein Überblick und eine Einordnung der Unterschiede des in der vorliegenden Arbeit verwendeten Modells findet sich in den Tabellen 3.1 a) - c).

In der Tabelle 3.1 a) wird der Unterschied bezüglich des kooperativen bzw. nicht-kooperativen Verhaltens der Agenten beleuchtet. In klassischen

Einsatzplanproblemen, siehe Gleichung (3.17), und in Leistungsflussproblemen nach [72][76][80] wird eine Kostenfunktion in Form einer Summe der lokalen Kosten der Agenten optimiert. Dagegen wird bei spieltheoretischen Modellierungen des Einsatzplan-/Energiemanagementproblems angenommen, dass die Agenten nicht miteinander kooperieren. Das erhöht die Komplexität des Problems, da nun nicht mehr ein einfaches Optimum die Lösung des Problems darstellt, sondern nach einer Lösung des Spiels, beispielsweise in Form eines Nash-Gleichgewichts, gesucht werden muss. Hierfür müssen komplexere Analysen durchgeführt werden, siehe hierzu z. B. den Zusammenhang zwischen Spielmapping und Existenz eines Nash-Gleichgewichts in Abschnitt 2.4 des Grundlagenkapitels. Die Modellierung in der vorliegenden Arbeit nach Gleichung (3.16) berücksichtigt sowohl die fehlende Kooperation der Microgrids untereinander, als auch das kooperative Verhalten der Agenten innerhalb des Microgrids. Beide Problemstellungen sollen gleichzeitig gelöst werden, was die Komplexität auf ein höheres Niveau anhebt.

In Tabelle 3.1 b) ist eine Übersicht über die Verkopplungen der Bestandteile der Modelle gegeben. Verkopplung der Kostenfunktionen bedeutet in diesem Kontext, dass die lokale Kostenfunktion eines Agenten von einer Variable abhängig ist, die in der Kostenfunktion von mindestens einem weiteren Agenten auftaucht und somit von diesem Agenten auch im Verlauf des Optimierungsverfahrens angepasst werden kann. Gleiches gilt für die Verkopplung von Nebenbedingungen. Besteht eine Verkopplung der Kostenfunktionen der Agenten, müssen sich diese untereinander koordinieren, um eine optimale Entscheidung zu treffen. Die Kostenfunktion in klassischen Einsatzplanproblemen, siehe Gleichung (3.17), und in optimalen Leistungsflussproblemen, siehe [72], sind nicht verkopplert, wodurch einfachere Optimierungstechniken zum Einsatz kommen können als es für die verkopplerte Problemstellung der Fall ist. In dem Modell der vorliegenden Arbeit aus Gleichung (3.16) wird eine Verkopplung der Kostenfunktionen des kooperativen Problems innerhalb der Microgrids durch die Kostenfunktionen f^L , siehe Gleichung (3.9), bezüglich der Übertragungsleitungen erzeugt. Nicht-kooperative Spiele besitzen per Definition verkopplerte Kostenfunktionen, da ansonsten jeder Spieler seine eigenen Kosten unabhängig von den restlichen Spielern minimieren könnte. Sind in nicht-kooperativen Spielen zusätzlich die Nebenbedingungen verkopplert, muss nach einem Verallgemeinerten Nash-Gleichgewicht, siehe Grundlagenabschnitt 2.4, gesucht werden. Die Nebenbedingung der Modellierung der vorliegenden Arbeit sind nur bezüglich des kooperativen Problems innerhalb der Microgrids und nicht bezüglich des nicht-

Tabelle 3.1: Vergleich der Modelle. EPP: klassisches Einsatzplanproblem (nach (3.17)), EMP: nicht-kooperatives Energiemanagementproblem (siehe z. B. [16]), **MM-EPP**: Multimicrogrideinsatzplanproblem (Modellierung für diese Arbeit nach (3.16)), LFP: Leistungsflussproblem (siehe z. B. [72]). Zunahme des Detailgrades bzw. der Komplexität von links nach rechts.

a) *Vergleich Komplexität der Kooperation.*

Komplexität	vollst. Kooperation	keine Kooperation	2-Level (Koop. + keine Koop.)
Modell	EPP [74], LFP [72][80]	EMP [16][17]	MM-EPP

b) *Verkopplung Kosten und Spielnebenbedingungen.*

	nicht-verkoppelte Kosten	verkoppelte Kosten	verkoppelte Nbd. Spiel
Modell	EPP[74], LFP [72][80]	MM-EPP , EMP [16]	EMP [17]

c) *Vergleich Detailgrad Übertragungsleitungen.*

Detailgrad	1-Bus	Mehr-Bus/ Kapazität	Mehr-Bus/ lin. Modell	Mehr-Bus/ phys. Modell
Modell	EPP [72]	MM-EPP EMP [16][17]	DC-LFP [72]	AC-LFP [80]

kooperativen Problems zwischen den Microgrids verkoppelt, sodass ein einfaches Nash-Gleichgewicht als Lösungskonzept verwendet werden kann.

Ein weiterer wichtiger Unterschied zwischen dem Modell in Gleichung (3.16) und der Literatur besteht bezüglich der Modellierung der Übertragungsleitungen, siehe Tabelle 3.1 c). In klassischen Einsatzplanproblemen, sowohl im nicht-kooperativen als auch im kooperativen Fall, wird von einem 1-Bussystem ausgegangen und auf eine Modellierung des Übertragungsnetzes verzichtet. Im Modell der vorliegenden Arbeit wird von einem Mehrbussystem ausgegangen, bei dem explizit die Lokalität, d. h. ob eine Leitung zwischen zwei Bussen überhaupt existiert, und die Kapazität der Übertragungsleitungen berücksichtigt werden. Diese Modellierung entspricht einer vereinfachten Version des linearen (DC) Leistungsflussmodells, bei dem der Leistungsfluss über eine Übertragungsleitung zwischen

Bus i und j durch $\frac{\delta_i - \delta_j}{x_{ij}}$ festgelegt wird, wobei δ_i, δ_j die Spannungswinkel der Busse und x_{ij} der Blindwiderstand der Leitung ist [72]. In dem Modell der vorliegenden Arbeit werden die Spannungswinkel nicht explizit berücksichtigt, sondern die übertragene Leistung $e_{ij} = \frac{\delta_i - \delta_j}{x_{ij}}$ als Optimierungsvariable in das Problem mitaufgenommen. Diese Vereinfachung ist aufgrund der Anforderungen an das Modell, die durch die oben beschriebene höhere Komplexität der Kooperation des Multimicrogridmodells entstehen, notwendig. Die nächste Detailstufe bezüglich der Leitungen ist schließlich das nichtlineare bzw. nicht konvexe AC Modell, dem ein physikalisches Modell der Leitungen zugrunde liegt [80].

3.5 Herausforderungen bezüglich der verteilten Lösung

Durch die obig beschriebenen Erweiterungen des in der vorliegenden Arbeit verwendeten Modells zur bestehenden Literatur entstehen verschiedene Herausforderungen bezüglich der Lösung des Einsatzplanproblems. Diese werden im Folgenden beschrieben und dienen der Motivation der Algorithmen, die in den nachfolgenden Hauptkapiteln der vorliegenden Arbeit beschrieben werden.

Kooperatives Problem: Kopplung der Kostenfunktionen und lokale Nebenbedingungen Bezüglich des kooperativen Problems innerhalb der Microgrids stellt zum einen die Kopplung der lokalen Kostenfunktionen der Agenten in der gemeinsamen Variable x^h in Folge der Berücksichtigung der Übertragungsleitungen eine Herausforderung dar, die beispielsweise in der klassischen Formulierung des Einsatzplanproblems in Gleichung (3.17) nicht existiert. Damit können bekannte Lösungsstrategien, wie die Entkopplung des Optimierungsproblems über Lagrangeparameter, siehe beispielsweise [68][81], nicht mehr angewendet werden.

Darüber hinaus besitzt jeder Agent neben globalen Nebenbedingungen bezüglich der Beschränkung der Übertragungsleitungen auch eine Reihe von lokalen Nebenbedingungen, die im verteilten Fall privat bleiben sollen. Viele bekannte Algorithmen können zwar die oben genannte Verkopplung der Kostenfunktionen verarbeiten, allerdings nicht gleichzeitig die Privatsphäre der lokalen Nebenbedingungen wahren, siehe zum Beispiel [82][83][84].

Die Algorithmen der Beiträge 1 und 2 der vorliegenden Arbeit können

sowohl die Verkopplung der Kostenfunktionen als auch die privaten Nebenbedingungen berücksichtigen. Für eine ausführliche Diskussion, auch in Bezug auf bestehende Literatur, sei auf das Kapitel 4 verwiesen.

Vollständig verteilte Lösung von verkoppelten kooperativen und nicht-kooperativen Optimierungsproblemen In der beschriebenen Modellierung treten innerhalb der Microgrids kooperative, verteilte Optimierungsprobleme auf, die auf der oberen Ebene zwischen den Microgrids nicht-kooperativ von den Aktionen anderer Microgrids abhängig sind. Um sowohl das kooperative, als auch das nicht-kooperative Problem optimal zu lösen, müssen beide Probleme gleichzeitig betrachtet werden. Das resultierende, kombinierte Problem wird als Multiclusterspiel bezeichnet.

Während, wie oben beschrieben, bereits Lösungsalgorithmen existieren, die entweder das kooperative oder das nicht-kooperative Problem lösen können, ist der Stand der Forschung bezüglich Multiclusterspielen noch nicht stark ausgeprägt. Für unbeschränkte Problemstellungen gibt es bereits einige Lösungsalgorithmen, siehe beispielsweise [25][26][27]. Beschränkte Probleme allerdings, wie auch die Problemstellung (3.16), sind bisher weniger stark behandelt.

Im Beitrag 5 der vorliegenden Arbeit wird daher ein Algorithmus vorgestellt, mit dem ein beschränktes Multiclusterspiel, und damit auch die Problemstellung (3.16), gelöst werden kann. Für die Vorstellung des Algorithmus und für eine detaillierte Diskussion des Standes der Forschung sei auf das Kapitel 5 verwiesen.

4 Verteilte Lösung von kooperativen Optimierungsproblemen

4.1 Einleitung

Durch verteilte Optimierung werden Optimierungsprobleme gelöst, die auf ein System, bestehend aus mehreren Agenten, verteilt sind. Jeder Agent des Systems besitzt eigene, *lokale* Kostenfunktionen und Nebenbedingungen, die ausschließlich diesem Agenten bekannt sind. Darüber hinaus können auch *globale* Nebenbedingungen existieren, von denen alle Agenten des Systems Kenntnis haben.

Bei *kooperativer*, verteilter Optimierung wird angenommen, dass die Agenten zusammenarbeiten, um die Gesamtkostenfunktion des Systems in Form der Summe der lokalen Kostenfunktionen zu minimieren. Dieses Ziel wird auch dann verfolgt, wenn für einzelne Agenten die egoistische Optimierung der lokalen Kostenfunktion günstiger wäre. Es wird somit die Wohlfahrt des Multiagentensystems maximiert.

In diesem Kapitel werden kooperative, verteilte Problemstellungen betrachtet, bei denen die Kostenfunktionen und die Nebenbedingungen der Agenten untereinander verkoppelt sind. Damit müssen sich die Agenten sowohl bezüglich ihrer Kostenfunktionen, als auch bezüglich der Nebenbedingungen auf eine gemeinsame Aktion einigen¹¹. Alle betrachteten Problemstellungen sind konvex, was bedeutet, dass alle lokalen Kostenfunktionen und alle lokalen sowie globalen Nebenbedingungsmengen konvex sind.

Um eine koordinierte Aktion zur Minimierung der Gesamtkostenfunktion zu wählen, kommunizieren die Agenten über einen Kommunikationsgraphen \mathcal{G} , siehe das Beispielsystem in Abbildung 4.1. Die Algorithmen,

¹¹Es existieren verteilte Problemstellungen, bei denen ausschließlich die Nebenbedingungsmengen miteinander verkoppelt sind, siehe hierzu den Spezialfall in Gleichung (4.3).

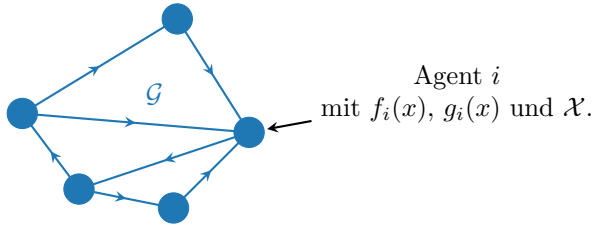


Abbildung 4.1: Ein Multiagentensystem, bestehend aus $N = 5$ Agenten, die durch den gerichteten Kommunikationsgraphen \mathcal{G} miteinander verbunden sind. Nur der Agent i kennt die lokale Kostenfunktion $f_i(x)$ und die Nebenbedingungen $g_i(x) = \text{col}(g_{i1}(x), \dots, g_{iM_i}(x)) \leq 0$, während die Beschränkungsmenge \mathcal{X} allen Agenten des Systems bekannt ist.

die in diesem Kapitel den Stand der Forschung erweitern, basieren auf Konsensusalgorithmen zur Koordination unter den Agenten und verwenden Gradienteninformationen bezüglich der lokalen Kostenfunktionen, um das Optimum zu erreichen.

4.2 Problemstellung

Sei ein Multiagentensystem gegeben, das aus N Agenten besteht, die sich in der Indexmenge $\mathcal{I} = \{1, \dots, N\}$ befinden. Ein solches System ist beispielhaft in der Abbildung 4.1 dargestellt. Jeder Agent i besitzt eine lokale Kostenfunktion $f_i : \mathbb{R}^d \rightarrow \mathbb{R}, \forall i \in \mathcal{I}$. Gekoppelt sind diese Kostenfunktionen durch die gemeinsame Aktion $x \in \mathbb{R}^d$, wobei sich die Dimension d entsprechend der Problemstellung ergibt. Zusätzlich existieren für den Agenten i genau M_i lokale Nebenbedingungen in Form von Ungleichungen $g_{ij}(x) \leq 0$ mit $g_{ij} : \mathbb{R}^d \rightarrow \mathbb{R}, \forall i \in \mathcal{I}, j \in \mathcal{M} = \{1, \dots, M_i\}$, und einer globalen Nebenbedingungsmenge $\mathcal{X} \subseteq \mathbb{R}^d$. Die Agenten kooperieren miteinander, um das Gesamtoptimierungsproblem, d. h. die Summe der lokalen Kostenfunktionen, unter den gegebenen Nebenbedingungen zu minimieren. Es ergibt sich damit das Optimierungsproblem

Problem 3 (Verteiltes Optimierungsproblem)

$$\min_x F(x) = \min_x \sum_{i=1}^N f_i(x), \quad (4.1a)$$

$$s.t. \ g_i(x) \leq 0, \ g_i(x) = \text{col}(g_{i1}(x), \dots, g_{ij}(x), \dots, g_{iM_i}(x)), \quad (4.1b)$$

$$x \in \mathcal{X}. \quad (4.1c)$$

Die Ungleichungen $g_{ij}(x) \leq 0$, vektorisiert in der Abbildung $g_i(x)$, sind *lokale* Nebenbedingungen des Problems. Das bedeutet, dass diese Funktionen, wie auch die Kostenfunktion f_i , nur dem jeweiligen Agenten i bekannt sind und auch während des Optimierungsprozesses privat bleiben sollen. Die allgemeine Beschränkungsmenge \mathcal{X} ist eine *globale* Nebenbedingung des Problems und somit allen Agenten im System bekannt.

Sowohl die lokalen Kostenfunktionen $f_i(x)$ als auch die lokalen Abbildungen $g_i(x)$ sind verkoppelt bezüglich der Variablen x : Eine Änderung dieser Variablen durch Agent i hat einen Einfluss auf Kosten und die Einhaltung der Nebenbedingungen von Agent $j \in \mathcal{I} \setminus i$. Wäre dies nicht der Fall, d. h. wären sowohl Kosten als auch Nebenbedingungen nicht verkoppelt, könnte jeder Agent lokal sein entkoppeltes Optimierungsproblem lösen.

Aufgrund der existierenden Verkopplung müssen die Agenten sich bezüglich der Änderungen der Optimierungsvariablen x untereinander koordinieren. Dazu verwaltet jeder Agent $i \in \mathcal{I}$ eine Schätzung $x_i \in \mathbb{R}^d$ bezüglich der Optimierungsvariablen $x \in \mathbb{R}^d$, die nur von dem jeweiligen Agenten i verändert werden kann. Damit das ursprüngliche Problem 3 als gelöst gilt, müssen sich die Schätzungen über den Optimierungsverlauf aneinander annähern, sodass bei Konvergenz eines entsprechenden Verfahrens sich ein Konsensus entstellt, der dem Optimum $x^* \in \mathbb{R}^d$ des Problems entspricht. Das Problem 3 kann nun so erweitert werden, dass die Schätzungen und der zu erreichende Konsensus berücksichtigt werden:

Problem 4 (Verteiltes Optimierungsproblem mit Schätzung)

$$\min_{(x_i)_{i \in \mathcal{I}}} \sum_{i=1}^N f_i(x_i), \quad (4.2a)$$

$$s.t. \ g_i(x_i) \leq 0, \ g_i(x_i) = \text{col}(g_{i1}(x_i), \dots, g_{ij}(x_i), \dots, g_{iM_i}(x_i)), \quad (4.2b)$$

$$x_i \in \mathcal{X}, \quad (4.2c)$$

$$x_i = x_j, \ \forall i, j \in \mathcal{I}. \quad (4.2d)$$

Die Gleichungsbedingung in (4.2d) wird auch als Konsensusbedingung bezeichnet. Ist diese erfüllt, kann leicht nachvollzogen werden, dass dann das Optimum x^* von Problem 4 ebenso das Problem 3 löst.

Um einen für die Konsensusbildung unerlässlichen Informationsaustausch zu ermöglichen, sind die Agenten untereinander durch einen Kommunikationsgraphen $\mathcal{G} \{\mathcal{V} \equiv \mathcal{I}, \mathcal{E}\}$ verbunden, siehe Abbildung 4.1. Unter Verwendung eines Konsensusprotokolls, siehe Grundlagenabschnitt 2.6, und damit einer entsprechend gewichteten Kommunikationsmatrix, können die Schätzungen iterativ aneinander angenähert werden und damit die Konsensusbedingung (4.2d) asymptotisch erfüllt werden.

Die in diesem Kapitel vorgestellten Algorithmen zur Lösung von verteilten Optimierungsproblemen bestehen aus zwei miteinander verzahnten Dynamiken: Der *Konsensusdynamik*, welche die Schätzungen der Agenten über den Iterationsverlauf einander angleicht und die *Optimierungsdynamik*, welche die Schätzung so beeinflusst, dass diese sich in Richtung des Optimums der Problemstellung bewegen. Die Konsensusdynamik wird dabei maßgeblich von der Kommunikationsarchitektur und damit der dieser zugrundeliegenden gewichteten Kommunikationsmatrizen beeinflusst, während die Optimierungsdynamik sich durch die Wahl der Optimierungsmethode bestimmt. Räumlich betrachtet beschreibt folglich die Konsensusdynamik den Optimierungsvorgang *zwischen* den Agenten, während die Optimierungsdynamik den Vorgang *innerhalb* der Agenten repräsentiert.

Um die behandelte Problemklasse dieses Kapitels genauer festzulegen, werden die nachfolgenden Pauschalannahmen getroffen, die für alle in diesem Kapitel behandelten Problemstellungen gelten.

Pauschalannahme 4.1 (Konvexität) *Das Problem 3 ist konvex, d. h. die lokalen Kostenfunktionen $f_i(x)$, $\forall i \in \mathcal{I}$, und die Nebenbedingungsfunktionen $g_i(x_i)$, $\forall i \in \mathcal{I}$, sind konvex.*

nen $g_{ij}(x)$, $\forall i \in \mathcal{I}, j \in \{1, \dots, M_i\}$, sind konvex auf \mathbb{R}^d . Die Beschränkungsmenge $\mathcal{X} \subset \mathbb{R}^d$ ist konvex und kompakt.

Pauschalannahme 4.2 (Differenzierbarkeit) Die lokalen Kostenfunktionen $f_i(x)$, $\forall i \in \mathcal{I}$, und lokalen Nebenbedingungsfunktionen $g_{ij}(x)$, $\forall i \in \mathcal{I}, j \in \{1, \dots, M_i\}$, sind stetig differenzierbar auf \mathbb{R}^d .

Diese Pauschalannahmen ermöglichen die Verwendung von gradientenbasierten Optimierungsfahren.

Spezialfall: Entkoppelte Kostenfunktionen Wie in Problem 3 erkannt werden kann, sind sowohl die lokalen Kostenfunktionen als auch die Nebenbedingungen durch die gemeinsame Verwendung der Variable x verkoppelt. Es existieren allerdings auch Problemstellungen, bei denen die lokalen Kostenfunktionen separierbar und ausschließlich die Nebenbedingungen verkoppelt sind, wie zum Beispiel

$$\min_x \sum_{i=1}^N f_i(x_i), \quad (4.3a)$$

$$\text{s.t. } g_i(x) \leq 0, \forall i \in \mathcal{I} \quad (4.3b)$$

mit $x = \text{col}((x_i)_{i \in \mathcal{I}})$. Durch die sogenannte *Duale Entkopplung* (englisch: Dual Decomposition) kann die Problemstellung in eigenständige Subprobleme zerlegt werden, die ausschließlich über Aktualisierung eines Lagrangemultiplikators verkoppelt sind. Zur effizienten Lösung dieser Problemklasse kann die *Alternating Direction of Multiplier* Methode, kurz ADMM, bzw. Augmented-Lagrangian-Ansätze verwendet werden. Verteilte Ansätze finden sich beispielsweise in [85][86][87] und [88].

Da sich die separierbare Problemstellung von der in Problem 3 definierten Problematik unterscheidet, wird im Folgenden nicht vertieft auf diese Problemklasse eingegangen.

4.3 Stand der Forschung und Beiträge

Die verteilte Optimierung steht seit mehreren Jahren im aktiven Fokus der Forschung. Die Initialzündung entstand durch die Erforschung und breite Veröffentlichung von Konsensusprotokollen, die eine effiziente Konsensusfindung bezüglich relevanter Variablen in einem Agentennetzwerk ermöglichen. Eine Auswahl dieser Protokolle sowie deren Funktionsweise

findet sich im Grundlagenabschnitt 2.6. In den letzten zehn Jahren wurden viele konsensusbasierte Optimierungsalgorithmen erforscht, die den Forschungsstand in einzelnen Nischen erweitern. Im Fokus der vorliegenden Arbeit stehen gradientenbasierte Verfahren und deren Erweiterungen, weswegen der Brennpunkt der nachfolgenden Literaturdiskussion auf diesem Teilgebiet der verteilten Optimierung liegt.

Um einen Überblick über die zahlreichen Veröffentlichungen zu schaffen, ist in Abbildung 4.2 eine Einordnung der Veröffentlichungen bezüglich dieser Algorithmuskategorie gegeben. Dabei werden zunächst gradientenbasierte Verfahren von ADMM bzw. Augmented Lagrangian und proximalen Methoden separiert. Bezüglich gradientenbasierter Verfahren wird anschließend unterschieden, ob mit den Verfahren auch beschränkte oder nur unbeschränkte Probleme gelöst werden können. In der nächsten Ebene wird anhand der anwendbaren Schrittweite diskriminiert. Die Bedeutung von konstanten bzw. zeitabhängigen Schrittweiten wird in den nachfolgenden Abschnitten diskutiert. Die Verfahren zur Lösung von beschränkten Optimierungsproblemen werden anschließend anhand der verwendeten Einbindungsmethode der Nebenbedingungen unterschieden. Schließlich ist bei allen angegebenen Quellen bezüglich der gradientenbasierten Lösung von verteilten Optimierungsproblemen durch die farbliche Codierung die notwendige Gewichtung der Kommunikationsgraphen indiziert.

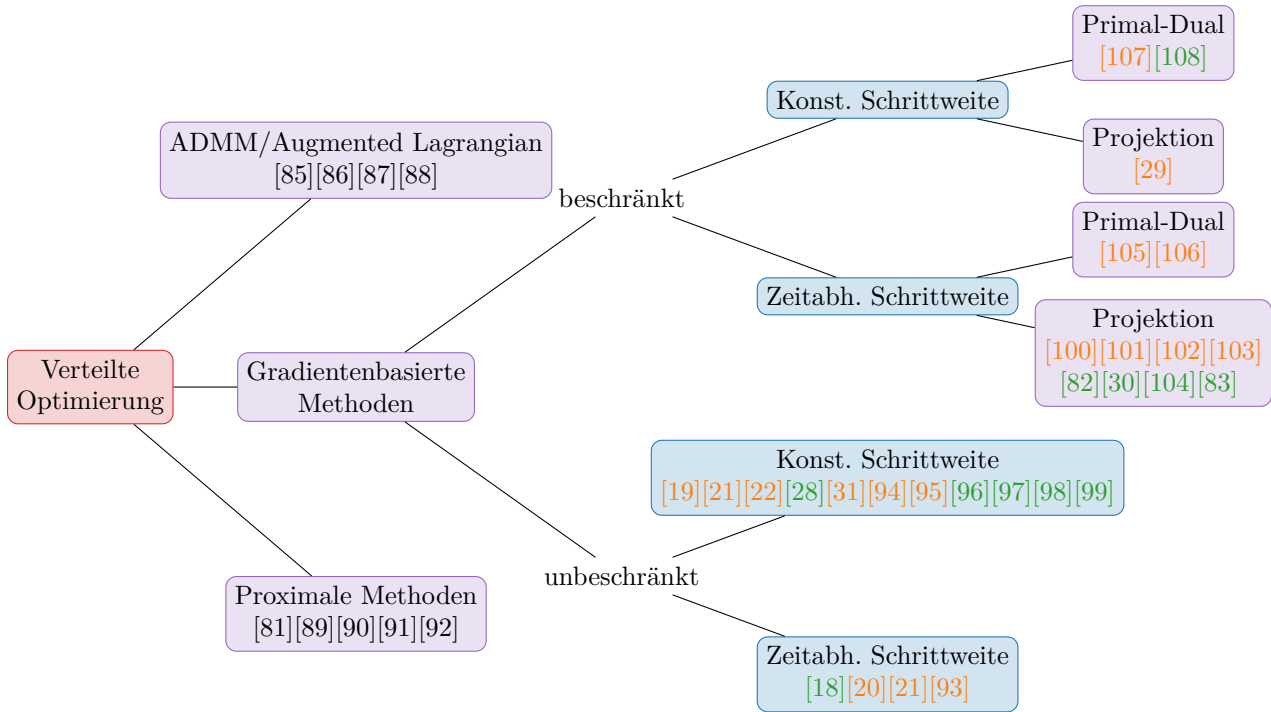


Abbildung 4.2: Klassifikation der Literatur zu kooperativer, verteilter Optimierung. Farbkodierung: **Wurzelknoten**, **Methoden**, **Schrittweitereigenschaft**, **doppelt-stochastische Gewichtung**, **reihen-/spalten-stochastische Gewichtung**.

4.3.1 Stand der Forschung: Gradientenbasierte Verfahren für verteilte Optimierung

Viele der veröffentlichten Verfahren basieren im wesentlichen auf zwei Algorithmen, die für unbeschränkte, verteilte Optimierungsprobleme entwickelt sind. Im Folgenden werden diese beiden Algorithmen als *Basisalgorithmen* vorgestellt und entsprechende Erweiterungen diesen zugeordnet. Unterschieden werden veröffentlichte Verfahren darüber hinaus bezüglich

- der Lösung von *beschränkten* gegenüber *unbeschränkten* Problemstellungen,
- *doppelt-* gegenüber *reihen-/spalten-stochastisch gewichteter* Kommunikation und
- *zeitabhängiger* gegenüber *konstanter* Schrittweite.

Weitere Eigenschaften und Annahmen bezüglich einer engeren Auswahl der Algorithmen sind in Tabelle 4.1 in Abschnitt 4.3.2 zusammengefasst.

Ausführliche Literaturübersichten, die über die für diese Arbeit relevanten Aspekte hinausgehen, finden sich in [13] und [14].

Zusätzlich gibt [109] eine Übersicht über verteilte Algorithmen zur Lösung von unbeschränkten Problemen und die Veröffentlichung [110] einen Überblick über Algorithmen, bei denen die verteilte Lösung von Problemen des Maschinellen Lernens im Mittelpunkt steht.

Die nachfolgend verwendete Notation zur Beschreibung der Algorithmen bezieht sich auf die Problemstellung 3 bzw. 4. Entsprechend bezeichnet $x \in \mathbb{R}^d$ die zu optimierende Variable der Problemstellung, $x_i \in \mathbb{R}^d$ die Schätzung des Agenten i bezüglich der Optimierungsvariablen x und $f_i(x)$ bzw. $f_i(x_i)$ die lokale Kostenfunktion des Agenten i .

Verfahren für unbeschränkte Probleme

Eine der ersten Veröffentlichungen zu unbeschränkter, verteilter Optimierung im abgesteckten Rahmen der vorliegenden Arbeit findet sich in [93] und ist der erste Basisalgorithmus, der durch diverse Veröffentlichungen erweitert wird. Die Aktualisierungsvorschriften, die jeder Agent i in der

Iteration k durchführt, lauten

$$\hat{x}_i(k) = \sum_{j=1}^N A_{ij}(k)x_j(k) \quad (4.4a)$$

$$x_i(k+1) = \hat{x}_i(k) - \alpha_{i,k}d_i(k), \quad (4.4b)$$

wobei $A(k) \in \mathbb{R}^{N \times N}$ eine zeitvariante, doppelt-stochastische Kommunikationsmatrix, $d_i(k) \in \mathbb{R}^d$ den jeweiligen (Sub-)Gradienten der lokalen Kostenfunktion f_i des Agenten i am Punkt $\hat{x}_i(k)$ und $\alpha_{i,k} \in \mathbb{R}^+$ die zeitabhängige, agentenspezifische Schrittweitenfolge beschreiben. Nach diesem Algorithmus teilt zunächst jeder Agent seine aktuelle Schätzung $x_i(k)$ bezüglich der Optimierungsvariablen seinen direkten Nachbarn über die Kommunikationsmatrix $A(k)$ mit und aktualisiert seine eigene Schätzung $\hat{x}_i(k)$ als gewichtete Summe seiner eigenen und der empfangenen Schätzungen. Dies ist der Konsensusschritt. Anschließend folgt der Gradientenschritt, bei dem zunächst jeder Agent lokal seinen (Sub-)Gradienten $d_i(k)$ berechnet und multipliziert mit dem aktuellen Wert der Schrittweitenfolgen $\alpha_{i,k}$ von der aktualisierten Schätzung abzieht.

Diese Struktur, gekennzeichnet durch die direkte Aktualisierung der eigenen Schätzung durch den lokalen (Sub-)Gradienten, ist das Gerüst für viele weitere Algorithmen und wird im Folgenden aufgrund der Ähnlichkeit von Gleichung (4.4b) zum zentralen Gradientenabstieg, siehe Gleichung (2.13) im Grundlagenkapitel, als *verteilter Gradientenabstieg* bezeichnet.

Wird die Schrittweite $\alpha_{i,k} = \alpha, \forall i \in \mathcal{I}$, konstant gewählt, konvergieren die Schätzungen x_i unter Vorschrift (4.4) nur in eine $\mathcal{O}(\alpha)$ -Umgebung des Optimums und zusätzlich wird je nach Problemstellung die Konsensusbedingung (4.2d) nicht erreicht [19][93][111]. Für eine ausführliche Argumentation, warum dies der Fall ist, siehe beispielsweise [19]. Um eine exakte Konvergenz des Verfahrens zu erreichen, muss die Schrittweitenfolge als eine absinkende Folge gewählt werden. Allerdings ist die theoretische und praktische Konvergenz bei abfallenden Schrittweitenfolgen langsamer als bei konstanter Wahl der Schrittweite [19].

In [18] wird die Aktualisierungsstruktur aus (4.4) so erweitert, dass eine Konvergenz auch auf zeitvarianten, spalten-stochastisch gewichteten Graphen erreicht wird. Dazu wird die auf einer doppelt-stochastischen Matrix beruhende Konsensusaktualisierung aus Gleichung (4.4a) durch das Push-Sum-Konsensusprotokoll, siehe Unterabschnitt 2.6.1 aus dem

Grundlagenkapitel, ersetzt und die Aktualisierungsstruktur des Verfahrens an das neue Konsensusprotokoll angepasst.

Aufgrund der im Vergleich langsameren Konvergenz von Algorithmen mit abfallenden Schrittweitenfolgen der Struktur aus (4.4) sind Verfahren, bei denen konstante Schrittweiten verwendet werden können, mit der Zeit immer mehr in den Fokus gerückt. Als eine besonders effiziente Methode hat sich das sogenannte *Gradient-Tracking* etabliert. Dieses Verfahren basiert auf einer Methodik aus [112], mit dem ein Konsensus bezüglich eines zeitvarianten Signals in einem Agentensystem erreicht werden kann. Klassische Konsensusprotokolle, wie beispielsweise das Push-Sum-Protokoll aus Unterabschnitt 2.6.1, sind dagegen für die Schätzung von statischen (Anfangs-)Werten entwickelt. In dem unbeschränkten Optimierungsverfahren aus [19] sowie in den später entwickelten Methoden aus [31] und [95] wird der Ansatz aus [112] verwendet, um den Gradienten des Gesamtproblems $\sum_{i=1}^N \nabla f_i(x)$ zu schätzen. Ein entsprechender Optimierungsalgorithmus kann, beispielsweise nach [31], unter Verwendung von zeitvarianten, doppelt-stochastischen Kommunikationsmatrizen $A(k) \in \mathbb{R}^{N \times N}$ wie folgt formuliert werden:

$$\hat{x}_i(k) = \sum_{j=1}^N A_{ij}(k)x_j(k), \quad (4.5a)$$

$$x_i(k+1) = \hat{x}_i(k) - \alpha y_i(k), \quad (4.5b)$$

$$y_i(k+1) = \sum_{j=1}^N A_{ij}(k)y_j(k) + \nabla f_i(x_i(k+1)) - \nabla f_i(x_i(k)), \quad (4.5c)$$

Im Vergleich zum *verteilten Gradientenabstieg* aus (4.4) ändert sich hierbei, neben der zusätzlichen Iteration zur Gradientenschätzung, nur die Richtung des Gradientenabstieges. Während in Gleichung (4.4b) der (Sub-)Gradient der lokalen Kostenfunktion die Richtung des Schrittes bestimmt, wird in Gleichung (4.5b) ein Schritt in Richtung der Schätzung des Gradienten der globalen Kostenfunktion, gespeichert in $y_i(k) \in \mathbb{R}^d$, durchgeführt. Diese Schätzung erfolgt in Gleichung (4.5c) durch einen Konsensuschritt für den Informationsaustausch der Agenten untereinander bezüglich der Gradientenschätzung und Auswertung der lokalen Gradienten des jeweiligen Agenten zu den Zeitpunkten $k+1$ und k . Der Gradient $\sum_{i=1}^N \nabla f_i(x_i(k))$ entspricht somit dem zu schätzenden, zeitlich veränderlichen Zustand aus der Quelle [112].

In [31] wird unter Ersetzung des Konsensusprotokolls in Gleichung (4.5a) durch das Push-Sum-Konsensusprotokoll, siehe Unterabschnitt 2.6.1 aus dem Grundlagenkapitel, eine Erweiterung auf nicht-doppelt-stochastisch gewichtete Graphen vorgestellt. Hierbei beruht die Konsensusbildung, auch diejenige für die Gradientenschätzung in Gleichung (4.5c), auf spalten-stochastischen Kommunikationsmatrizen. Diese Erweiterung erfordert eine zusätzliche Gewichtung der Gradienten, die über ein weiteres Schätzverfahren ermittelt werden muss.

Auch in [113] wird eine Erweiterung auf allgemeinere Graphgewichtungen vorgestellt. Hierbei werden die doppelt-stochastischen Kommunikationsmatrizen aus (4.5) durch reihenstochastische Matrizen ersetzt, wobei auch hier eine entsprechende Gewichtung der lokalen Gradienten, die in jedem Zeitschritt über ein Schätzungsverfahren ermittelt werden muss, notwendig wird. Eine Erweiterung dieses Algorithmus auf agentenspezifische Schrittweiten erfolgt in [96].

Um die Gradientengewichtung und damit das zusätzliche Schätzverfahren der letzten beiden Erweiterungen zu vermeiden, wird in [28] und [98] die Kommunikationsmatrix für den Konsensus bezüglich der Schätzung x_i in Gleichung (4.5a) reihen-stochastisch gewichtet, während die Kommunikationsmatrix für den Konsensus bezüglich der Gradientenschätzung y_i in Gleichung (4.5c) spalten-stochastisch gewählt wird.

Gleiche Gewichtungen werden auch in der Arbeit [99] verwendet, wobei ein Konsensus bezüglich der Optimierungsvariablen und Gradientenschätzungen durch ein Gossip-Verfahren erreicht wird, bei dem in jeder Iteration k nur jeweils eine Kommunikationsverbindung des Graphen aktiv ist.

Verfahren für beschränkte Probleme

Im Grundlagenabschnitt 2.3 wird eine Auswahl von zentralen Methoden vorgestellt, mit denen Nebenbedingungen in Gradientenabstiegsverfahren integriert werden können. Die nachfolgende Literaturdiskussion gibt einen Überblick über die gradientenbasierte Lösung von beschränkten, verteilten Problemen, bei denen oft die im Grundlagenabschnitt vorgestellten Techniken verwendet werden. Viele dieser Algorithmen bauen dabei auf den unbeschränkten Basisalgorithmen (4.4) und (4.5) auf. Zunächst werden Erweiterungen des *verteilten Gradientenabstiegs* aus

den Gleichungen (4.4) betrachtet.

Die am häufigsten eingesetzte Methode, Nebenbedingungen in die verteilten Optimierung miteinzubeziehen, ist die Projektion des Gradientenschrittes. Wie in Abschnitt 2.3 des Grundlagenkapitels beschrieben, wird hierbei die Gradientenaktualisierung auf eine konvexe Beschränkungsmenge projiziert, d. h. der Punkt in der Beschränkungsmenge gesucht, der den geringsten Abstand zu der Gradientenaktualisierung besitzt.

In [100] wird durch die Projektion des Gradientenschrittes (4.4b) eine globale Beschränkungsmenge berücksichtigt, wobei doppelt-stochastische Kommunikationsmatrizen verwendet werden. Zusätzlich wird erörtert, dass lokale und voneinander verschiedene Beschränkungsmengen nur dann verwendet werden können, wenn die Gewichtung der Kommunikationsmatrix neben der doppelt-stochastischen Eigenschaft auch uniform ist, d. h. $A_{ij} = 1/m$ mit einem entsprechend gewählten m gilt. In [101] wird auf diesen Algorithmus Bezug genommen und gezeigt, dass der projizierte verteilte Gradientenabstieg mit doppelt-stochastischen Kommunikationsmatrizen auch für lokale Beschränkungsmengen konvergiert.

In [103] wird ein ähnlicher Algorithmus vorgestellt, wobei dort auf eine zufällige Untermenge der Beschränkungsmenge projiziert wird. Dies ist beispielsweise notwendig, wenn die Beschränkungsmenge zu Beginn des Optimierungsverfahrens nicht bekannt oder die Projektion auf die gesamte Menge zu aufwendig ist.

In [84] wird ein Push-Sum-Protokoll-basierter Optimierungsalgorithmus mit einer Projektion kombiniert, wobei der Gradientenschritt durch eine proximale Minimierung durchgeführt wird. Mit dem resultierenden Algorithmus ist eine beschränkte, verteilte Optimierung unter spaltenstochastisch gewichteten Kommunikationsmatrizen möglich.

Die Quelle [82] erweitert den Basisalgorithmus (4.4) so, dass dieser unter reihen-stochastisch gewichteter Kommunikation konvergiert. Dazu ist ein zusätzlicher Konsensussschritt in jeder Iteration zur Schätzung des linken Eigenwertes der reihen-stochastischen Kommunikationsmatrix und eine Skalierung des (Sub-)Gradienten mit dieser Schätzung notwendig. Nebenbedingungen werden durch Projektion auf eine globale Beschränkungsmenge berücksichtigt. Die Veröffentlichung [83] erweitert

die Ergebnisse aus [82] für zeitvariante Kommunikationsgraphen.

In [106] wird der Algorithmus aus (4.4) durch einen primal-dualen Ansatz, siehe Grundlagenabschnitt 2.3, zur Berücksichtigung von lokalen Ungleichungsnebenbedingungen $g_i(x) \leq 0$ erweitert. Dafür wird der Gradient bezüglich der primalen Variablen x der lokalen Lagrangefunktion $\mathcal{L}_i(x, \mu) = f_i(x) + \mu_i^T g_i(x)$ zur Aktualisierung verwendet. Zusätzliche Nebenbedingungen in Form von lokalen Beschränkungen können durch Projektion mit eingebunden werden. Der Algorithmus benötigt eine doppelt-stochastisch gewichtete Kommunikationsstruktur, die zeitvariant sein kann.

Der Beitrag in [29] stellt eine Erweiterung des *Gradient-Tracking*-Verfahrens aus (4.5) dar. In dieser Veröffentlichung wird der unbeschränkte Basisalgorithmus (4.5) unter Verwendung einer zusätzlichen Konstante $\beta \in (0, 1)$ so erweitert, dass eine Projektion auf eine globale Beschränkungsmenge möglich wird. Mit dieser Konstante kann der Anteil der Konsensusaktualisierung gegenüber der Gradientenaktualisierung gewichtet werden. Für die konstante Schrittweite α ist eine Einschränkung sowohl durch eine obere Schranke als auch durch eine untere Schranke für den Konvergenzbeweis notwendig, wobei die jeweiligen Schranken von der Wahl von β abhängig sind. Das Verfahren verwendet doppelt-stochastische Kommunikationsmatrizen.

Einen Sonderfall stellt das Verfahren aus [108] dar. Bei diesem wird ein beschränktes, bezüglich der Kostenfunktionen entkoppeltes Problem betrachtet. Dieses Problem wird dann in seine duale Form überführt, die sich wiederum als verkoppeltes Problem mit einer Konsensusnebenbedingung darstellen lässt. Die so transformierte Problemstellung wird dann unter Verwendung einer Variante des *Gradient-Tracking*-Ansatzes (4.5) gelöst.

Verfahren ohne Bezug auf die Basisalgorithmen (4.4) und (4.5)

Um Algorithmen, die für doppelt-stochastisch gewichtete Kommunikation entworfen sind, für die Verwendung von allgemeineren Gewichtungen zu erweitern, wird in den bisher genannten Algorithmen entweder das Push-Sum-Konsensusverfahren oder reihen-stochastische Matrizen verwendet. In beiden Fällen ist eine zusätzliche Gewichtung der Gradienten notwendig, um die Asymmetrie der Kommunikation auszugleichen. In [104] wird ein Algorithmus präsentiert, bei dem diese Gewichtung nicht

erforderlich ist. Dafür wird eine zusätzliche Variable eingeführt, die in jeder Iteration aktualisiert wird und welche die Asymmetrie ausgleicht. Die Notwendigkeit einer expliziten Gewichtung des Gradienten entfällt. Dieses Verfahren beruht auf dem sogenannten Surplus-Konsensusverfahren, siehe [114], welches gegenüber dem Push-Sum-Verfahren allerdings den Nachteil hat, dass eine zusätzliche Schrittweite für eine schnelle Konvergenz des Konsensusverfahrens optimiert werden muss. Globale Nebenbedingungen können bei dem Optimierungsalgorithmus aus [104] über eine Projektion eingebunden werden.

In der Veröffentlichung [107] wird ein Algorithmus mit konstanter Schrittweite vorgestellt, bei dem über einen primal-dualen Ansatz die Konsensusnebenbedingung (4.2d) berücksichtigt wird, während durch eine Projektion lokale Beschränkungsmengen eingebunden werden. Der Algorithmus verwendet doppelt-stochastisch gewichtete Kommunikationsgraphen für das Konsensusverfahren, lässt sich allerdings nicht direkt in eines der beiden Grundverfahren (4.4) und (4.5) überführen.

In [20][21][22] werden Methoden zur Lösung von unbeschränkten Problemen vorgestellt, die unter Verwendung eines Nesterov Gradientenabstieges eine Beschleunigung der Konvergenzgeschwindigkeit erreichen können. Für die Methoden wird von doppelt-stochastisch gewichteten Kommunikationsstrukturen ausgegangen.

In der Veröffentlichung [105] wird ein Algorithmus zur Lösung von beschränkten, verteilten Optimierungsproblemen vorgestellt, bei denen das Konsensusprotokoll nicht die Optimierungsvariable x schätzt, sondern den Wert der globalen Kostenfunktion $\sum_{i=1}^N f_i(x)$ sowie den Wert der Funktionen der Nebenbedingungen. Diese Schätzwerte werden verwendet, um lokal durch einen primal-dualen Ansatz die verteilte Problemstellung zu lösen.

4.3.2 Stand der Forschung: Zusammenfassung

In Tabelle 4.1 sind die für dieses Kapitel relevantesten Veröffentlichungen bezüglich beschränkter, verteilter Optimierung zusammengetragen. Neben den bereits besprochenen Eigenschaften der Kommunikationsgewichtung, der Schrittweite und der Methode zur Einbindung der Nebenbedingungen, sind dort zusätzlich weitere notwendige Anforderungen und Eigenschaften bezüglich des verteilten Optimierungsproblems aufgeführt. Darüber hinaus

sind die Annahmen und Eigenschaften der Algorithmen der Beiträge 1, 2 und 3 dieses Abschnittes dort zum Vergleich aufgeführt.

Bei Analyse der Tabelle können folgende Kernerkenntnisse des Forschungsstandes bezüglich der Einbeziehung von Nebenbedingungen, der Kommunikationsgewichtung und der notwendigen Wahl der Schrittweite gewonnen werden:

1. Lokale Nebenbedingungen können bisher nur bei gleichzeitiger Annahme von doppelt-stochastisch gewichteter Kommunikation berücksichtigt werden.
2. Ebenso ist der verteilte primal-duale Ansatz im gesetzten Kontext ausschließlich unter doppelt-stochastischer Kommunikation bewiesen.
3. Es existieren nur wenige Verfahren zur Lösung von beschränkten, verteilten Problemen, bei denen konstante Schrittweiten verwendet werden können.

Tabelle 4.1: Übersicht einer Auswahl von Eigenschaften der für diese Arbeit relevantesten Algorithmen.

*: Nur wenn alle Gewichte des Graphen gleich sind.

√: wird gefordert/ist vorhanden, ×: wird nicht gefordert/ist nicht vorhanden,

+ : doppelt-stochastisch, — : reihen-stochastisch, | : spalten-stochastisch.

proj: Projektion, μ : Primal-dualer Ansatz, Ψ : Straftermansatz.

Weitere Abkürzungen: konv.: konvex, gl. konv.: gleichmäßig konvex.

	[100]	[101]	[84]	[82]	[83]	[106]	[29]	Beitr. 1	Beitr. 2	Beitr. 3
$f_i(x)$	konv.	konv.	konv.	konv.	konv.	konv.	gl. konv.	konv.	gl. konv.	gl. konv.
∇ Lip. stetig	×	×	√	√	√	√	√	√	√	√
Nbd.-Menge \mathcal{X}	√	√	√	√	√	√	√	×	√	√
Nbd. $g_i(x) \leq 0$	×	×	×	×	×	√	×	√	√	×
Gewichtung \mathcal{G}	+	+		—	—	+	+			+
zeitvariante \mathcal{G}	×	√	×	×	√	√	×	√	√	×
lokale Nbd.	×*	√	×	×	×	√	×	√	√	×
globale Nbd.	√	×	√	√	√	√	√	×	√	√
α konstant	×	×	×	×	×	×	√	×	×	√
∇f_i beschränkt	×	√	×	√	√	×	×	√	√	√
Methode Nbd.	proj	proj	proj	proj	proj	proj/ μ	proj	Ψ	proj/ μ	proj

4.3.3 Beiträge

Beitrag 1 ([32]) Der erste Beitrag ist ein Algorithmus, der als eine direkte Erweiterung des verteilten Gradientenabstieges aus (4.4) bzw. des Push-Sum-basierten (Sub)-Gradientenabstieges aus [18] zur Berücksichtigung von Nebenbedingungen interpretiert werden kann.

Zur Einbeziehung der Nebenbedingungen wird die Straftermmethode aus Abschnitt 2.3 verwendet, wobei davon ausgegangen wird, dass die Beschränkungen als Ungleichungen, wie in Problem 3 für $\mathcal{X} \equiv \mathbb{R}^d$ definiert, vorliegen. Durch Ansetzen jeweils eines Strafterms für jede der Ungleichungen der Abbildungen $g_i(x)$ können die lokalen Nebenbedingungen berücksichtigt werden¹².

Da das Verfahren auf dem Push-Sum-Konsensus beruht, können die Kommunikationsmatrizen als spalten-stochastisch gewichtet und zeitvariant angenommen werden.

Damit ist das Push-Sum-Strafterm-Verfahren des ersten Beitrages in der betrachteten Klasse von Optimierungsproblemen der erste Algorithmus, der lokale Nebenbedingungen unter zeitvarianter, nicht-doppeltstochastisch gewichteter Kommunikation verarbeiten kann und stellt einen Beitrag bezüglich der ersten Forschungslücke der Zusammenfassung der Literaturübersicht in Unterabschnitt 4.3.2 dar.

Beitrag 2 ([33]) Im Algorithmus des Beitrages 1 muss zusätzlich zu der zeitvarianten Schrittweitenfolge auch eine Folge des Straftermparameters, siehe Abschnitt 2.3, festgelegt werden. Beide Folgen müssen aufeinander abgestimmt werden und deren Wahl hat bedeutende Auswirkung auf die Konvergenzgeschwindigkeit des Verfahrens. Allerdings gibt es wenige theoretische Anhaltspunkte, wie diese Folgen zu wählen sind. Um die Komplexität der Bestimmung der Folgenparameter und damit der notwendigen Berechnung im Vorfeld der eigentlichen Ausführung des verteilten Verfahrens zu verringern, wird daher als zweiter Beitrag ein primal-dualer Algorithmus vorgestellt. Dieser hat im Vergleich zu Beitrag 1 gleichwertige Annahmen bezüglich des Optimierungsproblems, mit Ausnahme der strengeren Annahme der gleichmäßigen Konvexität der lokalen Kostenfunktionen, und zusätzlich den Vorteil, dass die Strafparameterfolge entfällt, da die Ungleichungsnebenbedingungen über Lagrangeparameter in die Kostenfunktion eingebunden werden.

Auch dieser Algorithmus fußt auf dem durch Push-Sum-Konsensus er-

¹²Es können auch globale Ungleichungsnebenbedingungen verarbeitet werden, indem diese von jedem Agenten als lokal interpretiert werden.

weiteren, verteilten Gradientenabstieg, siehe [18], und erweitert diesen durch einen primal-dualen Ansatz, bei dem abwechselnd ein Schritt in Richtung des primalen und dualen Optimums vorgenommen wird. Globale Nebenbedingungen können dabei über eine Projektion berücksichtigt werden, während die lokale Nebenbedingungen in die Lagrangefunktion mit aufgenommen werden.

Im Vergleich zu dem Algorithmus aus [106], bei dem ebenfalls ein primal-duales, verteiltes Verfahren zum Einsatz kommt, erfordert der Algorithmus dieses Beitrages nur eine spalten-stochastische und keine doppelt-stochastische Gewichtungsmatrix des Kommunikationsgraphen. Zusätzlich können in dem Algorithmus dieses Beitrags zeitvariante Kommunikationsgraphen verarbeitet werden¹³. Damit werden im Vergleich zu [106] weniger Anforderungen an die Kommunikationsstruktur gestellt. Ein weiterer Unterschied ergibt sich in Bezug auf die Berücksichtigung von globalen und lokalen Nebenbedingungen. Während in diesem Beitrag der Gradientenschritt, wie auch in den Veröffentlichungen [29][82][83][84][100] auf die globale Beschränkungsmenge \mathcal{X} projiziert und die lokalen Ungleichungsbedingungen über das duale Problem berücksichtigt werden, ist in [106] genau das Gegenteil der Fall: Hier wird auf lokale Beschränkungsmengen projiziert und globale Ungleichungsbedingungen über Lagrangemultiplikatoren in die Kostenfunktion mit aufgenommen. Hierdurch sind die lokalen Lagrangefunktionen \mathcal{L}_i nicht nur durch die Variable x verkoppelt, sondern auch durch die Lagrangeparameter μ_i . Daraus folgt, dass in [106] eine zusätzliche Konsensusbildung bezüglich der lokal aktualisierten Lagrangemultiplikatoren notwendig ist, während dies für den Beitrag nicht der Fall ist.

Folglich stellt der Algorithmus einen Beitrag bezüglich des ersten und zweiten Punktes der Erkenntnisse der Literaturübersicht aus Unterabschnitt 4.3.2 dar.

Beitrag 3 ([34]) Die zwei vorausgegangenen Beiträge sowie viele weitere Arbeiten bezüglich verteilter, beschränkter Optimierung verwenden Schrittweiten, die, um Konvergenz zum Optimum zu erreichen, unter anderem als absinkende Schrittweitenfolge gewählt werden müssen. Die Wahl der Folge hat einen starken Einfluss auf die Konvergenzgeschwindigkeit und entsprechende Verfahren konvergieren im Allgemeinen langsamer als solche mit konstanten Schrittweiten [19]. Zusätzlich ist der Berechnungsaufwand im

¹³In [33] wird Konvergenz nur für den Fall statischer Kommunikationsgraphen nachgewiesen. Die Erweiterung des Konvergenzbeweises auf zeitvariante Graphen erfolgt in der vorliegenden Arbeit in Abschnitt 4.5.

Vorfeld zur Ausführung bei konstanten Schrittweiten geringer.

Der dritte Beitrag ist eine Variante¹⁴ des Algorithmus aus [34], der in Abschnitt 5.4 beschrieben wird. Auch hier wird, ähnlich wie in [29], eine Projektion des Gradientenschrittes mit dem *Gradient-Tracking*-Ansatz verknüpft. Im Gegensatz zu [29] reicht eine einzige konstante Schrittweite α zur Kontrolle des Gradientenschrittes aus und zusätzlich ist diese Schrittweite nur durch eine obere Schranke und nicht durch eine untere beschränkt. Dadurch ist die Konvergenz auch bei beliebig kleinen Schrittweiten sichergestellt, was die Wahl dieses Parameters vereinfacht.

Im Vergleich zu den Beiträgen 1 und 2 bzw. zu den Algorithmen des beschränkten *verteilten Gradientenabstieg*, ist der Vorteil der konstanten Schrittweite hervorzuheben. Zusätzlich ist es für diesen Algorithmus möglich, lineare Konvergenz zum Optimum nachzuweisen. Allerdings muss hierfür die im Vergleich zu Beitrag 1 und weiteren Verfahren, siehe Tabelle 4.1, strengere Annahme der gleichmäßigen Konvexität der lokalen Kostenfunktionen getroffen werden. Das Verfahren konvergiert, wie auch das Verfahren in [29], auf statischen Graphen mit doppelt-stochastischen Kommunikationsmatrizen und es können ausschließlich globale Beschränkungen berücksichtigt werden.

Beitrag 4 Der vierte Beitrag umfasst die Anwendung der Algorithmen der Beiträge 1, 2 und 3 auf eine Variante des in Kapitels 3 beschriebenen Einsatzplanproblems innerhalb von Microgrids. Weitere Analysen des Konvergenzverhaltens der Algorithmen finden sich in den Veröffentlichungen [32][33][34] und [35].

4.4 Algorithmus 1: Verteilter Gradientenabstieg mit Straftermen

In diesem Abschnitt wird der Beitrag 1 in Form eines Algorithmus vorgestellt, mit dem das beschränkte, verteilte Optimierungsproblem 3 mit $\mathcal{X} \equiv \mathbb{R}^d$ optimal gelöst werden kann. Der behandelte Algorithmus basiert auf dem *verteilten Gradientenabstieg* aus den Gleichungen (4.4) bzw. dessen Erweiterung in [18]. Die Berücksichtigung der Nebenbedingungen erfolgt

¹⁴Der in [34] veröffentlichte Algorithmus ist für die Anwendung in Multiclusterspielen ausgelegt, siehe Kapitel 5. Verteilte Problemstellungen der Form aus Problem 3 sind ein Spezialfall von Multiclusterspielen, wenn bei diesen nur ein einziger Cluster existiert.

unter Verwendung von Straftermen, siehe hierzu auch den Grundlagenabschnitt 2.3. Durch die Verwendung des Push-Sum-Protokolls, siehe Grundlagenunterabschnitt 2.6.1, ist die Konvergenz auf spalten-stochastisch gewichteten und zeitvarianten Kommunikationsgraphen möglich.

Im Folgenden wird zunächst die Problemstellung unter Verwendung von Straftermen formuliert, bevor der Algorithmus vorgestellt und anschließend dessen Konvergenz bewiesen wird. Um die Übersichtlichkeit des Abschnittes zu verbessern, findet sich ein Großteil der mathematischen Beweise in Anhang A.

Die Formulierung des Algorithmus sowie der zugehörige Konvergenzbeweis sind in [32] veröffentlicht. In dieser Veröffentlichung ist der Autor der vorliegenden Arbeit Zweitautor, weswegen durchweg die genannte Veröffentlichung als Quelle angegeben wird.

4.4.1 Problemformulierung unter Verwendung von Straftermen

Die Grundlage der Problemformulierung in diesem Abschnitt ist das Optimierungsproblem 3 und es wird zusätzlich davon ausgegangen, dass alle Nebenbedingungen ausschließlich als lokale Ungleichungsbedingungen $g_i(x) \leq 0$ bei den Agenten i vorliegen. Das bedeutet insbesondere, dass für die Beschränkungsmenge $\mathcal{X} \equiv \mathbb{R}^d$ in Problem 3 gilt, insofern die Menge nicht durch Ungleichungen beschrieben werden kann. Um die Notation zu vereinfachen, wird im Folgenden davon ausgegangen, dass die lokalen Beschränkungsfunktionen $g_i(x)$ skalare Funktionen¹⁵ sind, d. h. es gilt $g_i : \mathbb{R}^d \rightarrow \mathbb{R}$ und $M_i = 1$.

Unter Verwendung von Straffunktionen $\Psi_i(x)$, die der Definition 2.3.2 entsprechen, für die Beschränkung $g_i(x) \leq 0$ des Agenten i ergibt sich das von der Iteration $k \in \mathbb{N}_0$ abhängige Optimierungsproblem

$$\min_x F_k(x) = \min_x \sum_{i=1}^N (f_i(x) + r_k \Psi_i(x)), \quad (4.6)$$

wobei r_k die iterationsabhängige Strafparameterfolge mit ausschließlich positiven Elementen darstellt. Bezüglich der verteilten Lösung des Problems wird davon ausgegangen, dass neben den lokalen Kostenfunktionen

¹⁵Für den mehrdimensionalen Fall des Vektors der Beschränkungsfunktionen $g_i(x)$ aus Gleichung (4.3b) wird für jede Dimension eine eigene Straffunktion $\Psi_{ij}(x)$ angesetzt und als Summe $\sum_{j=1}^{M_i} \Psi_{ij}(x)$ zur Kostenfunktion hinzugefügt. Die Konvergenzargumentation erfolgt analog zum skalaren Fall.

$f_i(x)$ auch die lokalen Nebenbedingungen $g_i(x) \leq 0$ und damit die Straffunktion $\Psi_i(x)$ privat bleiben sollen. Entsprechend sind diese nur dem Agenten i bekannt. Es gilt die Definition $\Psi(x) = \sum_{i=1}^N \Psi_i(x)$.

Als Straffunktion wird in der vorliegenden Arbeit

$$\Psi_i(x) = \begin{cases} \ln\left(\frac{e^{g_i(x)} + e^{-g_i(x)}}{2}\right), & \text{wenn } g_i(x) > 0, \\ 0, & \text{sonst,} \end{cases} \quad (4.7)$$

verwendet, da diese einen beschränkten Gradienten besitzt, wenn die Gradienten der Nebenbedingungsfunktionen $\nabla g_i(x)$ gleichmäßig beschränkt sind. Diese Eigenschaft ist für den Konvergenzbeweis notwendig, siehe Unterabschnitt 4.4.3. Eine andere Wahl der Straftermfunktionen ist unter Erfüllung der für die Konvergenz notwendigen Annahmen selbstverständlich möglich. Unter Verwendung von konvexen Straffunktionen $\Psi_i(x)$ sind unter Berufung auf die Pauschalannahme 4.1 auch die Funktionen $F_k(x)$ für alle $k \in \mathbb{N}_0$ konvex.

\mathcal{S} beschreibe die Menge der Lösungen von Problem 3 mit $\mathcal{X} \equiv \mathbb{R}^d$ und \mathcal{S}_k die Menge der Lösungen von Problem (4.6) für eine bestimmte Realisierung des Strafparameters r_k . Bezüglich der Beziehung des Problems 3 und der unbeschränkten Formulierung unter Verwendung von Straftermen in (4.6) wird folgender Satz aus [32] zitiert:

Satz 4.4.1 (Proposition 1 [32]) *Die Funktion $F(x)$ aus Problem 3 sei inf-kompakt¹⁶ und $r_k \rightarrow \infty$ für $k \rightarrow \infty$. Dann sind die Lösungsmengen \mathcal{S} , \mathcal{S}_k nicht leer und \mathcal{S}_k konvergiert zu \mathcal{S} , wenn $k \rightarrow \infty$. Zusätzlich gilt, dass $\lim_{k \rightarrow \infty} F_k^* = F^*$, wobei $F_k^* = \min_x F_k(x)$ und F^* dem optimalen Lösungswert von Problem 3 mit $\mathcal{X} \equiv \mathbb{R}^d$ entspricht.*

Unter Berücksichtigung des Schätzungsproblems der Agenten bezüglich der Variablen x lässt sich nun die verteilte Problemstellung aus Problem 4 unter Verwendung der Strafterme als zeitabhängiges Optimierungsproblem angeben, dass ausschließlich die Konsensusforderung als Nebenbedingung besitzt. Es gilt

$$\min_{(x_i)_{i \in \mathcal{I}}} \sum_{i=1}^N (f_i(x_i) + r_k \Psi_i(x_i)) \quad (4.8a)$$

$$\text{s.t. } x_i = x_j, \forall i, j \in \mathcal{I}. \quad (4.8b)$$

¹⁶Eine Funktion $F : \mathbb{R}^d \rightarrow \mathbb{R}$ wird als inf-kompakt bezeichnet, wenn die Menge $\{x \in \mathbb{R}^d : F(x) \leq a\}$ kompakt ist für alle $a \in \mathbb{R}$ [32].

Befinden sich die Schätzungen im Konsensus zueinander, d. h. die Bedingung (4.8b) ist erfüllt, reduziert sich das Problem auf die Form aus (4.6).

4.4.2 Algorithmus

Nachdem nun geklärt ist, wie die Strafterme in das Problem integriert werden, stellt sich die Frage, wie gleichzeitig ein Konsensus der Schätzungen x_i und die Konvergenz der Schätzungen zum Optimum x^* sichergestellt werden kann.

Um einen Konsensus zu erreichen, müssen die Agenten untereinander Informationen austauschen können. Wie in der Einleitung zu diesem Kapitel beschrieben und in Abbildung 4.1 dargestellt, sind die Agenten durch einen Graphen miteinander vernetzt, der mit $\mathcal{G} \{\mathcal{V} \equiv \mathcal{I}, \mathcal{E}, W\}$ bezeichnet wird. Die Matrix $W \in \mathbb{R}^{N \times N}$ repräsentiert die gewichteten Kanäle zwischen den Agenten, die sich in der Kantenmenge \mathcal{E} befinden. In Abschnitt 2.6 ist beschrieben, wie unter Verwendung der Gewichtungsmatrix W ein Aktualisierungsprotokoll für die Schätzungen x_i generiert werden kann, sodass diese asymptotisch einen Mittelwert erreichen. Dort wird auch auf die Bedeutung der Wahl der Gewichtung der Matrix W eingegangen und wie diese verteilt erzeugt werden kann. Als besonders effizient hat sich das sogenannte Push-Sum-Protokoll herausgestellt, da zur Generierung der für dieses Protokoll erforderlichen spalten-stochastischen Gewichtung der Matrix W die Agenten ausschließlich ihren eigenen Ausgangsgrad δ_i^- kennen müssen und die Konsensusbildung auch unter zeitvarianten Kommunikationskanälen $\mathcal{E}(k)$ und damit zeitvarianten Matrizen $W(k)$ erfolgen kann.

Unter entsprechender Wahl der Matrixgewichtung für W konvergiert das Push-Sum-Protokoll, siehe Gleichungen (2.56), zu einem Mittelwert-Konsensus, d. h. zu dem Mittelwert der Startzustände $x_i(0)$. Um das verteilte Problem zu lösen, muss der Konsensuswert x_c nun so beeinflusst werden, dass dieser in Richtung der Minima der Subprobleme F_k vorschoben wird und für $k \rightarrow \infty$ schließlich zur Lösung x^* von Problem 3 für $\mathcal{X} = \mathbb{R}^d$ konvergiert. Da die Subprobleme aufgrund der Pauschalannahmen 4.1, 4.2 und der Wahl der Straffunktion nach Gleichung (4.7) konvex und differenzierbar sind, bietet sich für einen verteilten Ansatz ein Gradientenverfahren an, wie es auch in [18] für unbeschränkte Optimierungsprobleme mit dem Push-Sum-Konsensus kombiniert wird. So motiviert sich der folgende Push-Sum-basierte Algorithmus 1 mit zeitabhängiger Schrittweitenfolge α_k und der Straftermfolge r_k .

Algorithmus 1 Verteilter Gradientenabstieg mit Straftermen [32].

Initialisierung: $y_i(0) = 1, w_i(0) = z_i(0) = x_i(0) = x_{i,0} \in \mathbb{R}^d, \forall i \in \mathcal{I}$.

Iteriere bis zur Konvergenz: Jeder Agent i berechnet

$$y_i(k+1) = \sum_{j=1}^N W_{ij}(k)y_j(k), \quad (4.9a)$$

$$w_i(k+1) = \sum_{j=1}^N W_{ij}(k)x_j(k), \quad (4.9b)$$

$$z_i(k+1) = \frac{w_i(k+1)}{y_i(k+1)}, \quad (4.9c)$$

$$x_i(k+1) = w_i(k+1) - \alpha_k [\nabla f_i(z_i(k+1)) + r_k \nabla \Psi_i(z_i(k+1))]. \quad (4.9d)$$

In der Gewichtungswariablen $y_i(k) \in \mathbb{R}^1$ wird der rechte Eigenvektor der Kommunikationsmatrix $W(k)$ geschätzt. Diese Schätzung erfolgt in der Gleichung (4.9a) durch die Kommunikation der eigenen Schätzung $y_i(k)$ der Agenten $i \in \mathcal{I}$ über die spalten-stochastisch gewichteten Kommunikationskanäle des Graphen. Jeder Agent aktualisiert dann seine Schätzung unter Bildung einer gewichteten Summe aus seiner eigenen Schätzungen zum Zeitpunkt k und den empfangenen Schätzungen seiner direkten Eingangsnachbarn. Die gleiche Aktualisierung findet in Gleichung (4.9b) bezüglich der Aktualisierung der Schätzung der Optimierungsvariablen statt. Das Ergebnis wird in der Hilfsvariablen $w_i(k) \in \mathbb{R}^d$ gespeichert, die aufgrund der Asymmetrie der Kommunikation über eine spalten-stochastische Matrix in der Gleichung (4.9c) unter Verwendung von $y_i(k)$ gewichtet werden muss. Das Ergebnis wird in der Variablen $z_i(k+1) \in \mathbb{R}^d$ festgehalten. Diese wird wiederum verwendet, um den lokalen Gradienten $\nabla f_i(z_i(k+1))$ und den Gradienten des lokalen Strafterms $\nabla \Psi_i(z_i(k+1))$ auszuwerten. In Gleichung (4.9d) wird schließlich der verteilte Gradientenschritt unter Verwendung der Schrittweite α_k und des Strafparameters r_k durchgeführt. Das Ergebnis dieses Schrittes, gespeichert in $x_i(k+1)$, wird im nächsten Schritt an die Ausgangsnachbarn kommuniziert und so die Gradientenaktualisierung in das Push-Sum-Konsensusverfahren, das durch die Gleichungen (4.9a) - (4.9c) repräsentiert wird, eingeschleust.

Die Einhaltung der Beschränkungen durch den Straftermansatz funktioniert wie im zentralen Fall, siehe Grundlagenabschnitt 2.3: Es sei zum Zeitpunkt k bei Agent i die Ungleichungsbedingung verletzt, d. h. es gilt $g_i(z_i(k)) > 0$. Dann nimmt die Straffunktion nach Gleichung (4.7) einen Wert $\Psi_i(z_i(k)) > 0$ an. Entsprechend steigt auch der Wert des Gradienten der Funktion an, vergleiche hierzu die Abbildung 2.3 im Grundlagenkapitel. Der Straftermgradient wird über den Strafparameter r_k in den Gradientenabstieg des Verfahrens in Gleichung (4.9d) hinzugefügt. Dieser Anteil des Gesamtgradient bewirkt somit eine Verschiebung des durch den Gradientenschritt aktualisierten Zustandes in Richtung des erlaubten Bereiches $g_i(x) \leq 0$. Der Anteil der Straffunktion am Gesamtgradienten wird dabei über den Strafparameter r_k gesteuert. Wächst dieser Strafparameter über die Zeit an, so wird der Einfluss des Gradienten der Straffunktion immer größer. Wird die Schätzung $z_i(k)$ wieder in die Beschränkung $g_i(z_i(k)) \leq 0$ geschoben, fällt der Wert der entsprechenden Straffunktion bzw. des Gradienten auf null. Die Tatsache, dass durch dieses Verfahren die Beschränkungen bei dem jeweiligen Agenten i , der als einziger Agent die Ungleichungsfunktion $g_i(x)$ kennt, eingehalten werden und dass der Algorithmus 1 gegen das Optimum des Problems 3 mit $\mathcal{X} \equiv \mathbb{R}^d$ konvergiert, wird im nachfolgenden Abschnitt bewiesen.

Für eine kompaktere Darstellung können die auf den Agenten i bezogenen Vorschriften von Algorithmus 1 vektorisiert werden. Dazu werden zunächst die Vektoren

$$\begin{aligned} x(k) &= \text{col}((x_i(k)^T)_{i \in \mathcal{I}}) \in \mathbb{R}^{N \times d}, \\ y(k) &= \text{col}((y_i(k))_{i \in \mathcal{I}}) \in \mathbb{R}^N, \\ z(k) &= \text{col}((z_i(k)^T)_{i \in \mathcal{I}}) \in \mathbb{R}^{N \times d}, \\ \nabla f(z(k)) &= \text{col}((\nabla f_i(z_i(k))^T)_{i \in \mathcal{I}}) \in \mathbb{R}^{N \times d}, \\ \nabla \Psi(z(k)) &= \text{col}((\nabla \Psi_i(z_i(k))^T)_{i \in \mathcal{I}}) \in \mathbb{R}^{N \times d}, \end{aligned}$$

definiert. Unter Verwendung der Matrix

$$Q(k) = \text{diag}(y(k+1))^{-1} W(k) \text{diag}(y(k)) \quad (4.10)$$

kann nun die Variable $w_i(k)$ durch Einsetzen der Aktualisierungsvorschrift (4.9d) für Zeitschritt k in Gleichung (4.9b) und Einsetzen des Re-

sultats in Gleichung (4.9c) eliminiert werden. Es ergibt sich die Form

$$y(k+1) = W(k)y(k), \quad (4.11a)$$

$$z(k+1) = Q(k)x(k), \quad (4.11b)$$

$$x(k+1) = z(k) - \alpha_k \text{diag}(y(k+1))^{-1} (\nabla f(z(k+1)) + r_k \nabla \Psi(z(k+1))). \quad (4.11c)$$

4.4.3 Konvergenz

Zunächst werden einige für die Beweisführung notwendigen Annahmen bezüglich Problemstellung und Algorithmus vorgenommen. Anschließend werden Präliminarien präsentiert, bevor der eigentliche Konvergenzbeweis in Theorem 4.4.1 durchgeführt werden kann.

Annahmen

Neben den nachfolgend aufgeführten Annahmen gelten zusätzlich die Pauschalannahmen 4.1¹⁷ und 4.2. Zunächst wird der Kommunikationsgraph betrachtet, der die Agenten miteinander verbindet. Es wird dabei von einem zeitvarianten Kommunikationsgraphen $\mathcal{G}\{\mathcal{V}, \mathcal{E}(k), W(k)\}$ ausgegangen, der die Agenten untereinander verbindet.

Annahme 4.4.1 *Der Kommunikationsgraph $\mathcal{G}\{\mathcal{V}, \mathcal{E}(k), W(k)\}$ ist zeitvariant. Die Graphsequenz $\{\mathcal{G}\{\mathcal{V}, \mathcal{E}(k), W(k)\}\}$ ist B -stark verbunden.*

Die Definition von B -starker Verbundenheit findet sich in Definition 2.5.3. Diese Annahme ist erforderlich, damit jeder Agent von jedem anderen Agenten innerhalb des endlichen Intervalls der Länge B Informationen erhalten kann und somit Informationen aller Agenten in die Konsensusbildung über das Push-Sum-Protokoll einfließen können.

Annahme 4.4.2 *Die nicht-negative, gewichtete Adjazenzmatrix $W(k)$ des Graphen $\mathcal{G}\{\mathcal{V}, \mathcal{E}(k), W(k)\}$ hat positive Diagonaleinträge $W(k)_{ii} > 0, \forall i \in \mathcal{I}$. Die Einträge $W(k)_{ij}$ der Matrix sind so gewählt, dass die Matrix spaltenstochastisch ist und somit $\mathbf{1}^T W(k) = \mathbf{1}^T$ gilt.*

Wie in Abschnitt 2.5 diskutiert kann eine spaltenstochastische Gewichtung erzeugt werden, indem jeder Agent i seinen Ausgangsgrad δ_i^- kennt,

¹⁷Wie in der Problemformulierung in Unterabschnitt 4.4.1 beschrieben, wird keine globale Beschränkungsmenge $\mathcal{X} \subset \mathbb{R}^d$ berücksichtigt bzw. es gilt $\mathcal{X} \equiv \mathbb{R}^d$. Entsprechend ist \mathcal{X} auch nicht wie in der Pauschalannahme 4.1 gefordert kompakt.

der gegebenenfalls auch mit einem verteilten Verfahren ermittelt werden kann. Somit stellt die notwendige spalten-stochastische Gewichtung keine Einschränkung der vollständig verteilten Lösung des Verfahrens dar.

Bezüglich der Gradienten der lokalen Kostenfunktionen und der Beschränkungsfunktionen wird folgendes angenommen:

Annahme 4.4.3 Die Gradienten $\nabla f_i(x)$ und $\nabla g_i(x)$ sind gleichmäßig beschränkt auf \mathbb{R}^d für alle $i \in \mathcal{I}$.

Aus der gleichmäßigen Beschränkung der Nebenbedingungsfunktionen folgt für die in Gleichung (4.7) definierte Straffunktion, dass deren Gradient $\nabla \Psi_i(x)$ ebenfalls gleichmäßig beschränkt ist für alle $i \in \mathcal{I}$. Das bedeutet, dass eine positive Konstante M_i existiert, sodass $\|\nabla \Psi_i(x)\| \leq M_i$ für alle $x \in \mathbb{R}^d$. Ebenso existiert nach obiger Annahme eine positive Konstante L_i , sodass $\|\nabla f_i(x)\| \leq L_i$ für alle $x \in \mathbb{R}^d$. Zusätzlich sei $M = \max_{i \in \mathcal{I}} M_i$ und $L = \max_{i \in \mathcal{I}} L_i$.

Annahme 4.4.4 Die Gradienten $\nabla f_i(x)$ und $\nabla g_i(x)$ sind lipschitzstetig auf \mathbb{R}^d für alle $i \in \mathcal{I}$.

Anmerkung 4.4.1 Unter Wahl der Straffunktion nach Gleichung (4.7) folgt aus der Lipschitzstetigkeit von $\nabla g_i(x)$ die Lipschitzstetigkeit der Gradienten der entsprechenden Straffunktionen $\nabla \Psi_i(x)$. Somit gilt $\|\nabla f_i(x_1) - \nabla f_i(x_2)\| \leq l_i \|x_1 - x_2\|$ und $\|\nabla \Psi_i(x_1) - \nabla \Psi_i(x_2)\| \leq m_i \|x_1 - x_2\|$. Ferner sei $l = \max_{i \in \mathcal{I}} l_i$ und $m = \max_{i \in \mathcal{I}} m_i$.

Sei \mathcal{C} die Beschränkungsmenge, definiert durch die Ungleichungsfunktionen $g_i(x)$, $i \in \mathcal{I}$. Für den Konvergenzbeweis wird die Implikation $\nabla \Psi(x) = \sum_{i=1}^N \nabla \Psi_i(x) = 0 \implies \Psi(x) = \sum_{i=1}^M \Psi_i(x) = 0$ benötigt. Da für den Gradienten der Straffunktion für $x \notin \mathcal{C}$

$$\nabla \Psi(x) = \sum_{i=1}^N \frac{e^{2g_i(x)} - 1}{e^{2g_i(x)} + 1} \nabla g_i(x) \quad (4.12)$$

gilt, wird die folgende Annahme benötigt:

Annahme 4.4.5 Für jeden Vektor $x \notin \mathcal{C}$ sind die Vektoren $\{\nabla g_i(x)\}$ linear unabhängig mit den jeweiligen Koeffizienten $\left\{ \frac{e^{2g_i(x)} - 1}{e^{2g_i(x)} + 1} \right\}$ für alle $i \in \mathcal{I}$.

Diese Annahme kann jedoch folgendermaßen relaxiert werden: Es existiert kein Punkt $x \notin \mathcal{C}$, sodass $\nabla g_i(x) = 0$ und $\nabla \Psi_i(x) = 0$, $\forall i \in \mathcal{I}$.

Daher gilt $\nabla\Psi(x) = 0$ für einen Punkt $x \notin \mathcal{C}$ dann und nur dann, wenn die Vektoren $\{\nabla g_i(x)\}$ linear abhängig mit den Koeffizienten $\left\{\frac{e^{2g_i(x)}-1}{e^{2g_i(x)}+1}\right\}$ sind. Um diese lineare Abhängigkeit zu umgehen, können lokale Faktoren k_i gewählt und mit den lokalen Beschränkungsfunktionen $\Psi_i(x)$ multipliziert werden, sodass $\nabla\Psi(x) = \sum_{i=1}^N k_i \frac{e^{2g_i(x)}-1}{e^{2g_i(x)}+1} \nabla g_i(x) \neq 0$ für alle $x \notin \mathcal{C}$ gilt.

Bezüglich der Schrittweiten- sowie die Strafparameterfolge soll die folgende Annahme gelten:

Annahme 4.4.6 Die positive Schrittweitenfolge α_k steigt nicht an und die aus ihr generierte Reihe divergiert, d. h.

$$\alpha_k \leq \alpha_s, \forall k \geq s, \text{ und } \sum_{k=0}^{\infty} \alpha_k = \infty.$$

Für die Elemente der Strafparameterfolge gilt

$$r_k > 1, \forall k, \text{ und } r_k \rightarrow \infty \text{ für } k \rightarrow \infty.$$

Zusätzlich gilt $\sum_{k=0}^{\infty} \alpha_k^2 r_k^3 < \infty$ und $r_{k+1} - r_k = o(\alpha_k)^{18}$, wenn $\alpha_k \rightarrow 0$. In Kombination gilt folglich $\lim_{k \rightarrow \infty} \alpha_k = 0$.

Eine mögliche Wahl für α_k und r_k , welche die obigen Bedingungen erfüllt, ist

$$\alpha_k = \frac{1}{k^{0,5+b}} \text{ und } r_k = k^{0,25b}, \quad (4.13)$$

für $0 < b < 0,4$.

Präliminarien

In Anhang A.1 sind zwei Theoreme zu finden, welche die Beweisführung stützen. Teil (a) von Theorem A.1.1 beweist beispielsweise, dass die notwendige Konsensusbedingung (4.2d) für Algorithmus 1 unter gegebenen Annahmen für $k \rightarrow \infty$ erfüllt wird. Dieses Ergebnis wird auch im Hauptkonvergenzbeweis aufgegriffen.

Eine wichtige Größe für den Konvergenzbeweis ist der zeitliche Mittelwert (engl. running average) $\bar{x}(k) \in \mathbb{R}^d$ der Schätzungen $(x_i(k))_{i \in \mathcal{I}}$, der

¹⁸Das bedeutet, dass $r_{k+1} - r_k$ langsamer wächst als α_k , siehe Abschnitt zur Notation zu Beginn der vorliegenden Arbeit.

durch

$$\bar{x}(k) = \frac{1}{N} \sum_{i=1}^N x_i(k) \quad (4.14)$$

definiert wird. Unter Algorithmus 1 gilt für diesen

$$\begin{aligned} \bar{x}(k+1) &= \bar{x}(k) - \frac{\alpha_k}{N} \sum_{i=1}^N [\nabla f_i(z_i(k+1)) + r_k \nabla \Psi_i(z_i(k+1))] \\ &= \bar{x}(k) - \alpha_k \left(\sum_{i=1}^N [\nabla f_i(\bar{x}(k)) + r_k \nabla \Psi_i(\bar{x}(k))] \right) \\ &\quad - \alpha_k \sum_{i=1}^N \left[\frac{1}{N} \nabla f_i(z_i(k+1)) - \nabla f_i(\bar{x}(k)) \right] \\ &\quad - \alpha_k r_k \sum_{i=1}^N \left[\frac{1}{N} \nabla \Psi_i(z_i(k+1)) - \nabla \Psi_i(\bar{x}(k)) \right], \end{aligned} \quad (4.15)$$

siehe auch [18].

Für die nachfolgenden Überlegungen wird die Notation

$$\mathbf{q}(k, \bar{x}(k)) = \sum_{i=1}^N \left[\frac{1}{N} \nabla f_i(z_i(k+1)) - \nabla f_i(\bar{x}(k)) \right] \quad \text{und} \quad (4.16)$$

$$\mathbf{p}(k, \bar{x}(k)) = \sum_{i=1}^N \left[\frac{1}{N} \nabla \Psi_i(z_i(k+1)) - \nabla \Psi_i(\bar{x}(k)) \right] \quad (4.17)$$

verwendet. Im folgenden Lemma werden die Normen dieser Vektoren beschränkt.

Lemma 4.4.1 [32] *Es gelten die Annahmen 4.4.2, 4.4.3, 4.4.4 und 4.4.6. Dann existieren die Skalare $q(k)$ und $p(k)$, sodass unter Algorithmus 1 die folgenden Beschränkungen gelten:*

$$\|\mathbf{q}(k, \bar{x}(k))\| \leq \frac{l}{N} \sum_{i=1}^N \|z_i(k+1) - \bar{x}(k)\| = q(k), \quad (4.18)$$

$$\|\mathbf{p}(k, \bar{x}(k))\| \leq \frac{m}{N} \sum_{i=1}^N \|z_i(k+1) - \bar{x}(k)\| = p(k). \quad (4.19)$$

Zusätzlich gilt

$$\sum_{k=0}^{\infty} \alpha_k r_k^3 q(k) < \infty, \sum_{k=0}^{\infty} \alpha_k r_k^3 p(k) < \infty. \quad (4.20)$$

Der Beweis findet sich in Anhang A.2.

Anmerkung 4.4.2 Aus der Parameterwahl nach Annahme 4.4.6 und der Tatsache, dass unter den Annahmen 4.4.3, 4.4.4 und 4.4.6 nach Theorem A.1.1 aus Anhang A.1 sowohl $p(k)$ als auch $q(k)$ für $k \rightarrow \infty$ nach 0 konvergieren, folgt, dass $\sum_{k=0}^{\infty} \alpha_k^{q_1} r_k^{q_2} q^2(k) < \infty$, $\sum_{k=0}^{\infty} \alpha_k^{q_1} r_k^{q_2} p^2(k) < \infty$ und $\sum_{k=0}^{\infty} \alpha_k^2 r_k^2 p(k)q(k) < \infty$ für alle Ganzzahlen $q_1, p_1 \geq 1$, $q_2, p_2 \in \{0, 1, 2, 3\}$.

Hauptergebnis

Mit diesen Vorüberlegungen kann nun im nachfolgenden Theorem die Konvergenz von Algorithmus 1 bewiesen werden:

Theorem 4.4.1 ([32]) *Die Funktion $F(x)$ aus Problem 3 sei inf-kompakt. Zusätzlich gelten die Annahmen 4.4.1 - 4.4.6. Unter Aktualisierung durch Algorithmus 1 erreichen dann alle Variablen $z_i(k+1)$, $\forall i \in \mathcal{I}$, einen Konsensus für $k \rightarrow \infty$ und jeder Endpunkt dieses Konsensus entspricht einer Lösung des Problems 3 mit $\mathcal{X} \equiv \mathbb{R}^d$.*

Der Beweis dieses Theorems findet sich in Anhang A.3.

Gilt zusätzlich zu den im Theorem aufgeführten Annahmen, dass F strikt konvex ist, dann existiert eine eindeutige Lösung z^* von Problem 3. In diesem Fall impliziert Theorem 4.4.1, dass alle $z_i(k+1)$, $\forall i \in \mathcal{I}$, unter Aktualisierung nach Algorithmus 1 zum Optimum z^* konvergieren.

Letztendlich sei noch angemerkt, dass der vorgestellte Algorithmus zwar konvexe Ungleichungsbedingungen der Form $g(x) \leq 0$ erfordert, allerdings können die Ergebnisse auch auf lineare Gleichungsbedingungen $h(x) = 0$ erweitert werden, indem diese mit den beiden Ungleichungen $0 \leq h(x) \leq 0$ ersetzt wird.

4.4.4 Fazit

Durch den verteilten Gradientenabstieg mit Straftermansatz ist es möglich, lokale Ungleichungsbedingungen in den Optimierungsprozess mit

einzubinden. Da das zugrundeliegende Konsensusverfahren dem Push-Sum-Konsensusprotokoll entspricht, können spalten-stochastisch gewichtete und zeitvariante Kommunikationsmatrizen verwendet werden.

Ein Vorteil dieses Verfahrens ist der geringe Rechenaufwand. Im Vergleich zu Projektionsmethoden, siehe Grundlagenabschnitt 2.3 bzw. die nachfolgenden Abschnitte, müssen keine rechenintensiven Schritte in Algorithmus 1 durchgeführt werden: Unter den Aktualisierungsvorschriften müssen die Agenten, neben der Auswertung der Gradienten von lokaler Kostenfunktion und Straftermfunktion, ausschließlich Vektoradditionen und Multiplikation von Skalaren mit Vektoren durchführen. Damit benötigen die Agenten nur wenig Rechenleistung, um an diesem Optimierungsverfahren teilnehmen zu können.

Ein Nachteil dieser Methode ist, dass die Wahl der Strafparameterfolge entscheidend für die Konvergenzgeschwindigkeit ist, allerdings wenig theoretische Anhaltspunkte für deren optimale Wahl existieren, siehe hierzu auch [54]. In Algorithmus 1 tritt zudem ein weiterer Faktor auf, der das Konvergenzverhalten maßgeblich beeinflusst: Die Schrittweitenfolge α_k . In der für die Konvergenz notwendigen Annahme 4.4.6 werden die beiden Folgen charakterisiert. Nach dieser Annahme muss die Schrittweite α_k unter anderem als absinkende Folge gewählt werden. Wie in [19] festgestellt, hat diese notwendige Wahl Nachteile bezüglich des Konvergenzverhaltens im Vergleich zu konstanten Schrittweiten bei verteilten Gradientenverfahren. Darüber hinaus sind die Folgen für Strafparameter und Schrittweite nach Annahme 4.4.6 zusätzlich untereinander verkoppelt, was die optimale Wahl dieser Folgen weiter erschwert.

Die Konvergenzrate lässt sich bisher zudem nicht theoretisch bestimmen. Eine Auswertung der praktischen Konvergenzgeschwindigkeit in Bezug auf eine Variante des Einsatzplanproblems aus Kapitel 3 erfolgt zusammen mit den nachfolgenden Algorithmen in dem Abschnitt 4.7.

4.5 Algorithmus 2: Verteilter primal-dualer Gradientenabstieg

In diesem Abschnitt wird der Beitrag 2 in Form eines Algorithmus zur Lösung der verteilten Problemstellung 3 vorgestellt. Ebenso wie der Algorithmus aus dem vorherigen Abschnitt, handelt es sich um eine Erweiterung des *verteilter Gradientenabstieges* (4.4) in Kombination mit dem Push-Sum-Konsensus Protokoll aus Unterabschnitt 2.6.1. Das Push-Sum-

Protokoll stellt hierbei sicher, dass sich auf spalten-stochastisch gewichteten und zeitvarianten Kommunikationsgraphen ein Konsensus bezüglich der Schätzungen der Agenten einstellt.

Während der Algorithmus 1 aus dem vorherigen Abschnitt die lokalen Ungleichungsbeschränkungen des Problems über Straffunktion zu der Kostenfunktion addiert, integriert der Algorithmus dieses Abschnittes die lokalen Nebenbedingungen über einen dualen Ansatz und die globalen Nebenbedingungen in Form einer Beschränkungsmenge \mathcal{X} über eine Projektion in das Optimierungsverfahren.

Abschnitt 2.3 und Unterabschnitt 2.6.1 enthalten für diesen Abschnitt relevante Grundlagen bezüglich primal-dualer-Verfahren, der Projektion auf Mengen und des Push-Sum-Konsensusprotokolls.

Im Folgenden wird zunächst das Problem als verteiltes Sattelpunktproblem formuliert, welches dann durch einen verteilten primalen Gradientenabstieg und einen dualen Gradientenaufstieg gelöst werden kann. Es folgt eine Beschreibung des Algorithmus und anschließend die Konvergenzdiskussion. Mit Ausnahme des zentralen Konvergenztheorems finden sich alle weiteren mathematischen Beweise zur Wahrung der Übersichtlichkeit im Anhang A. Veröffentlicht ist dieser Algorithmus in [33].

4.5.1 Formulierung als verteiltes Sattelpunktproblem

Das zu lösende Problem 3 wird im Folgenden als das *primale Problem* bezeichnet. Der optimale Funktionswert zum primal optimalen Lösungsvektor x^* sei F^* .

Für die Formulierung des dualen Problems nach Grundlagenabschnitt 2.3 wird zunächst die Lagrangefunktion

$$\mathcal{L}(x, \mu) = \sum_{i=1}^N f_i(x) + \mu_i^T g_i(x) = \sum_{i=1}^N \mathcal{L}_i(x, \mu_i) \quad (4.21)$$

mit den Lagrangeparametern $\mu_i \in \mathcal{M}_i^0 = \{\mu_i \in \mathbb{R}^{M_i} \mid \mu_i \geq 0\}$ und $\mu = \text{col}((\mu_i)_{i \in \mathcal{I}})$ definiert. $\mathcal{L}_i(x, \mu_i) = f_i(x) + \mu_i^T g_i(x)$ für $i \in \mathcal{I}$ stellen die lokalen Lagrangefunktionen der lokalen Probleme der Agenten $i \in \mathcal{I}$ dar, die von der geteilten Variable x abhängen, aber aufgrund der Lokalität der Nebenbedingungen $g_i(x)$ eigene, lokale Lagrangeparameter μ_i besitzen. Die duale Zielfunktion ergibt sich entsprechend zu

$$G(\mu) = \inf_{x \in \mathcal{X}} \sum_{i=1}^N \mathcal{L}_i(x, \mu_i). \quad (4.22)$$

Die duale Lösung G^* , entspricht dem Supremum der dualen Zielfunktion, $G^* = \sup_{\mu \geq 0} G(\mu)$. Gilt starke Dualität, entspricht der dual optimale Funktionswert dem primalen und somit $G^* = F^*$.

Ziel ist es nun, ein primal-dual optimales Lösungspaar (x^*, μ^*) nach Definition 2.3.1 zu finden. Mit Hilfe des Sattelpunkttheorems 2.3.2 kann überprüft werden, ob ein Kandidatenpaar primal-dual optimal ist. Nach diesem Theorem ist $(x^* \in \mathcal{X}, \mu^* \geq 0)$ ein primal-dual optimales Paar, wenn das Paar ein Sattelpunkt der Lagrangefunktion ist, sodass

$$\mathcal{L}(x^*, \mu) \leq \mathcal{L}(x^*, \mu^*) \leq \mathcal{L}(x, \mu^*), \forall x \in \mathcal{X}, \mu \geq 0. \quad (4.23)$$

Eine intuitive Lösung des Sattelpunktproblems ist eine Minimierung der Lagrangefunktion $\mathcal{L}(x, \mu)$ bezüglich x und eine Maximierung bezüglich μ . Können die Gradienten der Lagrangefunktionen berechnet werden, bietet sich eine iterative Lösung, wie in den Gleichungen (2.31) aufgeführt, an, bei der ein Gradientenab- bzw. aufstieg mit einer Projektion des Vektors x auf \mathcal{X} und μ auf \mathcal{D} kombiniert wird, wobei \mathcal{D} der Beschränkungsmenge von μ entspricht.

Aufgrund der verteilten Formulierung des Optimierungsproblems 3 sind die lokalen Lagrangefunktionen $\mathcal{L}_i(x, \mu_i)$, $i \in \mathcal{I}$, allerdings nur den jeweiligen Agenten i bekannt. Entsprechend stehen die Gradienten $\nabla_x \mathcal{L}(x, \mu)$ und $\nabla_\mu \mathcal{L}(x, \mu)$ bezüglich der globalen Lagrangefunktion den jeweiligen Agenten nicht zur Verfügung. Zusätzlich muss das Schätzungsproblem bezüglich der primalen Variable x aus Problem 4 berücksichtigt werden. Entsprechend müssen die Agenten gemeinsam das Sattelpunktproblem

$$\min_{(x_i)_{i \in \mathcal{I}}} \sum_{i=1}^N \max_{\mu_i} \mathcal{L}_i(x_i, \mu_i), \quad (4.24)$$

$$\text{s.t. } \mu_i \geq 0, \quad (4.25)$$

$$x_i = x_j \in \mathcal{X}, \forall i, j \in \mathcal{I}, \quad (4.26)$$

lösen. Wichtig ist hierbei, dass die lokalen Lagrangefunktionen $\mathcal{L}_i(x, \mu_i)$ bezüglich der primalen Variable verkoppelt sind, aber nicht bezüglich des dualen Lagrangemultiplikators μ_i .

4.5.2 Algorithmus

Wie im vorherigen Abschnitt gezeigt, können die lokalen Nebenbedingungen des Problems über einen Lagrangemultiplikator zur Kostenfunktion

hinzugefügt werden, wodurch die Lagrangefunktion entsteht. Die Lösung des primalen und dualen Problems führt auf ein Sattelpunktproblem, dessen Lösung ein primal-dual optimales Paar (x^*, μ^*) nach Definition 2.3.1 ist.

Da die lokalen Kostenfunktionen $f_i(x)$ sowie die lokalen Nebenbedingungen $g_i(x)$ nur den jeweiligen Agenten $i \in \mathcal{I}$ bekannt sind, muss eine Kommunikationsstruktur zwischen den Agenten aufgebaut werden, über die eine Konsensbildung bezüglich der lokalen Schätzungen x_i der primalen Optimierungsvariablen x getätigt werden kann. Die zeitvariante Kommunikationsstruktur wird dabei durch den Graphen $\mathcal{G} \{\mathcal{V} \equiv \mathcal{I}, \mathcal{E}(k), W(k)\}$ beschrieben, wobei $\mathcal{E}(k)$ die Menge der zeitlich veränderlichen Kommunikationskanäle zwischen den Agenten darstellt. Wie auch für den Gradientenabstieg in Kombination mit Straftermen, siehe Abschnitt 2.3, wird hier das Push-Sum-Konsensusprotokoll für das Erreichen eines Konsensus bezüglich der Schätzungen verwendet, wobei davon ausgegangen wird, dass für jeden Zeitschritt die Gewichtungsmatrix $W(k)$ spalten-stochastisch gewichtet ist.

Der Algorithmus 2, mit dem das verteilte Sattelpunktproblem gelöst werden kann, ist eine Kombination aus dem beschriebenen Push-Sum-Protokoll und den zentralen Aktualisierungsvorschriften (2.31) zur Lösung von Sattelpunktproblemen.

Algorithmus 2 Verteilter primal-dualer Gradientenabstieg

Initialisierung: $y_i(0) = 1$, $z_i(0) = x_i(0) = x_0 \in \mathbb{R}^d$, $\mu_i(0) = \mu_{i,0} \in \mathbb{R}^{M_i}$, $\forall i \in \mathcal{I}$.

Iteriere bis zur Konvergenz: Jeder Agent i berechnet

$$y_i(k+1) = \sum_{j=1}^N W_{ij}(k)y_j(k), \quad (4.27a)$$

$$z_i(k) = \frac{1}{y_i(k+1)} \sum_{j=1}^N W_{ij}(k)y_j(k)x_j(k), \quad (4.27b)$$

$$x_i(k+1) = \text{proj}_{\mathcal{X}} \left[z_i(k) - \alpha_k \frac{\nabla_z \mathcal{L}_i(z_i(k), \mu_i(k))}{y_i(k+1)} \right], \quad (4.27c)$$

$$\mu_i(k+1) = \text{proj}_{\mathcal{M}_i} [\mu_i(k) + \alpha_k \nabla_{\mu} \mathcal{L}_i(z_i(k), \mu_i(k))]. \quad (4.27d)$$

Die ersten beiden Aktualisierungen des Algorithmus, Gleichungen (4.27a) und (4.27b), entsprechen den Konsensusaktualisierungen auf Basis des Push-Sum-Protokolls, das für eine erleichterte Vektorisierung des Algorithmus umformuliert ist. Im Vektor $y_i(k)$ des Konsensusprotokolls wird der rechte Eigenvektor der Kommunikationsmatrix $W(k)$ geschätzt, der für den Ausgleich der Asymmetrie der Kommunikation über eine spalten-stochastisch gewichtete Matrix notwendig ist. Diese Gewichtungen werden sowohl zur Berechnung der Optimierungsvariablen $z_i(k)$ in (4.27b) als auch zur Gewichtung des Gradienten der Lagrangefunktion in (4.27c) verwendet.

In Gleichung (4.27c) wird ein auf die Beschränkungsmenge \mathcal{X} projizierter Gradientenschritt der Lagrangefunktion bezüglich der primalen Variablen x unter Verwendung des Ergebnisses des Konsensussschrittes $z_i(k)$ durchgeführt. Die Länge des Schrittes wird durch die Schrittweitenfolge α_k gesteuert. Wichtig ist hierbei, dass die jeweiligen Gradienten mit der Gewichtung $1/y_i(k+1)$ versehen sind, um die Asymmetrie der Kommunikation über spalten-stochastische Matrizen auszugleichen. Aufgrund der Initialisierung $y_i(0) = 1, \forall i \in \mathcal{I}$, und aufgrund der Aktualisierung dieser Variablen durch die Gleichung (4.27a) ist sichergestellt, dass alle $y_i(k)$ positiv sind und explizit $y_i(k) \neq 0$ für alle $i \in \mathcal{I}$ gilt.

Im letzten Schritt (4.27d) erfolgt die Aktualisierung der dualen Variablen unter Verwendung eines Gradientenschrittes der Lagrangefunktion, ebenfalls mit der Schrittweite α_k , bezüglich der lokalen, dualen Variablen μ_i . Hierbei handelt es sich um einen Schritt in Richtung des positiven Gradienten und damit um einen Gradientenaufstieg. Der Schritt wird auf die Menge \mathcal{M}_i projiziert, welche die Beschränkungsmenge der dualen Variablen darstellt. Weitere Informationen dieser Mengen sind in der Annahme 4.5.1 bzw. der Anmerkung 4.5.1 festgehalten.

Folglich wird unter Algorithmus 2 in jedem Zeitschritt k zunächst eine Angleichung der Schätzungen der Optimierungsvariablen durchgeführt. Anschließend werden lokal bei den Agenten Gradientenschritte gemacht, wobei ein Gradientenabstieg zur Minimierung des primalen Problems und ein Gradientenaufstieg zur Maximierung des dualen Problems durchgeführt wird. Dies entspricht der Suche nach einem Sattelpunkt. Der Gradientenabstieg wird in der Variablen $x_i(k+1)$ festgehalten, bezüglich derer in der nächsten Iteration ein Konsensussschritt durchgeführt wird. Somit werden die lokalen Gradientenschritte im Gesamtoptimierungsvorgang berücksichtigt. Die lokalen Gradientenaufstiege bezüglich μ_i werden ausschließlich lokal zur Auswertung des Gradienten $\nabla_x \mathcal{L}_i(x_i, \mu_i)$ im nächsten Zeitschritt verwendet.

Die Darstellung des Algorithmus in vektorisierter Form erleichtert die Konvergenzanalyse des Verfahrens. Unter Verwendung der Vektoren bzw. Matrizen

$$\begin{aligned}
y(k) &= \text{col}((y_i(k))_{i \in \mathcal{I}}) \in \mathbb{R}^N, \\
x(k) &= \text{col}((x_i(k)^T)_{i \in \mathcal{I}}) \in \mathbb{R}^{N \times d}, \\
z(k) &= \text{col}((z_i(k)^T)_{i \in \mathcal{I}}) \in \mathbb{R}^{N \times d}, \\
\mu &= \text{col}((\mu_i)_{i \in \mathcal{I}}) \in \mathbb{R}^{N \times \max_{i \in \mathcal{I}} M_i}, \\
\mathbf{L}_x(k) &= \text{col}((\nabla_z \mathcal{L}_i(z_i(k), \mu_i(k))^T)_{i \in \mathcal{I}}) \in \mathbb{R}^{N \times d} \text{ und} \\
\mathbf{L}_\mu(k) &= \text{col}((\nabla_\mu \mathcal{L}_i(z_i(k), \mu_i(k))^T)_{i \in \mathcal{I}}) \in \mathbb{R}^{N \times \max_{i \in \mathcal{I}} M_i}
\end{aligned}$$

kann der Algorithmus 2 in kompakter und vektorisierter Form dargestellt werden:

$$y(k+1) = W(k)y(k), \quad (4.28a)$$

$$z(k) = Q(k)x(k), \quad (4.28b)$$

$$x(k+1) = \text{proj}_{\mathcal{X}} [z(k) - \alpha_k \mathbf{L}_x(k)], \quad (4.28c)$$

$$\mu(k+1) = \text{proj}_{\mathcal{M}} [\mu(k) + \alpha_k \mathbf{L}_\mu(k)], \quad (4.28d)$$

wobei $\mathcal{X} := \{x = \text{col}((x_i)_{i \in \mathcal{I}}) \mid x_i \in \mathcal{X}\}$ und $\mathcal{M} := \{\mu = \text{col}((\mu_i)_{i \in \mathcal{I}}) \mid \mu_i \in \mathcal{M}_i\}$. Die Matrix $Q(k) \in \mathbb{R}^{N \times N}$ ist eine neu gewichtete Version der spalten-stochastischen Matrix $W(k)$ und ist definiert durch

$$Q(k) = \text{diag}(y(k+1))^{-1} W(k) \text{diag}(y(k)). \quad (4.29)$$

Wie in Lemma 4.5.1 des nachfolgenden Abschnittes gezeigt wird, weist diese Matrix bei spalten-stochastischer Gewichtung von $W(k)$ immer eine reihen-stochastische Gewichtung auf.

4.5.3 Konvergenz

In diesem Abschnitt wird der Konvergenznachweis für Algorithmus 2 bezüglich der Problemstellung 3 erbracht. Aus notationstechnischen Gründen wird dabei hauptsächlich die vektorisierte Form aus den Gleichungen (4.28) verwendet. Darüber hinaus wird angenommen, dass $d = 1$ bezüglich der Dimension des Optimierungsvektors x sowie $M_i = 1$ bezüglich der Anzahl der lokalen Nebenbedingungsfunktion bei den Agenten $i \in \mathcal{I}$

gilt, um die Beschreibungen zu vereinfachen. Der Beweis lässt sich allerdings ohne Einschränkungen auf den mehrdimensionalen Fall erweitern. Zunächst werden jedoch einige Annahmen präsentiert, die notwendigerweise für die Problemstellung gelten müssen. Es bestehen darüber hinaus die Pauschalannahmen 4.1 und 4.2 bezüglich des Problems 3. Anschließend werden einige Präliminarien genannt, bevor der Hauptkonvergenz in Theorem 4.5.1 beschrieben wird.

Annahmen

Die erste Annahme bezieht sich auf die primale und duale Formulierung des Problems. Dabei wird, unter anderem, die Pauschalannahme 4.1 bezüglich der Konvexität des Problems verstärkt. Es wird nun folgendes gefordert:

Annahme 4.5.1 *Die Funktion $F(x)$ aus Problem 3 ist gleichmäßig konvex mit der Konstanten $\lambda > 0$ auf der Beschränkungsmenge \mathcal{X} . Der primal optimale Wert F^* ist endlich und die Dualitätslücke ist gleich null, sodass $F^* = G^*$. Es existieren kompakte Mengen $\mathcal{M}_i \subset \mathcal{M}_i^0$, $\forall i \in \mathcal{I}$, welche die dual optimalen Vektoren μ_i^* , $i \in \mathcal{I}$ beinhalten.*

Anmerkung 4.5.1 Die Menge $\mathcal{M}_i^0 \subset \mathbb{R}^{M_i}$ beschreibt die Menge der Lagrangemultiplikatoren, d. h. $\mu_i \in \mathcal{M}_i^0$. Im Allgemeinen ist diese Menge nicht kompakt.

Starke Dualität ist erfüllt, wenn beispielsweise die Slatorbedingung aus Satz 2.3.1 gilt. Ebenso gilt dann, dass die Mengen \mathcal{M}_i , welche die dual optimalen Vektoren beinhalten, kompakt sind, siehe Theorem 2.3.2 in [115]. Die Definition der gleichmäßigen Konvexität einer Funktion ist in Definition 2.2.3 zu finden.

Anmerkung 4.5.2 Aus der Annahme 4.5.1 bzw. der Kompaktheit von \mathcal{X} und \mathcal{M}_i , $i \in \mathcal{I}$, zusammen mit der Tatsache, dass $\nabla_x \mathcal{L}_i(x, \mu_i)$ und $\nabla_{\mu} \mathcal{L}_i(x, \mu_i)$ für alle $i \in \mathcal{I}$ kontinuierlich sind, folgt, dass die Gradienten $\nabla_x \mathcal{L}_i(x, \mu_i)$ und $\nabla_{\mu} \mathcal{L}_i(x, \mu_i)$ gleichmäßig beschränkt auf \mathcal{X} und \mathcal{M}_i sind. Das bedeutet $\exists L_x < \infty, L_{\mu_i} < \infty$, sodass $\|\nabla_x \mathcal{L}_i(x, \mu_i)\| \leq L_x$ und $\|\nabla_{\mu} \mathcal{L}_i(x, \mu_i)\| \leq L_{\mu_i}$, $\forall x \in \mathcal{X}, \mu_i \in \mathcal{M}_i, i \in \mathcal{I}$.

Anmerkung 4.5.3 Folglich gilt, dass für feste μ_i bzw. x die lokalen Lagrangefunktionen $\mathcal{L}_i(x, \mu_i)$ lipschitzstetig mit den Konstanten L_x, L_{μ_i} sind.

Die nächste Annahme bezieht sich auf den zeitvarianten Graph:

Annahme 4.5.2 *Der Kommunikationsgraph $\mathcal{G}\{\mathcal{V}, \mathcal{E}(k), W(k)\}$ ist zeitvariant. Die Graphsequenz $\{\mathcal{G}\{\mathcal{V}, \mathcal{E}(k)\}, W(k)\}$ ist B-stark verbunden.*

Die Definition für B-starke Verbundenheit findet sich in Definition 2.5.3. Zusätzlich gelte für die Gewichtungsmatrix $W(k)$ des Graphen in jedem Zeitschritt:

Annahme 4.5.3 *Die nicht-negative, gewichtete Adjazenzmatrix $W(k)$ des Graphen $\mathcal{G}\{\mathcal{V}, \mathcal{E}(k), W(k)\}$ hat positive Diagonaleinträge $W(k)_{ii} > 0, \forall i \in \mathcal{I}$. Die Einträge $W_{ij}(k)$ der Matrix sind so gewählt, dass die Matrix spaltenstochastisch ist und somit $\mathbf{1}^T W(k) = \mathbf{1}^T$ gilt.*

In Abschnitt 2.5 werden Möglichkeiten zur Generierung von spaltenstochastischen Gewichtungen diskutiert. Die letzten beiden Annahmen bezüglich der Kommunikationsstruktur sind identisch zu den Annahmen 4.4.1 und 4.4.2 des Algorithmus 1.

Schließlich muss die Schrittweitenfolge α_k notwendigerweise die folgende Annahme erfüllen:

Annahme 4.5.4 *Die nicht anwachsende, positive Schrittweitenfolge α_k besitzt die folgenden Eigenschaften:*

a) $\lim_{k \rightarrow \infty} \alpha_k = 0$, b) $\sum_{k=0}^{\infty} \alpha_k = \infty$, c) $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$.

Diese Annahme gilt beispielsweise für Folgen der Form $\alpha_k = \frac{c}{k^\gamma}, \forall k \geq 1$, mit $c > 0$ und $\gamma \in (\frac{1}{2}, 1]$.

Präliminarien

Es werden nun diverse Definitionen getätigt und Eigenschaften nachgewiesen, die für den Hauptkonvergenzbeweis des Algorithmus notwendig sind. Zunächst soll hierbei auf den Zusammenhang der Matrix $Q(k)$ mit dem Push-Sum-Konsensusprotokoll eingegangen und einige wichtige Eigenschaften dieser Matrix festgehalten werden.

Nach ihrer Definition entspricht die Matrix

$$Q(k) = \text{diag}(y(k+1))^{-1} W(k) \text{diag}(y(k))$$

einer Neugewichtung der spaltenstochastischen Matrix $W(k)$ unter Verwendung der Gewichte $y(k+1)$ und $y(k)$. Es sei die Matrix

$$\Phi(k, s) = Q(k)Q(k-1)\dots Q(s+1)Q(s) \in \mathbb{R}^{N \times N}, k > s \quad (4.30)$$

als das Produkt der Matrizen von Zeitpunkt s bis Zeitpunkt k definiert, wobei $\Phi(k, k) = Q(k)$. Das folgende Lemma fasst einige wichtige Eigenschaften dieser Matrix zusammen:

Lemma 4.5.1 *Es gelten die Annahmen 4.5.2 und 4.5.3 bezüglich des zeitvarianten Graphen und der gewichteten Kommunikationsmatrix $W(k)$. Die Matrizen $Q(k)$, definiert in Gleichung (4.29), und $\Phi(k, s)$, definiert in Gleichung (4.30), besitzen die folgenden Eigenschaften:*

1. $Q(k)$ und $\Phi(k, s)$ sind reihen-stochastisch für $0 \leq s \leq k$ und alle $k \geq 0$.
2. Es existiert ein nicht-negativer Vektor ϕ_s , sodass $\lim_{k \rightarrow \infty} \Phi(k, s) = \mathbf{1}\phi_s^T$ für beliebige, endliche Zeitpunkte s .

Der Beweis für dieses Lemma findet sich in Anhang A.4.

Im nächsten Lemma werden die obigen Ergebnisse verwendet, um die Konvergenz der Elemente der Matrix $\Phi(k, s)$ zu dem Mittelwert über die jeweiligen Spalten abzuschätzen:

Lemma 4.5.2 *Es gelten die Annahmen 4.5.2 und 4.5.3. Es existieren Konstanten $C > 0$ und $\lambda \in (0, 1)$, welche den Ausdruck*

$$\left| \Phi(k, s)_{ij} - \frac{1}{N} \sum_{i=1}^N \Phi(k, s)_{ij} \right| \leq C\lambda^{k-s} \quad (4.31)$$

für alle $i, j \in \mathcal{I}$, $0 \leq s < k$ und $\forall k$ erfüllen.

Der Beweis findet sich in Anhang A.5.

Diese Resultate sind essentiell für den Nachweis der Konsensusbildung unter den Agenten bezüglich ihrer Schätzungen x_j . Durch die Neugewichtung der nach Annahme 4.5.3 spalten-stochastischen Matrix $W(k)$ durch die Vektoren $y(k+1)$ und $y(k)$ entsteht in jedem Zeitschritt k eine reihen-stochastische Matrix $Q(k)$. Diese wird von den Agenten verwendet, um ihre lokalen Schätzungen zu aktualisieren, siehe Gleichung (4.28b). Wie im Grundlagenabschnitt 2.5 festgestellt, sind reihen-stochastische Matrizen konsensusbildend. Tatsächlich konvergiert die Multiplikation aller Matrizen $Q(k)$, ausgehend von einem Startzeitpunkt s , dargestellt durch die Matrix $\Phi(k, s)$, für $k \rightarrow \infty$ auf eine konstante Matrix, deren Zeilen durch den Vektor ϕ_s bestimmt sind, siehe zweiter Punkt des Lemmas 4.5.1.

In Lemma 4.5.2 wird unter Verwendung der Ergebnisse des Lemmas 4.5.1 der Abstand zwischen den Elementen der Matrix $\Phi(k, s)$ und dem Mittelwert über die Spalten dieser Matrix abgeschätzt. Da $\lambda \in (0, 1)$

gilt, konvergiert diese Differenz für $k \rightarrow \infty$ nach 0.

Würde nun eine Konsensusbildung ungestört, d. h. ohne Beeinflussung durch den Gradientenschritt bzw. die Projektion auf die Menge \mathcal{X} , ablaufen, so ergäbe sich ein Konsensus unter den Agenten, dessen Wert durch den Vektor ϕ_s und die jeweiligen Schätzungswerte $x_i(s)$ bestimmt ist.

Um den Einfluss des Gradientenschrittes mit Projektion aus den Gleichung (4.27c) und (4.27d) auf die Konsensusbildung zu analysieren, wird dieser durch die lokalen Terme $\epsilon_i^x(k) \in \mathbb{R}^d$ und $\epsilon_i^\mu(k) \in \mathbb{R}^{M_i}$, definiert durch

$$\epsilon_i^x(k) = \text{proj}_{\mathcal{X}} \left[z_i(k) - \alpha_k \frac{\nabla_z \mathcal{L}_i(z_i(k), \mu_i(k))}{y_i(k+1)} \right] - z_i(k), \quad (4.32)$$

$$\epsilon_i^\mu(k) = \text{proj}_{\mathcal{M}_i} \left[\mu_i(k) + \alpha_k \nabla_\mu \mathcal{L}_i(z_i(k), \mu_i(k)) \right] - \mu_i(k), \quad (4.33)$$

ausgedrückt. Die Terme können nun verwendet werden, um die Gleichungen (4.27c) und (4.27d) ohne den Projektionsoperator und Gradientenschritt in der Form

$$x_i(k+1) = z_i(k) + \epsilon_i^x(k), \quad (4.34a)$$

$$\mu_i(k+1) = \mu_i(k) + \epsilon_i^\mu(k), \quad (4.34b)$$

darzustellen. Die Gleichung (4.34a) kann somit als Konsensusaktualisierung bezüglich der Schätzungen x_i betrachtet werden, die durch den Term $\epsilon_i^x(k)$ gestört wird.

Das nachfolgende Lemma gibt eine Schranke bezüglich der Normen der Terme $\epsilon_i^x(k) \in \mathbb{R}^d$ und $\epsilon_i^\mu(k) \in \mathbb{R}^{M_i}$ an.

Lemma 4.5.3 *Es gelten die Annahmen 4.5.2 und 4.5.3. Für die Normen der Terme $\epsilon_i^x(k)$ und $\epsilon_i^\mu(k)$, $\forall i \in \mathcal{I}$, gilt die folgende obere Schranke unter Verwendung der Schranken L_x und L_{μ_i} der Gradientennormen aus Anmerkung 4.5.2:*

$$\|\epsilon_i^x(k)\| \leq \alpha_k \frac{L_x}{y_i(k+1)}, \quad (4.35)$$

$$\|\epsilon_i^\mu(k)\| \leq \alpha_k L_{\mu_i}. \quad (4.36)$$

Der Beweis findet sich in Anhang A.6, indem unter anderem durch die Ausnutzung der Eigenschaft der Nicht-Ausdehnung, siehe Gleichung

chung (2.15), die Projektion abgeschätzt wird. Aufgrund der Initialisierung in Algorithmus 2 gilt $y_i(k+1) > 0$, für alle $i \in \mathcal{I}$ und $k \in \mathbb{N}_0$.

In direkter Folge aus diesem Lemma lässt sich unter gegebener Annahme 4.5.4 die Implikationen

$$\lim_{k \rightarrow \infty} \alpha_k = 0 \implies \lim_{k \rightarrow \infty} \|\epsilon_i^x(k)\| = 0, \lim_{k \rightarrow \infty} \|\epsilon_i^\mu(k)\| = 0, \quad (4.37)$$

$$\sum_{k=0}^{\infty} \alpha_k^2 < \infty \implies \sum_{k=0}^{\infty} \alpha_k \|\epsilon_i^x(k)\| < \infty, \sum_{k=0}^{\infty} \alpha_k \|\epsilon_i^\mu(k)\| < \infty, \quad (4.38)$$

für alle $i \in \mathcal{I}$ ableiten. Damit geht der Einfluss der Gradienten für $k \rightarrow \infty$ nach 0, wodurch eine Konsensusbildung gewährleistet ist. Dieses Erkenntnis ist essentiell für das nachfolgende Lemma, welches die Konvergenz der Variablen $x_i(k)$, $\forall i \in \mathcal{I}$, zum iterationsaktuellen Mittelwert betrachtet:

Lemma 4.5.4 *Es gelten die Annahmen 4.5.2 und 4.5.3. Es sei $\bar{x}(k) = \frac{1}{N} \sum_{j=1}^N x_j(k)$ der Mittelwert von $x_i(k)$ für $i \in \mathcal{I}$ zum Zeitpunkt k . Dann gilt,*

1. *wenn Annahme 4.5.4 a) gegeben ist, dass $\lim_{k \rightarrow \infty} \|x_i(k) - \bar{x}(k)\| = 0$ und*
2. *wenn Annahme 4.5.4 c) gegeben ist, dass $\sum_{k=0}^{\infty} \alpha_k \|x_i(k) - \bar{x}(k)\| < \infty$.*

Der Beweis findet sich in Anhang A.7.

Hauptergebnis

Nach diesen Vorüberlegungen kann nun der eigentliche Konvergenzbeweis zu einem primal-dual optimalen Paar (x^*, μ^*) geführt werden. Hierfür werden die Normen $\|x(k) - x^*\|$ und $\|\mu(k) - \mu^*\|$ betrachtet, die im Grenzwert zu null konvergieren müssen. Zunächst wird dafür im nachfolgenden Satz die Veränderung der Normen zwischen zwei Zeitschritten begrenzt.

Satz 4.5.1 *Es gelten die Annahmen 4.5.2 und 4.5.3. Für ein primal-dual optimales Paar $(x^*, \mu^*) \in \mathcal{X} \times \mathcal{M}$ gilt für die Normen $\|x(k) - x^*\|$ und $\|\mu(k) - \mu^*\|$ unter Aktualisierung nach Algorithmus 2 die folgende*

Beschränkung für $k \geq k_0$:

$$\begin{aligned}
& \sum_{i=1}^N (y_i(k+1) \|x_i(k+1) - x^*\|^2 + \|\mu_i(k+1) - \mu_i^*\|^2) \\
& \leq \sum_{i=1}^N (y_i(k) \|x_i(k) - x^*\|^2 + \|\mu_i(k) - \mu_i^*\|^2) \\
& \quad - 2\alpha_k (\mathcal{L}(\bar{x}(k), \mu^*) - \mathcal{L}(x^*, \mu^*) + \mathcal{L}(x^*, \mu^*) - \mathcal{L}(x^*, \mu(k))) \\
& \quad + 2\alpha_k L_x N \sum_{i=1}^n \|x_i(k) - \bar{x}(k)\| + L_x^2 \alpha_k^2 \sum_{i=1}^N \frac{1}{w_i} + \alpha_k^2 \sum_{i=1}^N L_{\mu_i}^2,
\end{aligned}$$

wobei $\mu^* = \text{col}((\mu_1^*)^T, \dots, (\mu_N^*)^T)$ und w_i der Grenzwert der Variablen y_i ist, d. h. $\lim_{k \rightarrow \infty} y_i(k) = w_i$.

Der Beweis findet sich in Anhang A.8.

Bevor nun die Konvergenz des Algorithmus 2 für $k \rightarrow \infty$ nachgewiesen werden kann, wird ein weithin bekanntes Lemma bezüglich der Konvergenz von nicht-negativen Folgen zitiert, dessen deterministische Version wie folgt lautet:

Lemma 4.5.5 [116] *Seien $\{v_k\}_{k=0}^\infty$, $\{u_k\}_{k=0}^\infty$, $\{b_k\}_{k=0}^\infty$ und $\{c_k\}_{k=0}^\infty$ nicht-negative Folgen, sodass $\sum_{k=0}^\infty b_k < \infty$ und $\sum_{k=0}^\infty c_k < \infty$ sowie*

$$v_{k+1} \leq (1 + b_k)v_k - u_k + c_k, \forall k \geq 0.$$

Dann konvergiert die Folge v_k und es gilt $\sum_{k=0}^\infty u_k < \infty$.

Mit diesen Überlegungen kann nun das Haupttheorem formuliert werden:

Theorem 4.5.1 *Es gelten die Annahmen 4.5.1, 4.5.2, 4.5.3, und 4.5.4. Dann konvergieren die Schätzungen $x_i(k)$ und die Lagrangeparameter $\mu_i(k)$ unter Aktualisierung durch Algorithmus 2 im Grenzwert $k \rightarrow \infty$ zu einem primal-dual optimalen Paar $(x^*, \mu^*) \in \mathcal{X} \times \mathcal{M}$ für alle $i \in \mathcal{I}$.*

Beweis. Um Lemma 4.5.5 anwenden zu können, werden die Definitionen

$$\begin{aligned}
 v_k &= \sum_{i=1}^N (y_i(k) \|x_i(k) - x^*\|^2 + \|\mu_i(k) - \mu_i^*\|^2), \\
 u_k &= 2\alpha_k (\mathcal{L}(\bar{x}(k), \mu^*) - \mathcal{L}(x^*, \mu^*) + \mathcal{L}(x^*, \mu^*) \\
 &\quad - \mathcal{L}(x^*, \mu(k))), \\
 c_k &= 2\alpha_k L_x N \sum_{i=1}^N \|x_i(k) - \bar{x}(k)\| + L_x^2 \alpha_k^2 \sum_{i=1}^N \frac{1}{w_i} \\
 &\quad + \alpha_k^2 \sum_{i=1}^N L_{\mu_i}, \\
 b_k &= 0,
 \end{aligned}$$

getroffen. Um zu zeigen, dass die Reihe $\sum_{k=0}^{\infty} c_k$ konvergiert, wird sich auf den Punkt 2 aus Lemma 4.5.4 berufen, wo nachgewiesen wird, dass, unter gegebenen Annahmen, $\sum_{k=0}^{\infty} \alpha_k \|x_i(k) - \bar{x}(k)\| < \infty$ gilt. Es folgt somit

$$2L_x N \sum_{k=0}^{\infty} \alpha_k \sum_{i=1}^N \|x_i(k) - \bar{x}(k)\| < \infty.$$

Nach Annahme 4.5.4 gilt zusätzlich $\left(L_x^2 \sum_{i=1}^N \left(\frac{1}{w_i} + L_{\mu_i}\right)\right) \sum_{k=0}^{\infty} \alpha_k^2 < \infty$. Aus der Kombination dieser Ergebnisse folgt

$$\sum_{k=0}^{\infty} c_k < \infty$$

und Lemma 4.5.5 ist anwendbar, woraus wiederum folgt, dass

$$\begin{aligned}
 \exists \delta, \lim_{k \rightarrow \infty} v_k &= \lim_{k \rightarrow \infty} \sum_{i=1}^N (y_i(k) \|x_i(k) - x^*\|^2 + \|\mu_i(k) - \mu_i^*\|^2) \\
 &= \delta \geq 0,
 \end{aligned} \tag{4.39}$$

$$\begin{aligned}
 \sum_{k=0}^{\infty} u_k &= \sum_{k=0}^{\infty} 2\alpha_k (\mathcal{L}(\bar{x}(k), \mu^*) - \mathcal{L}(x^*, \mu^*) + \mathcal{L}(x^*, \mu^*) - \mathcal{L}(x^*, \mu(k))) \\
 &< \infty.
 \end{aligned} \tag{4.40}$$

Da nach Annahme 4.5.4 für die Schrittweitenfolge $\sum_{k=0}^{\infty} \alpha_k = \infty$ gilt, muss aus der Gleichung (4.40)

$$\lim_{k \rightarrow \infty} \mathcal{L}(\bar{x}(k), \mu^*) - \mathcal{L}(x^*, \mu^*) + \mathcal{L}(x^*, \mu^*) - \mathcal{L}(x^*, \mu(k)) = 0$$

folgen. Aufgrund der Sattelpunkteigenschaft (4.23) folgt $\forall k$, dass

$$\mathcal{L}(\bar{x}(k), \mu^*) - \mathcal{L}(x^*, \mu^*) \geq 0, \mathcal{L}(x^*, \mu^*) - \mathcal{L}(x^*, \mu(k)) \geq 0.$$

Daher gilt der obige Grenzwert dann und nur dann, wenn

$$\begin{aligned} \lim_{k \rightarrow \infty} \mathcal{L}(\bar{x}(k), \mu^*) &= \mathcal{L}(x^*, \mu^*), \\ \lim_{k \rightarrow \infty} \mathcal{L}(x^*, \mu(k)) &= \mathcal{L}(x^*, \mu^*). \end{aligned}$$

Nun werden die Teilfolgen $\{\bar{x}(k_l)\}$ und $\{\mu(k_l)\}$ der Folgen $\{\bar{x}(k)\}$ und $\{\mu(k)\}$ gewählt, sodass

$$\begin{aligned} \lim_{l \rightarrow \infty} \mathcal{L}(\bar{x}(k_l), \mu^*) &= \lim_{k \rightarrow \infty} \mathcal{L}(\bar{x}(k), \mu^*) = \mathcal{L}(x^*, \mu^*), \\ \lim_{l \rightarrow \infty} \mathcal{L}(x^*, \mu(k_l)) &= \lim_{k \rightarrow \infty} \mathcal{L}(x^*, \mu(k)) = \mathcal{L}(x^*, \mu^*). \end{aligned}$$

Aus der Konvergenz von v_k zu einer Konstante δ , siehe Gleichung (4.39), folgt, dass die Folgen $\{\bar{x}(k)\}$ und $\{\mu(k)\}$ beschränkt sein müssen. Daher können konvergierende Teilfolgen $\{\bar{x}(k_l)\}$ und $\{\mu(k_l)\}$ gewählt werden, sodass $\lim_{l \rightarrow \infty} (\bar{x}(k_l), \mu(k_l)) = (\hat{x}, \hat{\mu})$. Damit gilt, dass

$$\begin{aligned} \lim_{l \rightarrow \infty} \mathcal{L}(\bar{x}(k_l), \mu^*) &= \mathcal{L}(\hat{x}, \mu^*) = \mathcal{L}(x^*, \mu^*) \text{ und} \\ \lim_{l \rightarrow \infty} \mathcal{L}(x^*, \mu(k_l)) &= \mathcal{L}(x^*, \hat{\mu}) = \mathcal{L}(x^*, \mu^*). \end{aligned}$$

Da nach Annahme 4.5.1 die Gesamtkostenfunktion $F(x)$ gleichmäßig konvex auf \mathcal{X} ist, impliziert die Gleichung $\mathcal{L}(\hat{x}, \mu^*) = \mathcal{L}(x^*, \mu^*) = \min_{x \in \mathcal{X}} \mathcal{L}(x, \mu^*)$, dass $\hat{x} = x^*$. Die duale Lösbarkeit von $\hat{\mu}$, $\hat{x} = x^*$ und $\mathcal{L}(x^*, \hat{\mu}) = \mathcal{L}(x^*, \mu^*) = \max_{\mu \geq 0} \mathcal{L}(x^*, \mu)$ impliziert, dass $(\hat{x}, \hat{\mu}) = (x^*, \mu^*)$.

Aus dem ersten Teil von Lemma 4.5.4 und der Gleichung (4.39) resultiert unter obigen Ergebnissen

$$\begin{aligned} \delta &= \lim_{k \rightarrow \infty} \sum_{i=1}^N (y_i(k) \|x_i(k) - x^*\|^2 + \|\mu_i(k) - \mu_i^*\|^2) \\ &= \lim_{l \rightarrow \infty} \sum_{i=1}^N (y_i[t_l] \|\bar{x}[t_l] - x^*\|^2 + \|\mu_i[t_l] - \mu_i^*\|^2) \\ &= 0. \end{aligned}$$

Daher kann aufgrund von $y_i(k) > 0$ für alle $k \geq 0$ geschlossen werden, dass

$$\begin{aligned} \lim_{k \rightarrow \infty} \|x_i(k) - x^*\| &= 0, \\ \lim_{k \rightarrow \infty} \|\mu_i(k) - \mu_i^*\| &= 0 \text{ für alle } i \in \mathcal{I}. \end{aligned}$$

Dies schließt die Beweisführung. \square

4.5.4 Fazit

Wie im vorausgegangenen Abschnitt bewiesen, ist es unter Verwendung des Algorithmus 2 möglich, die Problemstellung 3 durch einen primal-dualen Ansatz verteilt zu lösen. Im Vergleich zu Algorithmus 1, bei dem die Nebenbedingungen über Strafterme berücksichtigt werden, entfällt für den primal-dualen Ansatz die Notwendigkeit der Definition einer Strafparameterfolge, die in Zusammenhang mit der Schrittweitenfolge für schnelle Konvergenz optimal gewählt werden muss. Die Abhängigkeit von einer nicht-anwachsenden Schrittweitenfolge bleibt allerdings bestehen. Zusätzlich können aufgrund der Projektion der lokalen Gradientenschritte auf die Beschränkungsmenge \mathcal{X} nun auch global bekannte Beschränkungsmengen effizient berücksichtigt werden. Die lokalen Ungleichungsnebenbedingungen bleiben hierbei privat.

Da im Allgemeinen bei jeder Auswertung des Projektionsoperators ein Minimierungsproblem gelöst werden muss, erhöht sich die Berechnungsaufwand jeder Iteration im Vergleich zum Staftermansatz aus Algorithmus 1. Für Nebenbedingungen in Form von beispielsweise Hyperebenen, Halbräumen und Rechtecken, siehe Grundlagenabschnitt 2.3, existieren allerdings analytische Lösungen des Projektionsoptimierungsproblems, die sich effizient auswerten lassen.

Ebenso wie für den Straftermalgorithmus steht auch für den primal-dualen Algorithmus der Nachweis der theoretischen Konvergenzrate noch aus. Eine praktische Überprüfung der Konvergenzgeschwindigkeit in Bezug auf das Simulationsmodell aus Kapitel 3 erfolgt im Abschnitt 4.7.

4.6 Algorithmus 3 - Projiziertes Gradient-Tracking-Verfahren

Beide Algorithmen der vorherigen Abschnitte basieren auf dem unbeschränkten, *verteilter Gradientenabstieg* aus Gleichung (4.4). Die Schritt-

weitenfolge ist dabei durch die Annahmen 4.4.6 und 4.5.4 als nicht ansteigende Folge festgelegt, deren Parametrierung auf jedes zu lösende Optimierungsproblem angepasst werden muss. Zudem ist bereits festgestellt, dass zeitvariante Schrittweite in der geforderten Form das verteilte Verfahren im Vergleich zu konstanten Schrittweiten verlangsamen, siehe hierzu die Literaturdiskussion im Abschnitt 4.3.

Der in diesem Abschnitt vorgestellte Algorithmus basiert auf dem verteilten *Gradient-Tracking* aus Gleichung (4.5), wodurch die Verwendung einer konstanten Schrittweite zur Festlegung der Länge des Gradientenschrittes möglich wird. Nebenbedingungen werden über einen Projektionsansatz in den Optimierungsprozess integriert, wofür alle Nebenbedingungen global und somit allen Agenten bekannt sein müssen.

Zu lösen ist die Problemstellung 3, wobei davon ausgegangen wird, dass alle Ungleichungsbedingungen $g_i(x) \leq 0$, $i \in \mathcal{I}$, in die globale Beschränkungsmenge absorbiert werden. Bei einer verteilten, kooperativen Problemstellung wie Problem 3 handelt es sich um einen Sonderfall von sogenannten Multiclusterspielen, die in Kapitel 5 behandelt werden. Bei diesen Problemen sollen verteilte Optimierungsprobleme der Form aus Problem 3 kooperativ gelöst werden, während Kopplungen unter den Optimierungsproblemen der Cluster nicht-kooperativ durch einen spieltheoretischen Ansatz betrachtet werden, siehe Problemstellung 5. Existiert allerdings nur ein einziger Cluster, reduziert sich diese Problemstellung auf die Problemstellung 3. Entsprechend stellen Multiclusterspiele eine Verallgemeinerung von rein kooperativen, verteilten Optimierungsproblemen dar. Ohne Einschränkung der Gültigkeit kann daher der in [34] für ein Multiclusterspiel entworfene Algorithmus auch auf die rein kooperative Problemstellung angewendet werden, wie in diesem Abschnitt motiviert werden soll.

Entsprechend empfiehlt es sich vor der Lektüre dieses Abschnittes zunächst den Abschnitt 5.4 aus Kapitel 5 sowie den Grundlagenabschnitt 2.3 zur Projektion zu lesen, da sich im Folgenden wiederholt auf den allgemeineren Ansatz der Multiclusterspiele bezogen wird.

Da der erweiterte Algorithmus in Kapitel 5 ausführlich besprochen wird, werden im Folgenden nur die grundlegenden Kenntnisse vermittelt.

4.6.1 Problemformulierung unter Verwendung einer globalen Nebenbedingungsmenge

Wie beschrieben werden die Nebenbedingungen des hier betrachteten Algorithmus unter Verwendung einer Projektion des lokalen Gradientenschrittes hinzugefügt. Damit ist im Vergleich zu den Verfahren aus den

Abschnitten 4.4 und 4.5 keine Umformulierung des Problems notwendig. Allerdings wird davon ausgegangen, dass alle Nebenbedingungen des Problems global, d. h. jedem Agenten bekannt, sind. In Bezug auf Problem 3 müssen somit entweder die lokalen Ungleichungsbedingungen $g_i(x) \leq 0$ wegfallen oder in die Beschränkungsmenge \mathcal{X} absorbiert werden. Folglich gilt es, das folgende Optimierungsproblem zu lösen:

$$\min_{(x_i)_{i \in \mathcal{I}}} \sum_{i=1}^N f_i(x_i), \quad (4.41a)$$

$$\text{s.t. } x_i = x_j \in \mathcal{X}, \forall i, j \in \mathcal{I}. \quad (4.41b)$$

Die lokalen Kostenfunktionen bleiben hierbei lokal und damit privat. Auch das Schätzungsproblem bezüglich der Optimierungsvariablen x bleibt erhalten.

4.6.2 Algorithmus

Um die verteilte Problemstellung aus (4.41) zu lösen, müssen die Agenten miteinander kommunizieren, um einen Konsensus bezüglich ihrer Schätzungen zu erreichen. Die Kommunikationsarchitektur wird durch einen statischen Graphen $\mathcal{G}\{\mathcal{V} \equiv \mathcal{I}, \mathcal{E}, W\}$ beschrieben, wobei von einer doppelt-stochastischen Gewichtung der Kommunikationsmatrix $W \in \mathbb{R}^{N \times N}$ ausgegangen wird. Da aufgrund der angenommenen Gewichtung die Kommunikation symmetrisch erfolgt, sind im Gegensatz zu den Algorithmen aus den Abschnitten 4.4 und 4.5 keine zusätzliche Konsensusbildung zur Generierung einer ausgleichenden Gewichtung notwendig. Allerdings stellt die doppelt-stochastische Gewichtung eine vor allem bei gerichteten, unbalancierten Graphen schwieriger zu generierende Gewichtung dar, siehe Diskussion in Grundlagenabschnitt 2.5.

Der hier vorgestellte Algorithmus entspricht einer Erweiterung des verteilten *Gradient-Tracking*-Algorithmus (4.5), der im Abschnitt zum Stand der Forschung diskutiert wird, durch eine Projektionsoperator dar, mit dem die Nebenbedingungen des Problems berücksichtigt werden können. Im Gegensatz zu *verteilten Gradientenabstiegen*, dem Basisalgorithmus der vorherigen Algorithmen, wird der Gradientenschritt nicht in Richtung des lokalen Gradienten durchgeführt, sondern in einer zusätzlichen Variablen $y_i(k)$ der Gesamtgradient $\sum_{i=1}^N \nabla_x f_i(x)$ geschätzt und in jeder Iteration ein über die konstante Schrittweite α gesteuerter Schritt in die geschätzte Richtung des steilsten Abstieges vollzogen. Für die Schätzung des Gesamtgradienten ist hierbei eine zusätzliche Konsensusbildung und damit

ein Kommunikationsschritt unter den Agenten notwendig. So ergibt sich der nachfolgende Algorithmus 3, der einer auf die rein kooperative Problemstellung reduzierten Variante des Lösungsalgorithmus aus Unterabschnitt 5.4.1 entspricht.

Algorithmus 3 *Gradient-Tracking-Algorithmus* mit Projektion.

Initialisierung: $x_i(0) = x_0 \in \mathbb{R}^d$, $y_i(0) = \nabla f_i(x_i(0)) \in \mathbb{R}^d$, $\forall i \in \mathcal{I}$.

Iteriere bis zur notwendigen Konvergenz: Jeder Agent i berechnet

$$\hat{x}_i(k) = \sum_{j=1}^N W_{ij} x_j(k), \quad (4.42a)$$

$$x_i(k+1) = \text{proj}_{\mathcal{X}} [\hat{x}_i(k) - \alpha y_i(k)], \quad (4.42b)$$

$$y_i(k+1) = \sum_{j=1}^N W_{ij} y_j(k) + \nabla f_i(x_i(k+1)) - \nabla f_i(x_i(k)). \quad (4.42c)$$

Die Variable $x_i(k) \in \mathbb{R}^d$ entspricht hierbei der aktuellen Schätzung des Agenten i bezüglich der Optimierungsvariablen x der verteilten Problemstellung, während $y_i(k) \in \mathbb{R}^d$, wie bereits beschrieben, der Schätzung von Agent i zum Zeitpunkt k bezüglich des Gesamtgradienten entspricht. Zu Beginn jeder Iteration erfolgt in Gleichung (4.42a) ein Konsensusschritt bezüglich der Schätzungen $x_i(k)$, bei dem jeder Agent seine Schätzung durch Bildung einer gewichteten Summe aus der Schätzung seiner Nachbarn und seiner eigenen Schätzung aktualisiert und in der Zwischenvariablen $\hat{x}_i(k) \in \mathbb{R}^d$ speichert. Im nächsten Schritt wird in Gleichung (4.42b) ein Gradientenschritt in Richtung der aktuellen Gradientenschätzung $y_i(k)$ durchgeführt, der anschließend auf die globale Beschränkungsmenge \mathcal{X} projiziert wird. Die Gradientenschätzung erfolgt in der Gleichung (4.42c) unter Auswertung des Gradienten der lokalen Kostenfunktion ∇f_i an den Stellen $x_i(k+1)$ und $x_i(k)$. Ein Austausch der Agenten untereinander ist notwendig, um die Schätzungen zu fusionieren und somit auf einen gemeinsamen Wert zu bringen, der dem Gesamtgradienten entspricht. Bei Initialisierung der lokalen Schätzungen mit $y_i(0) = \nabla f_i(x_i(0))$ kann unter entsprechender Gewichtung der Kommunikationsmatrix W gezeigt wer-

den, dass

$$\sum_{i=1}^N y_i(k) = \sum_{i=1}^N \nabla f_i(x_i(k)) \quad (4.43)$$

gilt. Das bedeutet, dass in jedem Zeitschritt k die Summe über alle Schätzungsvariablen des Systems dem summierten Gradienten der lokalen Kostenfunktionen, ausgewertet an den Schätzungen $x_i(k)$, entspricht. Konvergieren also sowohl die Gradientenschätzungen $y_i(k)$ als auch die Schätzungen $x_i(k)$ auf einen gemeinsamen Konsensuswert, entspricht der Konsensuswert y_c dem Mittelwert der Gradienten der lokalen Kostenfunktionen, ausgewertet am Konsensuswert x_c der Schätzung der Optimierungsvariablen. Selbstverständlich muss im Konvergenzbeweis die gegenseitige Beeinflussung der Aktualisierungen der beiden Schätzungen berücksichtigt werden.

Unter den Definitionen

$$x(k) = \text{col}((x_i(k)^T)_{i \in \mathcal{I}}) \in \mathbb{R}^{N \times d}, \quad (4.44)$$

$$y(k) = \text{col}((y_i(k)^T)_{i \in \mathcal{I}}) \in \mathbb{R}^{N \times d}, \quad (4.45)$$

$$G(k) = \text{col}((\nabla f_i(x_i(k)))_{i \in \mathcal{I}}^T) \in \mathbb{R}^{N \times d}, \quad (4.46)$$

$$\mathcal{X} = \{x | x_i \in \mathcal{X}\}, \quad (4.47)$$

kann Algorithmus 3 in der vektorisierten Form

$$\hat{x}(k) = Wx(k), \quad (4.48a)$$

$$x(k+1) = \text{proj}_{\mathcal{X}}[\hat{x}(k) - \alpha y(k)], \quad (4.48b)$$

$$y(k+1) = Wy(k) + G(k+1) - G(k), \quad (4.48c)$$

angegeben werden.

4.6.3 Konvergenz

Die Konvergenz des Algorithmus 3 folgt direkt aus der Konvergenzanalyse des Algorithmus 4 aus Kapitel 5 unter der Reduktion des Multiclusterproblems auf den vereinfachten Fall eines einzigen Clusters und damit der Problemstellung (4.41). Im Folgenden werden die für die Konvergenz notwendigen Annahmen aufgeführt und mit den Annahmen des Falls der Multiclusterspiele verglichen.

Existiert nur ein Cluster $\mathcal{H} = \{1\}$, so ist die Kombination der Pauschalannahmen 4.1 und 4.2 für kooperative, verteilte Optimierungsprobleme identisch mit der Pauschalannahme 5.1 für Multiclusterspiele. Dies folgt aus dem Wegfallen des spieltheoretischen Anteils des Multiclusterproblems und lässt sich leicht überprüfen. Die Pauschalannahmen 4.1 und 4.2 werden im Folgenden als gegeben vorausgesetzt.

Die Annahme

Annahme 4.6.1 *Die globale Kostenfunktion $F(x)$ ist gleichmäßig konvex auf \mathbb{R}^d mit einer Konstanten $\mu > 0$, sodass*

$$(\nabla F(x) - \nabla F(y))^T (x - y) \geq \mu \|x - y\|^2 \quad (4.49)$$

für $x, y \in \mathbb{R}^d$. Zusätzlich ist der Gradient $\nabla F(x)$ lipschitzstetig mit einer Konstanten $L_0 > 0$ auf \mathbb{R}^d , sodass

$$\|\nabla F(x) - \nabla F(y)\| \leq L_0 \|x - y\| \quad (4.50)$$

für alle $x, y \in \mathbb{R}^d$.

entspricht der Pauschalannahme 5.2, in der ein gleichmäßiges, monotonen Mapping $M(x)$ des Multiclusterspiels gefordert wird, siehe Definition 2.2.5. Bei Reduktion auf einen Cluster entspricht die Abbildung $M(x)$ dem Gradienten der Kostenfunktion $F(x)$, siehe hierzu deren Definition in Gleichung (5.3). Es gilt somit $M(x) = \nabla F(x)$ und $M(x)$ ist gleichmäßig monoton, wenn und nur wenn $F(x)$ gleichmäßig konvex ist, vergleiche hierfür die Definitionen 2.2.3 und 2.2.5.

Annahme 4.6.2 *Der konstante Graph $\mathcal{G}\{\mathcal{I}, \mathcal{E}, W\}$, der die Agenten untereinander verbindet, ist ungerichtet und verbunden oder gerichtet und stark verbunden. Die zugehörige nicht-negative, gewichtete Adjazenzmatrix W hat positive Diagonaleinträge $W_{ii} > 0, \forall i \in \mathcal{I}$. Die Einträge W_{ij} der Matrix sind so gewählt, dass die Matrix doppelt-stochastisch ist, d. h. es gilt $\mathbf{1}^T W = \mathbf{1}^T$ und $W \mathbf{1} = \mathbf{1}$.*

Diese Annahme ist unter $\mathcal{H} = \{1\}$ äquivalent zur Annahme 5.4.1.

Anmerkung 4.6.1 In diesem Verfahren sind doppelt-stochastische Matrizen notwendig. Zusätzlich wird davon ausgegangen, dass der Kommunikationsgraph konstant ist. Wie im Abschnitt 2.5 vorgestellt, lässt sich

jeder stark verbundene Graph mit einem geeigneten, verteilten Verfahren so gewichten, dass die Gewichtungsmatrix doppelt-stochastisch ist. Dieser Verfahren ist zwar aufwendig, siehe [31], allerdings kann bei einem konstanten Graphen jeder Agent einmal vor Beginn des Optimierungsalgorithmus die notwendige Gewichtung verteilt berechnen, z. B. durch ein Verfahren aus [61]. Damit hält sich der zusätzliche Aufwand in Grenzen. Bei einem zeitvarianten Graphen müsste in jeder Iteration vor dem Kommunikationsschritt die Gewichtung neu berechnet werden, was erheblich aufwendiger ist.

Das betrachtete Lösungskonzept des Multiclusterspiels ist das Nash-Gleichgewicht, welches für ein nicht-kooperatives Spiel in Gleichung (5.2) definiert ist. Existiert nur ein Cluster, reduziert sich diese Gleichung auf $F(x^*) \leq F(x)$, $\forall x \in \mathcal{X}$, was gleichbedeutend damit ist, dass x^* das Optimum des zu lösenden Optimierungsproblems ist. Da nach Annahme 4.6.1 $F(x)$ gleichmäßig konvex ist, gilt sogar eine strikte Ungleichung, da bei gleichmäßiger Konvexität der Kostenfunktion nur ein eindeutiges Optimum existiert. Durch die Reduzierung auf ein Cluster wird somit das Nash-Gleichgewichtsproblem in einfaches Optimierungsproblem überführt.

Die Äquivalenzen aus Lemma 5.2.1 zwischen Variationsungleichung und Nash-Gleichgewicht ergeben sich für das resultierende Optimierungsproblem unter den im Grundlagenabschnitt formulierten, zueinander äquivalenten Optimalitätsbedingungen 1 und 2 sowie der entsprechenden Anpassung der Zuordnungsmatrix \mathbf{Q} , definiert in Gleichung (5.6).

Der tatsächliche Konvergenzbeweis ergibt sich analog zum Konvergenzbeweis für das Multiclusterspiel aus Unterabschnitt 5.4.2 und soll an dieser Stelle nicht im Detail wiederholt werden. Daher wird der folgende Satz zur Konvergenz des Verfahrens angegeben, der analog zum Theorem 5.4.1 für den vereinfachten Fall eines einzigen Clusters formuliert ist:

Satz 4.6.1 *Es gelten die Pauschalannahmen 4.1 und 4.2 sowie die Annahmen 4.6.1 und 4.6.2. Dann hat das Problem (4.41) ein eindeutiges Optimum x^* .*

Die Schrittweite α des Algorithmus 3 sei so gewählt, dass $0 < \alpha < \bar{\alpha}$, wobei sich $\bar{\alpha}$ analog zu Satz 5.4.1 aus Parametern des Optimierungsproblem ergibt. Dann konvergieren alle Schätzungen x_i zum Optimum x^ , d. h. $\lim_{k \rightarrow \infty} x_i(k) = x^*$ für alle $i \in \mathcal{I}$ und die Konvergenzrate ist linear.*

Die Wahrheit dieses Satzes folgt aus der Kombination von Satz 5.4.1 und Theorem 5.4.1 für den Spezialfall $\mathcal{H} = \{1\}$.

4.6.4 Fazit

Auch wenn Algorithmus 3 und die dazugehörige Konvergenzargumentation nur Nebenprodukte des Algorithmus 4 aus Abschnitt 5.4 sind, so ist trotzdem die Bedeutung des Beitrags 3 nicht von der Hand zu weisen, da mit diesem Algorithmus erstmalig ein kooperatives, verteiltes Optimierungsproblem mit Nebenbedingungen unter Verwendung einer konstanten und beliebig kleinen Schrittweite gelöst werden kann. Dies wirkt sich theoretisch und praktisch positiv auf die Konvergenz aus: Es kann für den Algorithmus 3 eine lineare Konvergenzrate nachgewiesen werden, während dies für die Algorithmen 1 und 2 nicht möglich ist. Zusätzlich kann, wie auch im Abschnitt zum Stand der Forschung diskutiert, für Algorithmen mit konstanten Schrittweiten in der betrachteten Klasse von gradientenbasierten Algorithmen eine erhöhte Konvergenzgeschwindigkeit erwartet werden, was sich auch in den sich anschließenden Simulationen in Abschnitt 4.7 bestätigt.

Neben verbesserten Konvergenzeigenschaften kann der Algorithmus einfacher auf eine neue Problemstellung angepasst werden. Da ein Projektionsverfahren verwendet wird, müssen im Vergleich zu Algorithmus 1 keine auf die Problemstellung angepassten Strafterme gesucht werden und im Vergleich zu Algorithmus 2 keine Lagrangemultiplikatoren eingeführt werden. Zusätzlich kann bei gegebenen Annahmen immer eine beliebig kleine Schrittweite gefunden werden, für die der Algorithmus konvergiert, was die Suche nach einer optimalen Parametrierung der Schrittweite vereinfacht.

Allerdings kann Algorithmus 3 im Vergleich zu den Algorithmen 1 und 2 ausschließlich globale Nebenbedingung verarbeiten, wodurch jedem Agenten die gesamte Nebenbedingungsmenge bekannt sein muss. Dies kann bei entsprechenden Problemstellungen eine gewisse Einschränkung der Privatsphäre der Agenten bedeuten.

Darüber hinaus muss für den Informationsaustausch unter den Agenten ein Konsensusprotokoll mit doppelt-stochastisch gewichtetem, zeitinvariantem Graphen verwendet werden, während durch die Verwendung des Push-Sum-Konsensusprotokolls in den Algorithmen 1 und 2 eine Konvergenz auf spalten-stochastischen Graphen möglich ist, die darüber hinaus noch zeitvariant sein können. Wie im Grundlagenabschnitt 2.5 ausführlich besprochen wird, ist das Erstellen einer doppelt-stochastischen Gewichtung für einen gerichteten, nicht-balancierten Graphen zwar immer möglich, jedoch deutlich aufwendiger als ausschließliche Gewichtung der

Spalten oder Reihen des Graphen.

Letztlich kann, je nach Problemstellung, die Verwendung einer Projektion auf die gesamte Konvergenzzeit gesehen mathematisch aufwendiger sein, als die Einbindung der Nebenbedingungen durch Strafterme oder einen dualen Ansatz. Grund hierfür ist, dass bei Projektionen, wenn sich der zu projizierende Punkt außerhalb der Beschränkungsmenge befindet, immer ein Optimierungsproblem gelöst werden muss, bei dem ein Punkt innerhalb der Beschränkungsmenge gefunden werden muss, der den geringsten Abstand zum zu projizierenden Punkt hat. Nach Definition des Projektionsoperators in Gleichung (2.14) handelt es sich hierbei um ein beschränktes, quadratisches Problem. Weist die Beschränkungsmenge allerdings eine besondere Form auf, ist eine analytische Auswertung des Projektionsoperators möglich, siehe hierzu die Beispiele in den Gleichungen (2.16) bis (2.18).

4.7 Simulationen: Optimaler Generatoreinsatzplan

Um die theoretischen Betrachtungen zu komplementieren, werden im Folgenden die Algorithmen 1 bis 3 in Anwendung auf ein Generatoreinsatzplanproblem ausgewertet und miteinander verglichen. Diese Problemstellung basiert auf dem in Kapitel 3 vorgestellten Modell, das auf das kooperative Verhalten der Agenten angepasst ist.

Der Fokus der Analyse der Simulationsergebnisse liegt auf der Konvergenzgeschwindigkeit der Algorithmen. Dabei wird betrachtet, welcher Algorithmus nach einer festgelegten Iterationsanzahl den geringsten Fehler zum tatsächlichen Optimum aufweist. Zusätzlich wird die Skalierungsfähigkeit der Algorithmen abgeschätzt, indem das Konvergenzverhalten auf verschiedenen großen Agentensystemen miteinander verglichen wird. Daneben werden ebenfalls die Ergebnisse der untersuchten Einsatzplanprobleme diskutiert.

Dieser Vergleich der Algorithmen stellt den Beitrag 4 der vorliegenden Arbeit dar.

4.7.1 Simulationsmodell

Da es sich bei den zu untersuchenden Algorithmen um Methoden zur Lösung von kooperativen, verteilten Problemen handelt, wird der nicht-kooperative Anteil des Gesamtproblems (3.16) aus dem Kapitel 3

in den nachfolgenden Simulationen nicht betrachtet. Somit wird für die Simulation nur ein einziges Microgrid simuliert, das als Inselnetz betrieben wird und somit keine Verbindung zum Hauptnetz hat. In Abbildung 4.3 ist ein solches Microgrid beispielhaft dargestellt.

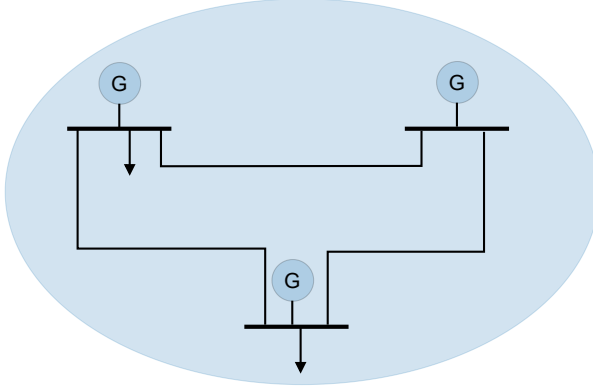


Abbildung 4.3: Beispielhafte Veranschaulichung eines Microgridmodells mit Generatoren und Lasten im Inselbetrieb nach der Formulierung aus Kapitel 3.

Um zusätzlich das Problem übersichtlicher zu gestalten, wird davon ausgegangen, dass ausschließlich Generatoren und Übertragungsleitungen aktive Bestandteile des Microgrids sind, für die eine optimale Aktion gefunden werden muss. Da mit dem Wegfall der Speicher auch die Kopplung der Zeitslots untereinander entfällt, wird im Folgenden nur ein Zeitslot betrachtet bzw. ein Zeithorizont von $T = 1$. Ausgehend von $i \in \mathcal{I} = \{1, \dots, N\}$ Bussen im System ergibt sich das folgende reduzierte, verteilte Optimierungsproblem:

$$\min_x F(x) = \min_x \sum_{i=1}^N f_i(x), \quad (4.51a)$$

$$\text{s.t. } g_i - \sum_{j \in \mathcal{N}_i^-} e_{ij} + \sum_{j \in \mathcal{N}_i^+} e_{ji} = v_i, \forall i \in \mathcal{I}, \quad (4.51b)$$

$$\underline{g}_i \leq g_i \leq \bar{g}_i, \forall i \in \mathcal{I}, \quad (4.51c)$$

$$-\bar{e}_{ij} \leq e_{ij} \leq \bar{e}_{ij}, \forall i, j \in \mathcal{E}^{\text{TB}}. \quad (4.51d)$$

Hierbei entsprechen die lokalen Kostenfunktionen der Summe der

Generator- und Übertragungskosten am jeweiligen Bus i , sodass

$$f_i(x) = f_i^G(g_i) + f_i^L(e) \text{ mit} \quad (4.52)$$

$$f_i^G(g_i) = a_{G,i}g_i^2 + b_{G,i}g_i + c_{G,i} \text{ und}$$

$$f_i^L(e) = \sum_{j \in \mathcal{N}_i^+} \frac{1}{2} a_{L,j} e_{ji}^2 + \sum_{j \in \mathcal{N}_i^-} \frac{1}{2} a_{L,j} e_{ij}^2. \quad (4.53)$$

In Unterabschnitt 4.4.1 der Problemformulierung mit Straftermen wird davon ausgegangen, dass alle Nebenbedingungen als Ungleichungen vorliegen. Die Leistungsbalance in (4.51b) stellt allerdings eine Gleichung dar. Das Ersetzen dieser Gleichung durch zwei Ungleichung ist zwar möglich, erhöht allerdings unnötigerweise die Anzahl der Nebenbedingungen des Problems. Tatsächlich reicht es aus, die Gleichung durch die Ungleichung

$$g_i - \sum_{j \in \mathcal{N}_i^-} e_{ij} + \sum_{j \in \mathcal{N}_i^+} e_{ji} \geq v_i, \forall i \in \mathcal{I}, \quad (4.54)$$

zu ersetzen. In Anhang A.9 wird die Äquivalenz der Lösungen der Problemstellung (4.51) mit Gleichung (4.51b) bzw. Ungleichung (4.54) unter Verwendung der KKT-Optimalitätsbedingungen gezeigt. Die Zulässigkeit der Substitution lässt sich auch intuitiv erklären: Ziel des Einsatzplanproblems ist es, die Kosten für die Leistungsversorgung des betrachteten Microgrids zu senken. Durch das Ersetzen der Balancegleichung mit der obigen Ungleichung ist es theoretisch möglich, mehr Leistung zu produzieren als verbraucht wird. Unter Berücksichtigung der Optimierungsziele ergibt dies allerdings keinen Sinn, da zusätzliche Leistungsproduktion immer mit zusätzlichen Kosten einhergeht.

4.7.2 Szenarien und Setup

Für die Simulationen werden drei unterschiedliche Szenarien betrachtet, die auf Beispielnetzwerken für Energieversorgungsnetze von Pandapower¹⁹ basieren, siehe [118]. Die Beispielnetzwerke stellen das Grundgerüst für die Szenarien in der vorliegenden Arbeit dar und sind entsprechend auf die Problemstellungen angepasst und ergänzt. Wie bei Leistungsflussbetrachtungen üblich, werden die Netzwerke als Bussysteme dargestellt, wobei die

¹⁹Pandapower ist eine Python-basierte open-source Bibliothek, mit der Energienetze modelliert, analysiert und optimiert werden können, siehe auch [117].

Busse als jeweilige Bilanzknoten der Leistungsflüsse durch Übertragungsleitungen untereinander verbunden sind.

In dem verteilten Optimierungsproblem stellen die jeweiligen Busse die Agenten des Netzwerkes dar. Für eine effiziente Darstellung der für Lastflussberechnungen üblichen Busnetzwerkdarstellungen werden diese in die für verteilte Optimierung übliche, agentenbasierte Darstellung überführt, wie in Abbildung 4.4 dargestellt.

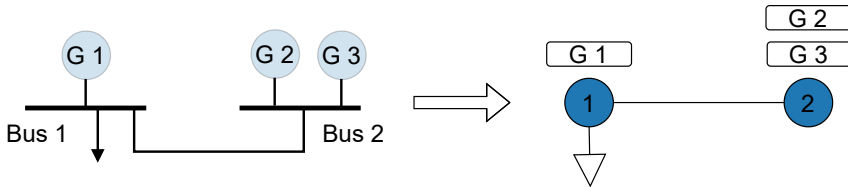


Abbildung 4.4: Verwendete Konvertierung der Busdarstellung der Lastflussberechnung in die Agentendarstellung der verteilten Optimierung.

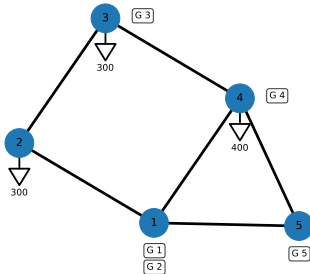
Die Parametrierung der verteilten Problemstellungen für die jeweiligen Busnetzwerke ist den Pandapowermodellen entnommen. Fehlende Informationen sind entsprechend ergänzt. Die Parameter für die jeweiligen Modelle sind im Anhang A.10 aufgeführt.

Das erste Szenario ist ein 5-Busnetzwerk, dargestellt als Agentensystem in Abbildung 4.5a, das insgesamt fünf Generatoren und drei Verbraucher umfasst. Dieses Netzwerk entspricht dem Case 5 der Pandapower Beispielnetzwerke und wird darüber hinaus in der Veröffentlichung [119] beschrieben.

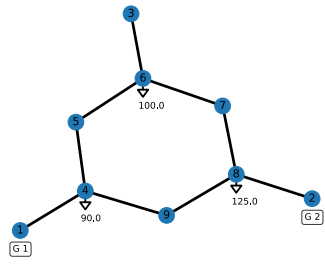
Das zweite Szenario besteht aus einem 9-Busnetzwerk, abgebildet in der Abbildung 4.5b. Dieses Netzwerk ist in [120] veröffentlicht, beinhaltet zwei Generatoren und drei Verbraucher und entspricht dem Pandapower Netzwerk Case 9.

Das dritte Szenario ist ein 30-Busnetzwerk, dargestellt in Abbildung 4.5c, basiert auf dem Washington 30 Bus Dynamic Test Case und entspricht dem Pandapower Netzwerk Case 30. Hier werden 20 Verbraucher von fünf Generatoren versorgt.

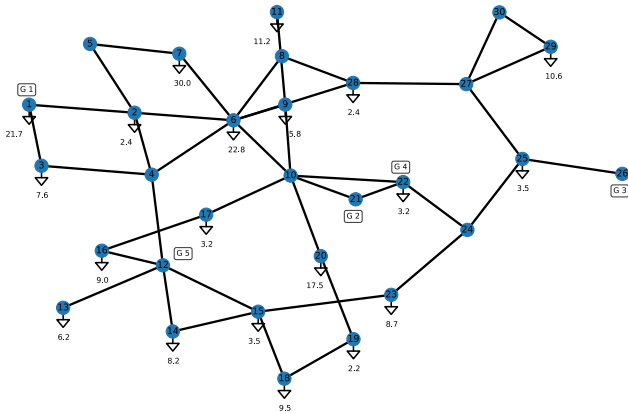
Für die Kommunikationsarchitektur werden für jedes Szenario zufällige, ungerichtete Graphen generiert. Die Darstellung der Graphen ist aus Gründen der Übersichtlichkeit in den Anhang A.11 verschoben.



(a) 5-Busnetzwerk.



(b) 9-Busnetzwerk.



(c) 30-Busnetzwerk.

Abbildung 4.5: Für die Simulationen verwendete Busnetzwerke, ausgewählt aus den Beispielnetzwerken von Pandapower [118].

Um die Konvergenzgeschwindigkeiten der Algorithmen miteinander vergleichen zu können, wird zunächst eine möglichst optimale Konfiguration der Schrittweite bzw. der Schrittweitenfolgen sowie weiterer, algorithmus-spezifischer Parameter identifiziert. Für Algorithmus 1 und 2 wird die Schrittweitenfolge durch

$$\alpha_k = \frac{\alpha}{k f_\alpha} \quad (4.55)$$

und damit durch die Parameter α und f_α bestimmt. Für Algorithmus 1 muss zusätzlich eine ansteigende Strafparameterfolge festgelegt werden, die sich durch

$$r_k = r \cdot k^{f_r} \quad (4.56)$$

und somit mittels der Parameter r und f_r bestimmt. Um diese Parameter möglichst optimal festzulegen, wird eine Rastersuche durchgeführt und der beste Kandidat für die nachfolgenden Versuche ausgewählt. Darüber hinaus werden für die Algorithmen 1 und 2 weitere, konstante Gewichtungen eingeführt, um die Gradienten der lokalen Kostenfunktionen stärker zu gewichten. Unter diesen Gewichtungen werden die Richtungen der Gradienten selbst nicht verändert, sondern ausschließlich die Länge der Vektoren angepasst, wodurch sich die Konvergenzzeit verbessern lässt.

In Problem (4.51) hängen die lokalen Kostenfunktionen f_i^g der Generatoren der jeweiligen Busse ausschließlich von der eigenen Produktion g_i ab. Zusätzlich tauchen die Optimierungsvariablen ausnahmslos in lokalen Nebenbedingungen auf. Da in den Algorithmen 1 und 2 die lokalen Nebenbedingungen, sprich die Generatorkapazitäten sowie die Leistungsbalance am jeweiligen Bus, auch ausschließlich lokal bei den betreffenden Bussen berücksichtigt werden, müssen die Optimierungsvariablen g_i nicht von den anderen Generatoren geschätzt werden und können lokal aktualisiert werden. Für den Algorithmus 3 ist das nicht der Fall: Bei diesem Algorithmus projizieren alle Agenten auf die gemeinsame Nebenbedingungsmenge, weswegen alle Agenten Schätzungen bezüglich der Leistungsproduktion der Generatoren der jeweils anderen Busse besitzen müssen.

Da die Leistungsflüsse auf den Übertragungsleitungen Kopplungsvariablen sowohl zwischen den Kostenfunktionen als auch zwischen Nebenbedingungen der Agenten darstellen, müssen diese geschätzt werden.

4.7.3 Lösung der Optimierungsprobleme

Um den Konvergenzverlauf der relativen Fehler für die Algorithmen auswerten zu können, werden im Vorfeld die Lösungen der Optimierungspro-

bleme der Szenarien ermittelt. Dafür wird von einem zentralen Fall ausgegangen, bei dem der Optimierer alle relevanten Informationen bezüglich des Optimierungsproblems besitzt und mit einer Standardmethode das Problem gelöst werden kann.

Zur Lösung des Problems wird die Python Bibliothek `scipy` verwendet und unter Verwendung des Solvers „Sequential Least Squares Programming optimizer“ (SLSQP) siehe [121] eine Lösung des Problems mit einer Toleranz von 10^{-9} berechnet.

In Abbildung 4.6 ist die optimale Lösung des Problems aus Gleichung (4.51) für das 5-Busszenario dargestellt. Die farbliche Kodierung gibt Aufschluss über die Auslastung der Generatoren und Leitungen in Prozent an. Wird ein Generator an seiner unteren bzw. oberen Leistungsgrenze betrieben, siehe Tabelle A.1 bezüglich der Kapazitätsgrenzen im Anhang, ist dieser zu 0 bzw. zu 100 Prozent ausgelastet. Entsprechendes gilt für die Leitungen, wobei eine Auslastung von 0 % einer Leitung einem Leistungsfluss von 0 entspricht. Auch die Kapazitäten der Leitungen sind in der Tabelle A.1 zu finden.

Die Lasten befinden sich jeweils an den Bussen 2, 3 und 4. Es fällt auf, dass, obwohl Generator 5 die günstigsten Kostenfunktionskoeffizienten besitzt, dieser nicht voll ausgelastet ist. Grund hierfür ist, dass bereits ab einer Produktion von 240 MW beide vom Bus 5 ausgehenden Leitungen zu 100 % ausgelastet sind. Da Bus 5 keinen eigenen Verbrauch hat, ist dessen Leistungsbilanz somit ausgeglichen. Eine Mehrproduktion dieses Generators würde zu unnötigen Mehrkosten führen und wird daher durch die Optimierung vermieden.

Die Verbraucher am Bus 2, der keine eigenen Generatoren besitzt, werden zu etwas mehr als zwei Dritteln von den Generatoren 1, 2 und 5 versorgt, wobei die Leistung von Generator 5 über Bus 1 weitergereicht wird. Die kostengünstigen Generatoren des Bus 1 sind voll ausgelastet, was zum einen auf die niedrigen Kapazitäten der Generatoren sowie die zentrale Lage des Bus 1 und damit einer guten Versorgungsmöglichkeit der restlichen Busse zurückzuführen ist. Der Verbrauch am Bus 4, der mit 400 MW der Höchste im Netzwerk ist, wird nur zu etwa 22% durch den buseigenen Generator 4 gedeckt. Die restliche Leistung wird von den Generatoren 1, 2 und 5 der Busse 1 und 5 bezogen. Grund hierfür ist, dass Generator 4 die höchsten Kosten aufweist und daher nur zu 44,62 % ausgelastet ist. Der Verbrauch am Bus 3 wird durch den eigenen, kostengünstigen Generator gedeckt. Zusätzlich wird die Überproduktion sowie die erhaltene Leistung von Bus 4 an Bus 2 weitergereicht.

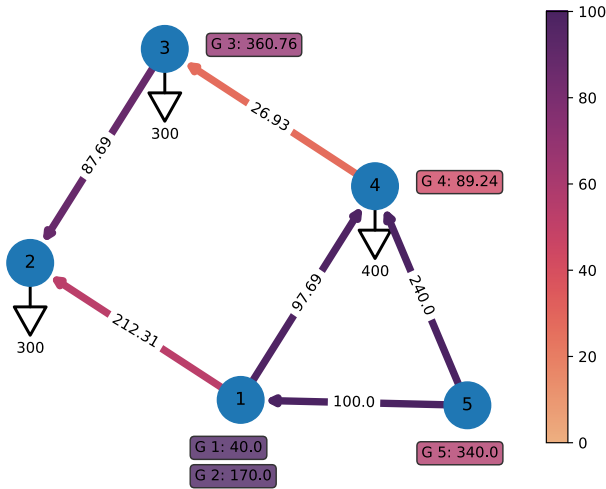


Abbildung 4.6: Ergebnisse der Optimierung für das 5-Busssystem. Die farbliche Kodierung stellt die Auslastung der Leitungen und Generatoren in Prozent dar. Die gerichteten Verbindungen zwischen den Agenten geben die Richtung des Leistungsflusses an, während die Bezifferung der Leitung die transferierte Leistung darstellt. Die fünf Generatoren des Systems sind mit G 1 - G 5 bezeichnet. Alle angegebenen Leistungen sind auf die Einheit MW bezogen.

Im 9-Busnetzwerk sind die Busse mit Ausnahme der Busse 1, 2 und 3 kreisförmig durch Leitungen miteinander verbunden, vergleiche Abbildung 4.7. Die Verbraucher befinden sich in diesem Kreis bei den Bussen 4, 6 und 8. Die Leistungsproduktion durch Generator 2 ist höher, da dieser eine geringere Kostenfunktion aufweist. Daher wird auch die Last an Bus 4 zusätzlich von Generator 2 gespeist, obwohl hierfür über drei Leitungen Leistung übertragen werden muss, während Generator 1 über eine einzige Leitung mit Bus 4 verbunden ist. Die übertragene Leistung ist allerdings mit 6,06 MW gering. Da Bus 6 keine eigenen Generatoren besitzt, muss dessen Verbrauch durch die Generatoren über die Buskette 1-4-5-6 bzw. 2-8-7-6 versorgt werden. Aufgrund der höheren Produktion von Generator 2 fließt mehr Leistung über die Kette 2-8-7-6 als über 1-4-5-6. In diesem Szenario wird keine Leitung und kein Generator an der

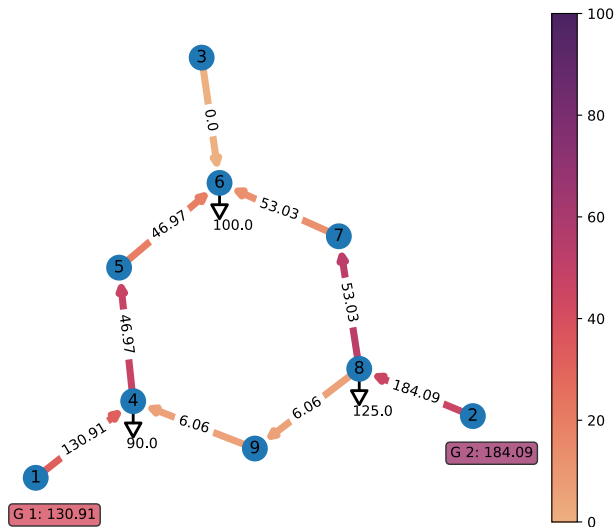


Abbildung 4.7: Ergebnisse der Optimierung für das 9-Busssystem. Die farbliche Kodierung stellt die Auslastung der Leitungen und Generatoren in Prozent dar. Die gerichteten Verbindungen zwischen den Agenten geben die Richtung des Leistungsflusses an. Alle angegebenen Leistungen sind auf die Einheit MW bezogen.

Kapazitätsgrenze betrieben. Die höchsten Auslastungen ergeben sich für Generator 2 mit etwa 68 % und der Leitung 7-8 mit 53 %.

Da eine Darstellung des 30-Busnetzwerkes nach Vorbild von Abbildung 4.6 und 4.7 zu unübersichtlich ist, wird auf eine Diskussion der Optimierungsergebnisse für dieses Szenario verzichtet.

4.7.4 Konvergenzverhalten und -geschwindigkeit der Algorithmen 1 - 3

Aufgrund der Übersichtlichkeit des 5-Busszenarios wird das Konvergenzverhalten der Algorithmen in Anwendung auf dieses Szenario genauer be-

trachtet. Ein Vergleich der Konvergenz der relativen Fehler erfolgt zwischen allen drei vorgestellten Busnetzwerken. Die verwendeten statischen Kommunikationsgraphen sind in Anhang A.11 dargestellt.

Lösung des 5-Busnetzwerk durch Algorithmus 1

Der Fokus dieses Abschnittes ist das Konvergenzverhalten der Algorithmen sowie die Konvergenzgeschwindigkeit der relativen Fehler. Bezüglich letzter werden Vergleiche zu dem 9-Bus und 30-Busnetzwerk gezogen, um die Skalierungsfähigkeit der Algorithmen abschätzen zu können.

Initialisierung Die Startwerte der Schätzungen werden für alle Agenten mit 0 initialisiert, sodass $x_i(0) = 0$ für alle $i = \{1, \dots, 5\}$. Wie in Algorithmus 1 gefordert, wird die GewichtungsvARIABLE $y_i(k)$ des Push-Sum-Konsensusprotokolls mit dem Skalar 1 initialisiert. Der Kommunikationsgraph wird entsprechend der Annahme 4.4.2 spalten-stochastisch gewichtet.

Für jede Nebenbedingungsungleichung des jeweiligen Busses bzw. Agenten wird eine Straffunktion nach Gleichung (4.7) angesetzt, welche einen beschränkten Gradienten besitzt, siehe auch Abbildung 2.3. Die Straffunktion für die Balanceungleichung aus (4.54) des Busses i wird mit Ψ_i^{Bilanz} bezeichnet, die Straffunktionen für die Leitungen mit $\Psi_i^{\text{Leitungen}}$ und die Straffunktion für die jeweiligen unteren bzw. oberen Kapazitäten der Generatoren mit $\Psi_i^{\underline{g}_i}$ bzw. $\Psi_i^{\bar{g}_i}$.

Damit die quadratischen Kostenfunktionen des Problems (4.51) die Annahme 4.4.3 erfüllen und damit beschränkte Gradienten besitzen, können die Kostenfunktionen außerhalb der Kapazitätsbeschränkungen der Leitungen und Generatoren linear fortgeführt werden, sodass beispielsweise für die Kostenfunktion eines Generators am Bus i

$$f_i^G(g_i) = \begin{cases} a_{G,i}g_i^2 + b_{G,i}g_i + c_{G,i}, & \text{für } g_i \in [\underline{g}_i, \bar{g}_i], \\ (2a_{G,i}\underline{g}_i + b_{G,i})g_i + c_{G,i}, & \text{für } g_i < \underline{g}_i, \\ (2a_{G,i}\bar{g}_i + b_{G,i})g_i + c_{G,i}, & \text{für } g_i > \bar{g}_i, \end{cases} \quad (4.57)$$

gilt.

Eine Parametersuche ergibt die optimierte Parametrisierung

$$\alpha_k = \frac{0,035}{k^{0,8}} \text{ und } r_k = 25 \cdot k^{0,5}$$

der Schrittweiten- und Strafparameterfolge.

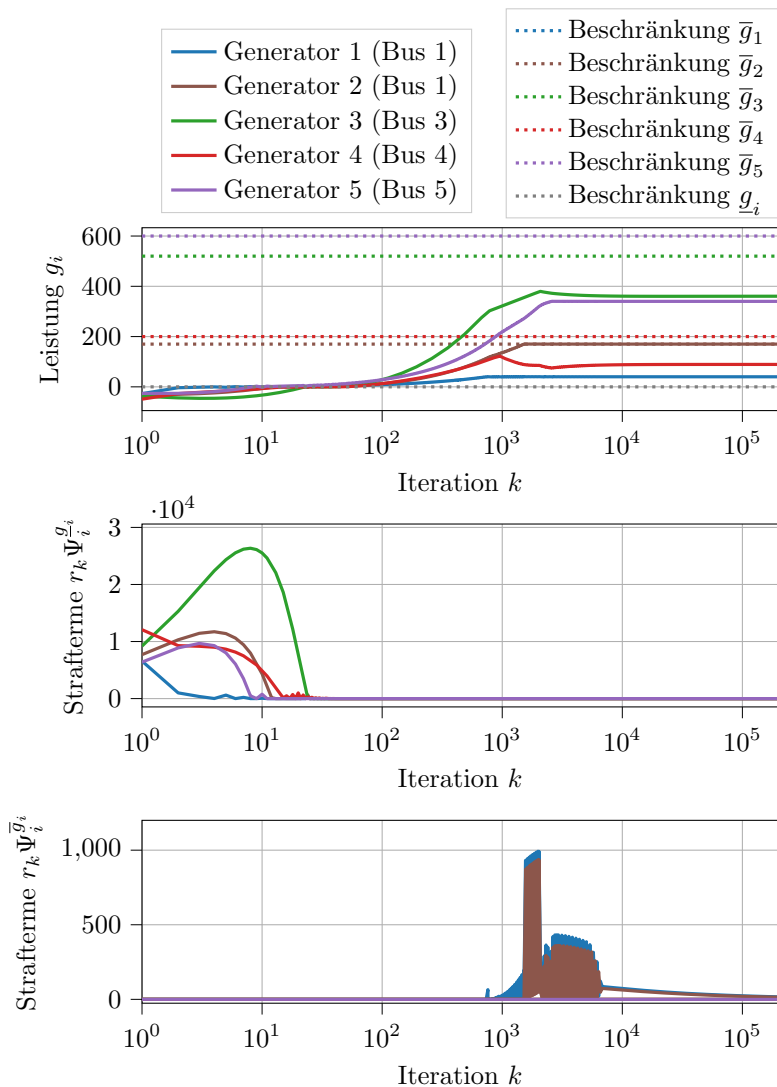


Abbildung 4.8: Verlauf der zu produzierenden Leistungen durch die Generatoren zusammen mit den Straffunktionen für die unteren und oberen Kapazitätsgrenzen der Generatoren für das 5-Busnetzwerk unter Algorithmus 1.

Verlauf der Leistungsproduktion In der Abbildung 4.8 ist der Verlauf der Leistungsproduktion der Generatoren über die Iterationen k aufgetragen. Die Abszisse ist hierbei logarithmisch skaliert. Die Kapazitäten der Generatoren sind gestrichelt eingezeichnet. Da alle Generatoren sich ein unteres Kapazitätslimit von 0 teilen, ist diese Grenze in grau für alle Generatoren zu sehen. Initialisiert wird die Leistungsproduktion bei jedem Generator i mit $g_i(0) = 0$. Da dies exakt der unteren Kapazitätsgrenze entspricht, sind zu Beginn auch alle Straffunktionen gleich null.

Im ersten Schritt verschiebt der Gradient der Generatorkostenfunktion die Leistungsproduktion ins Negative, wodurch die untere Kapazitätsgrenze verletzt wird. In Folge dessen steigen die Werte der Straffunktionen $\Psi_i^{g_i}$, multipliziert mit dem Strafparameter r_k , an, bis durch den Druck der Strafterme die Leistungsproduktion wieder in den erlaubten Bereich geschoben wird. Ab dem Nulldurchgang steigt die Leistungsproduktion kontinuierlich an, hervorgerufen durch den Anstieg der Strafterme der Leistungsbilanz Ψ_i^{Bilanz} der Busse, vergleiche hierzu die Abbildung 4.10. Dies geschieht zum einen bis die Generatoren 1 und 2 ihre oberen Kapazitätsgrenzen erreichen und in Folge dessen die entsprechenden Strafterme $\Psi_i^{\bar{g}_i}$ ansteigen und zum anderen bis die Bilanzgleichung erfüllt ist, siehe Abbildung 4.10. Im Verlauf der Strafterme der oberen Kapazitätsgrenzen $\Psi_i^{\bar{g}_i}$ zeigen sich starke Vibrationen, die sich aufgrund der Schwingung der Leistungsproduktionen um die Kapazitätsgrenzen ergeben. Durch das kontinuierliche Absenken der Schrittweite, vergleiche Abbildung 4.11, geht diese Schwingung in einen kontinuierlichen Verlauf über, der sich asymptotisch auf null zubewegt. Somit nähern sich die Leistungsproduktion für beide Generatoren ebenso asymptotisch ihren oberen Kapazität an. Wie aus der Abbildung 4.8 erkannt werden kann, befinden sich die restlichen Produktionsmengen nicht an ihrer Kapazitätsgrenze und somit sind deren Strafterme gleich 0.

Verlauf der Leistungsflüsse In der Abbildung 4.9 sind die Schätzungen bezüglich der Leistungsflüsse zusammen mit ihrem jeweiligen Optimum abgebildet. Jeder Agent des Systems schätzt alle Leistungsmengen, die über die jeweiligen Leitungen fließen. Die verschiedenen Leitungen sind in der Abbildung durch unterschiedliche Farben gekennzeichnet, während die Schätzung eines Agenten durch eine Linie in der entsprechenden Farbe dargestellt ist. Aus dieser Graphik kann geschlossen werden, dass die Konsensdynamik deutlich schneller ist als die Optimierungsdynamik, da die Linien nahezu deckungsgleich verlaufen. Unter Verwendung von Algorithmus 2 ist das nicht der Fall, vergleiche Abbildung 4.14.

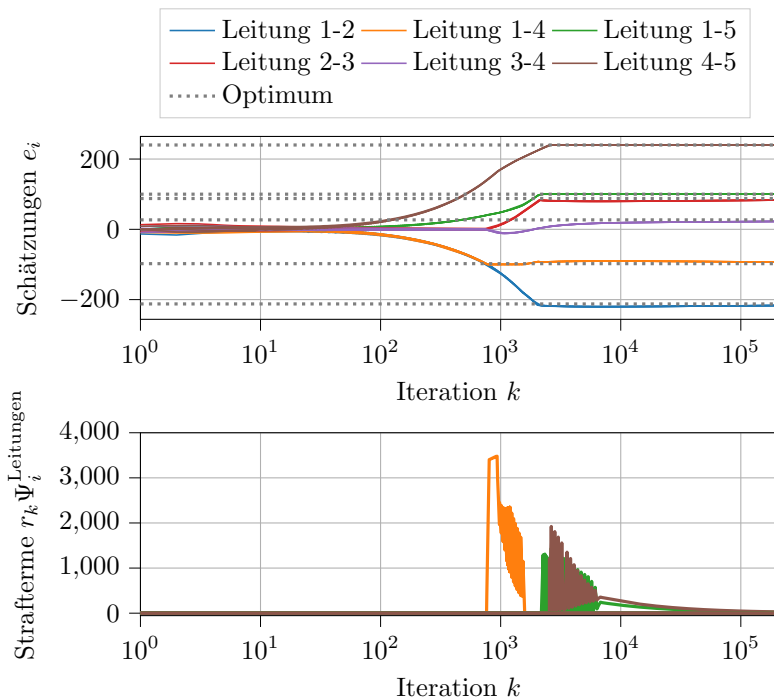


Abbildung 4.9: Verlauf der Schätzung $e_i = \text{col} \left((e_{kl})_{(k,l) \in \mathcal{E}_B} \right)$ bezüglich der Leitungen $k-l$ zusammen mit den Straftermen für die Leitungskapazitäten für das 5-Busnetzwerk unter Algorithmus 1.

Zu Beginn des Verfahrens sind alle Leistungsflüsse gleich 0 und somit werden die Leitungskapazitäten nicht verletzt. Im Optimum sind die Leitungen 1-5 und 4-5 zu 100 % ausgelastet. Im Verlauf des Optimierungsprozesses werden diese Kapazitätsgrenzen kurzzeitig überschritten, da aufgrund der günstigen Kostenfunktion des Generators 5 die Optimierung versucht, die Produktion dieses Generators weiter zu erhöhen. Die bei Verletzung der Kapazitätsgrenze ansteigenden Straffunktionen schieben den Wert des Leistungsflusses wieder in den zugelassenen Bereich.

Verlauf der Leistungsbilanz In Abbildung 4.10 ist der Verlauf der Leistungsbilanzen an jedem der fünf Busse dargestellt. Da alle Werte der Optimierung mit null initialisiert werden, wird die Bilanzgleichung der

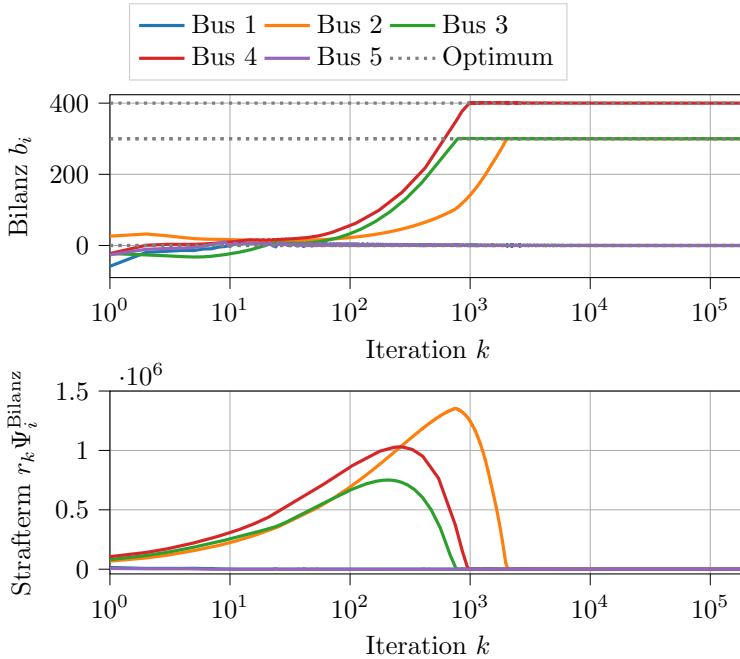


Abbildung 4.10: Die Bilanz $b_i = g_i - \sum_{j \in \mathcal{N}_i^-} e_{ij} + \sum_{j \in \mathcal{N}_i^+} e_{ji}$ des 5-Busnetzwerks unter Algorithmus 1 am jeweiligen Bus i bestehend aus der Summe der produzierten sowie zufließender und abfließender Leistung.

Busse 2, 3 und 4 von Beginn an nicht eingehalten, da an diesen Bussen Verbraucher angeschlossen sind. Die Strafterme steigen gemeinsam mit dem zunehmenden Straftermparameter, vergleiche Abbildung 4.11, an, bis die Bilanzen ausgeglichen sind. Obwohl die Bilanzen nur von unten und nicht von oben durch eine Ungleichung beschränkt sind, wird die Gleichung (4.51b) aus der Problemstellung eingehalten, da die Optimierung die unnötigen Kosten einer Überproduktion vermeidet. Im Vergleich zu den Bilanzverstößen der Busse 2, 3 und 4 fallen die Bilanzverstöße der anderen Busse nicht ins Gewicht.

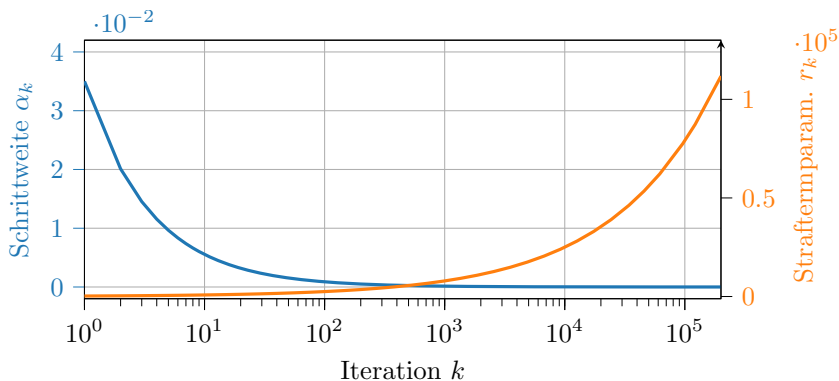


Abbildung 4.11: Verlauf der Schrittweitenfolge α_k und der Strafparameterfolge r_k unter Parameterwahl $\alpha_k = \frac{0,035}{k^{0,8}}$ und $r_k = 25 \cdot k^{0,5}$.

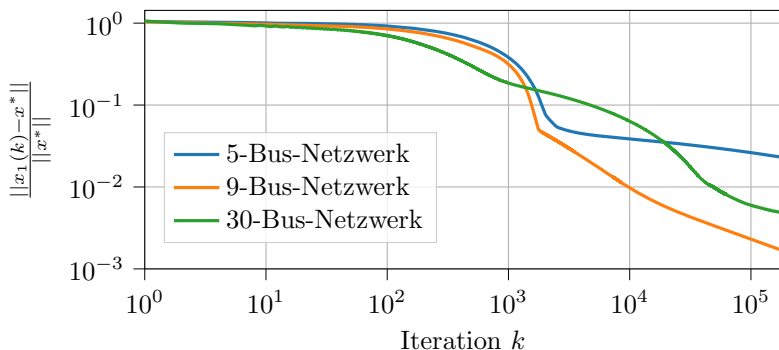


Abbildung 4.12: Verlauf der relativen Fehler unter Algorithmus 1 für 5-Bus-, 9-Bus- und 30-Busnetzwerke.

Konvergenz der relativen Fehler für 5-Bus-, 9-Bus- und 30-Busnetzwerk unter Algorithmus 1

In Abbildung 4.12 ist der Konvergenzverlauf der relativen Fehler für alle vorgestellten Busnetzwerke dargestellt. Zur Berechnung des relativen Fehlers wird die Schätzung des Agenten $i = 1$ zum Zeitpunkt k ausgewertet, d. h. $\epsilon_{\text{rel}} = \frac{\|x_1(k) - x^*\|}{\|x^*\|}$, wobei x^* das im Vorfeld zentral berechnete Optimum des Optimierungsproblems darstellt. In der Abbildung kann erkannt

werden, dass der relative Fehler für das 9-Bussystem am schnellsten absinkt, während der Fehler für das 5-Bussystem sogar langsamer konvergiert als der Fehler des 30-Bus-Verfahrens. Dies scheint zunächst kontraintuitiv zu sein, da bei größeren Netzwerken eine größere Anzahl von Variablen geschätzt und eine größere Anzahl von Bilanzen gedeckt werden müssen.

Bei Vergleich der Optimierungsergebnisse des 5-Busnetzwerkes mit den Resultaten des 9-Busnetzwerkes aus den Abbildungen 4.6 und 4.7 fällt auf, dass sich im Optimum des 5-Busnetzwerkes zwei Leitungen und ein Generator an ihrem Kapazitätslimit befinden, während sich im 9-Bussystem keine Leitung und kein Generator in Sättigung befinden. Entsprechend wird der verteilte Gradientenabstieg in diesen Fällen weniger stark durch die Strafterme gestört, die erst nach ausreichendem Anstieg der Strafparameter einen Einfluss auf den Optimierungsverlauf nehmen können. Durch die Aktivität einer größeren Anzahl von Ungleichungsbeschränkungen wird somit beim 5-Bussystem die Konvergenzzeit verlängert.

Nach $k = 2 \cdot 10^5$ Iterationen beträgt der relative Fehler beim 5-Bussystem $\epsilon_{\text{rel}} = 2,2 \cdot 10^{-2}$, beim 9-Bussystem $\epsilon_{\text{rel}} = 1,6 \cdot 10^{-3}$ und beim 30-Bussystem $4,7 \cdot 10^{-3}$.

Lösung des 5-Busnetzwerkes durch Algorithmus 2

Auch Algorithmus 2 wird zunächst in Anwendung auf das 5-Busnetzwerk untersucht, wobei anschließend ein Vergleich der Konvergenzgeschwindigkeit der relativen Fehler mit dem 9-Bus- und 30-Busnetzwerk gezogen wird. Für die Durchführung der Simulationen werden die in Anhang A.11 abgebildeten Kommunikationsgraphen verwendet.

Initialisierung Die Startwerte aller Schätzungen werden mit 0 initialisiert, sodass $x_i(0) = 0$ für alle Busse bzw. Agenten $i = \{1, \dots, 5\}$ gilt. Die GewichtungsvARIABLE $y_i(k)$ des Push-Sum-Konsensusprotokolls wird dem Algorithmus 2 entsprechend mit dem skalaren Wert 1 initialisiert, der statische Kommunikationsgraph wird spalten-stochastisch gewichtet und erfüllt die Annahme 4.5.3.

Im Gegensatz zum Algorithmus 1, bei dem alle Nebenbedingungen als lokal interpretiert werden, wird bei Algorithmus 2 zwischen globalen und lokalen Nebenbedingungen unterschieden, wobei es sich bei letzteren um Nebenbedingungen handelt, die ausschließlich einzelnen Agenten bekannt sind. In Bezug auf das zu lösende Optimierungsproblem aus Gleichung (4.51) werden die Leistungsbalance und die Generatorkapazitäten

als lokale Nebenbedingungen und die Leitungskapazitäten als globale Nebenbedingungen interpretiert. Hintergrund hierfür ist, dass der Verbrauch der Busse und deren Generatorlimits privat bleiben sollen, während die Leitungslimits den Bussen von dem Übertragungsnetzbetreiber mitgeteilt werden und somit keiner Informationseinschränkung unterliegen. Für jede lokale Nebenbedingungsungleichung wird ein Lagrangemultiplikator $\mu_i(k)$ mit dem Wert $\mu_i(0) = 0$ initialisiert und somit die lokalen Lagrangefunktionen nach Gleichung (4.21) erstellt.

Die Schrittweitenfolge α_k wird wie im Fall des Algorithmus 1 durch eine Rastersuche so gewählt, dass sich eine möglichst schnelle Konvergenz einstellt. Es resultiert die Folge

$$\alpha_k = \frac{0,045}{k^{0,5}}$$

für das 5-Bussystem.

Verlauf der Leistungsproduktion In der Abbildung 4.13 ist der Optimierungsverlauf für die Generatoren über die Iterationen k aufgetragen. Zusätzlich sind in der Graphik die Verläufe der Lagrangemultiplikatoren für die obere Kapazitätsgrenze abgebildet. Die oberen Grenzen selbst sind durch gepunktete Linien dargestellt.

Zu Beginn der Optimierung wird die Leistungsproduktion durch den Gradienten der Kostenfunktionen unter den Startwert von 0 gedrückt. Durch den ansteigenden Einfluss der Multiplikatoren $\mu_i^{\text{Gen.}}$ der unteren Kapazitätsgrenzen der Generatoren und μ_i^{Bilanz} , die für eine Einhaltung der Bilanzungleichung verwendet werden, siehe Abbildung 4.15, steigen die Leistungsproduktionen im ersten Abschnitt des Verlaufes an, bis die oberen Kapazitätsgrenzen für die Generatoren 1 und 2 erreicht sind. Ab diesem Punkt steigen die Werte der Lagrangemultiplikatoren $\bar{\mu}_i^{\text{Gen.}}$ für $i = 1, 2$ an, bis sich der dual optimale Wert $\bar{\mu}_i^{\text{Gen.,*}}$ einstellt und die jeweiligen Kapazitätsgrenzen für die beiden Generatoren eingehalten werden.

Verlauf der Leistungsflüsse Die Veränderung der Schätzung der Leistungsflüsse über den Optimierungsverlauf sind in der Abbildung 4.14 dargestellt. Die jeweiligen Farben kennzeichnen den Leistungsfluss auf einer Übertragungsleitung, während jede Linie in der gleichen Farbe die Schätzung eines Agenten bezüglich dieses Leistungsflusses repräsentiert. Da es sich bei den Kapazitätsgrenzen der Leitungen um globale Nebenbedingungen handelt, werden die jeweiligen Gradientenaktualisierungen auf den erlaubten Bereich projiziert, sodass die Leitungskapazitäten in jedem Schritt

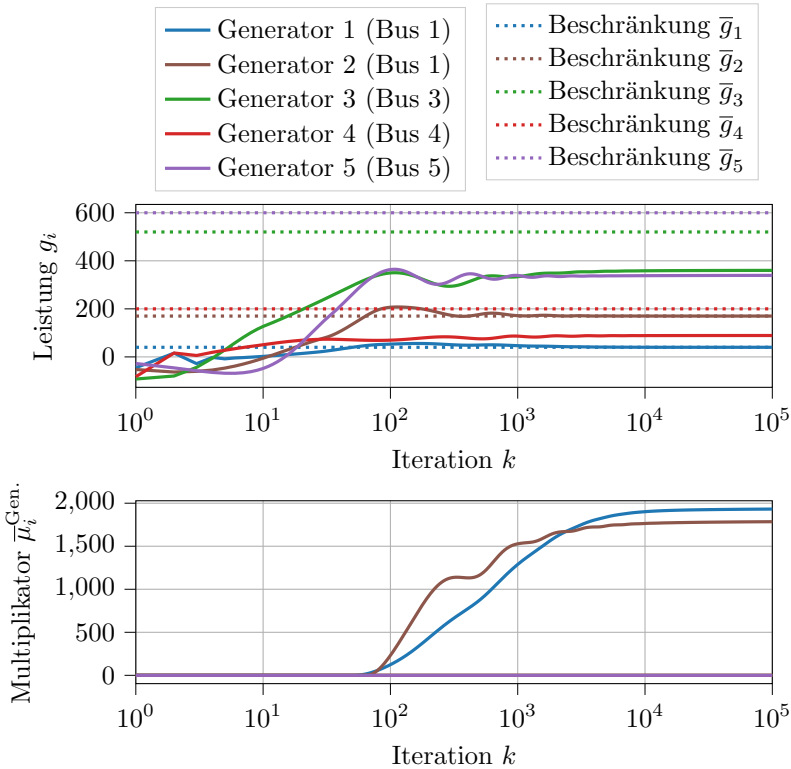


Abbildung 4.13: Verlauf der zu produzierenden Leistungen durch die Generatoren zusammen mit den Lagrangemultiplikatoren für die oberen Kapazitätsgrenzen der Generatoren für das 5-Busnetzwerk unter Algorithmus 2.

eingehalten werden. Dies kann in der Abbildung anhand der Leitungen 1-5 und 4-5 überprüft werden, für die der optimale Leistungsfluss genau auf der Leitungskapazität liegt. Besonders deutlich ist das scharfe Abknicken der Leitung 4-5 zu erkennen, wenn die Kurve erstmals den Wert 240 erreicht. Zusätzlich kann in der Abbildung die Konsensusbildung bezüglich der Schätzungen verfolgt werden. Der schwingende Verlauf ist durch die ebenfalls schwingenden Multiplikatorverläufe für die Einhaltung der Balanceungleichung aus Abbildung 4.15 zu erklären.

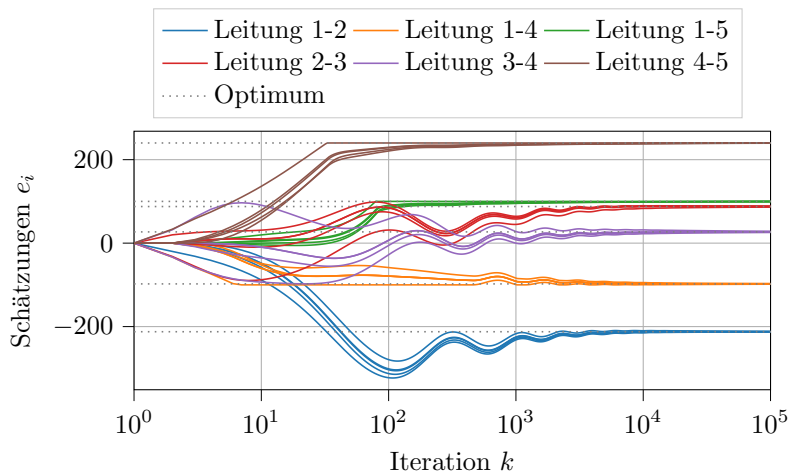


Abbildung 4.14: Verlauf der Schätzungen $e_i = \text{col} \left((e_{kl})_{(k,l) \in \mathcal{E}^B} \right)$ bezüglich der Leitungen k - l für das 5-Busnetzwerk unter Algorithmus 2. Die Gradientenschritte werden auf die durch die Kapazitätsgrenzen erzeugte Nebenbedingungsmenge projiziert.

Verlauf der Leistungsbilanz Der Verlauf der Leistungsbilanz an den jeweiligen Bussen findet sich in Abbildung 4.15. Zu Beginn sind die Bilanzgleichungen für die Busse 2,3 und 4 verletzt, weswegen der Einfluss der Multiplikatoren μ_i^{Bilanz} für $i = 2, 3, 4$ zu Beginn stark ansteigt. Für alle drei Kurven kommt es zu einem Überschwingen über den Verbrauch an den jeweiligen Bussen und es folgt ein Einschwingen sowohl der Multiplikatoren als auch der Leistungsverläufe um den Verbrauchswert. Diese Schwingungen übertragen sich, wie bereits festgestellt, auch auf die Verläufe der Leistungsflüsse aus Abbildung 4.14 und der Leistungsproduktion der Generator, siehe Abbildung 4.13. Nach dem Einschwingen bleiben die Werte der Lagrangemultiplikatoren konstant auf ihrem jeweiligen dual optimalen Wert.

Konvergenz der relativen Fehler für 5-Bus, 9-Bus und 30-Busnetzwerk unter Algorithmus 2

Die Abbildung 4.16 stellt die Konvergenzen der relativen Fehler unter Algorithmus 2 in Anwendung auf die Busnetzwerke dar. Um vergleichbare und möglichst optimale Konvergenzverläufe zu erhalten, werden im Vor-

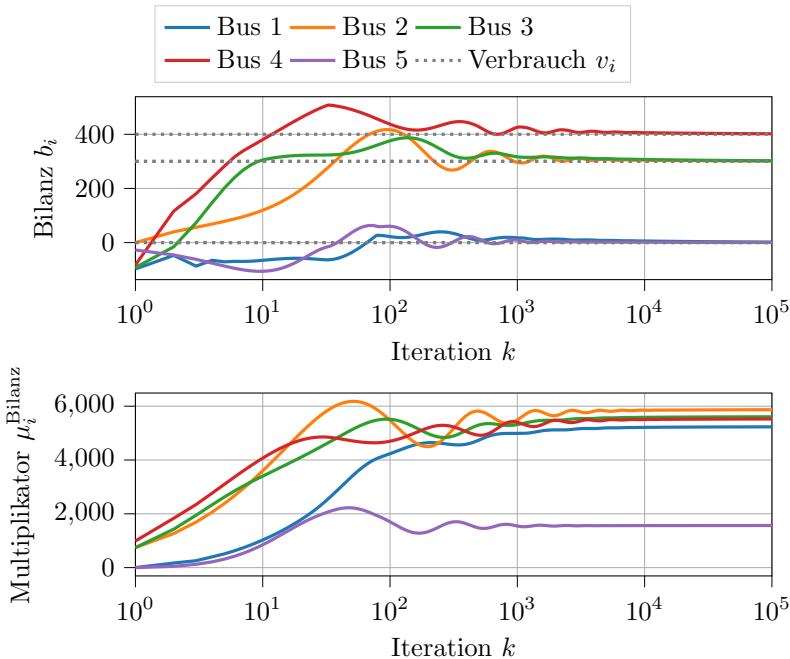


Abbildung 4.15: Die Bilanz $b_i = g_i - \sum_{j \in \mathcal{N}_i^-} e_{ij} + \sum_{j \in \mathcal{N}_i^+} e_{ji}$ des 5-Busnetzwerks unter Algorithmus 2 am jeweiligen Bus i bestehend aus der Summe der produzierten sowie zufließender und abfließender Leistung.

feld Rastersuchen zur Parametrisierung der Schrittweitenfolgen α_k durchgeführt. Die relativen Fehler werden in Bezug auf die Schätzungen des ersten Agenten durchgeführt, sodass $\epsilon_{\text{rel}} = \frac{\|x_1(k) - x^*\|}{\|x^*\|}$.

Es kann erkannt werden, dass die Fehler für das 5-Busssystem am schnellsten konvergieren, gefolgt von den Fehlern für das 9-Busnetzwerk. Das 30-Busnetzwerk weist die langsamste Konvergenz auf. Diese Beobachtung folgt der intuitiven Annahme, dass größere Optimierungsprobleme mit mehr Variablen, Kostenfunktionen und Nebenbedingungen langsamer konvergieren als kleinere. Wie bereits besprochen kann diese Intuition unter Verwendung des Algorithmus 1 nicht festgestellt werden, da für diesen das Optimierungsproblem des 5-Busnetzwerkes langsamer konvergiert als dasjenige des 9-Bussystems. Dieses Phänomen wird durch die höhere Aktivität der Nebenbedingungen bezüglich der Leitungs- und Generator-

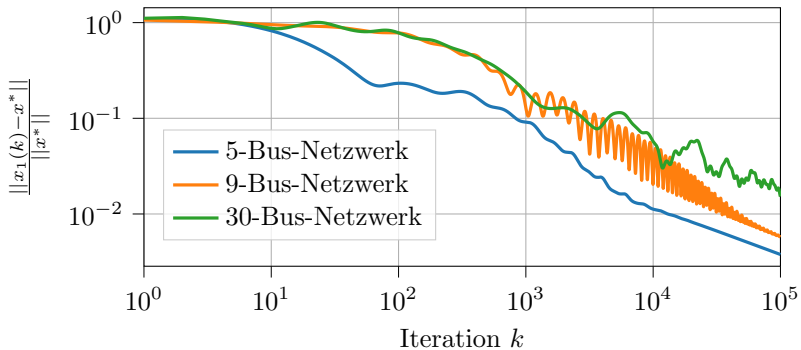


Abbildung 4.16: Verlauf der relativen Fehler unter Algorithmus 2 für 5-Bus-, 9-Bus- und 30-Busnetzwerke.

kapazitäten erklärt. Im Fall des Algorithmus 2 werden die Leitungskapazitäten unter Verwendung einer Projektion in das Optimierungsverfahren eingebunden. Dadurch wird eine höhere Komplexität, die sich durch die Aktivität der Nebenbedingung bei Algorithmus 1 ergibt, vermieden.

Die Schwingungen, die durch den Verlauf der Lagrangemultiplikatoren bei aktiven Nebenbedingungen entstehen, siehe beispielsweise Abbildung 4.15, übertragen sich auf den Verlauf des relativen Fehlers. Besonders deutlich werden diese im Verlauf des relativen Fehlers für das 9-Busnetzwerk.

Nach $k = 10^5$ Iterationen beträgt der relative Fehler für das 5-Busnetzwerk $\epsilon_{\text{rel}} = 3,77 \cdot 10^{-3}$, für das 9-Busnetzwerk $\epsilon_{\text{rel}} = 5,74 \cdot 10^{-3}$ und für das 30-Busnetzwerk $15,24 \cdot 10^{-3}$.

Lösung des 5-Busnetzwerks durch Algorithmus 3

Auch für den Algorithmus 3 werden die Ergebnisse für das 5-Busnetzwerk im Detail analysiert, während für die 9- und 30-Busnetzwerke ausschließlich die Konvergenz des relativen Fehlers betrachtet wird.

Initialisierung Ebenso wie für die Algorithmen 1 und 2 werden die Startwerte der Schätzungen der Agenten mit dem Nullvektor initialisiert, sodass $x_i(0) = 0$ für alle $i = \{1, \dots, 5\}$ gilt. Wie in Algorithmus 3 gefordert, werden die Startwerte der Schätzungen bezüglich des Gradienten des Systems mit den lokalen Gradienten initialisiert, ausgewertet an den jewei-

ligen Initialwerten der Schätzung. Entsprechend gilt $y_i(0) = \nabla f_i(x_i(0))$. Der Annahme 4.6.2 folgend, wird der Kommunikationsgraph statisch und doppelt-stochastisch gewichtet initialisiert.

Der Gradientenschritt wird auf die Beschränkungsmenge des Problems (4.51), bestehend aus den Kapazitätsbeschränkungen der Übertragungsleitungen und der Generatoren sowie der Balancenebenbedingung aus (4.54), projiziert, weswegen keine weiteren Initialisierungen für die Nebenbedingungen durchgeführt werden müssen.

Eine optimierte Wahl der Schrittweite zur schnellen Lösung des 5-Busproblems ist $\alpha = 5,95$.

Verlauf der Leistungsproduktion In der Abbildung 4.17 ist der Konvergenzverlauf der zu produzierenden Leistungen durch die Generatoren des 5-Busnetzwerkes unter Algorithmus 3 dargestellt. Es kann erkannt werden, dass die Beschränkungen in jedem Iterationsschritt eingehalten werden, da die Gradientenschritte in jeder Iteration auf die Beschränkungsmenge projiziert werden. Unter der verwendeten Schrittweite zeigt sich ein schwingendes Verhalten für den Leistungsverlauf zwischen einzelnen Iterationsschritten, besonders deutlich für Generator 4 und abgeschwächt auch für Generator 1.

Verlauf der Leistungsflüsse Der Verlauf der Leistungsflüsse für Algorithmus 3 ist in Abbildung 4.18 abgebildet. In gleichen Farben sind die Kurven für die Schätzungen bezüglich des Leistungsflusses auf einer Leitung dargestellt. In der gleichen Farbe sind zusätzlich die Schätzungen der Agenten des Gradienten des Gesamtsystems bezüglich der jeweiligen Variable des Leistungsflusses in der unteren Grafik abgebildet. Es kann erkannt werden, dass sich sowohl für die Fluss-schätzungen als auch für die Gradientenschätzungen schnell ein Konsensus einstellt. Auch in diesem Fall werden die Beschränkungen durch die Verwendung der Projektion in jedem Iterationsschritt eingehalten.

Konvergenz der relativen Fehler für 5-Bus-, 9-Bus- und 30-Busnetzwerk unter Algorithmus 3

In der Abbildung 4.19 sind schließlich die Konvergenzverläufe des relativen Fehlers $\epsilon_{\text{rel}} = \frac{\|x_1(k) - x^*\|}{\|x^*\|}$ für die drei Busnetzwerke dargestellt. Die Schrittweiten für die Algorithmen sind optimiert, sodass die Konvergenzverläufe vergleichbar sind. Es kann erkannt werden, dass ähnlich wie im Fall des

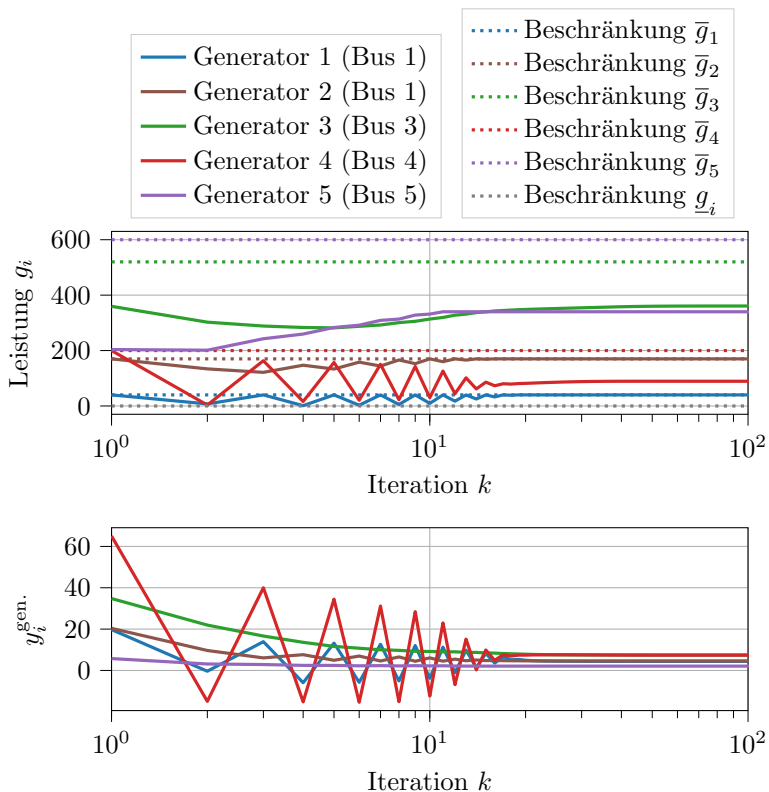


Abbildung 4.17: Verlauf der zu produzierenden Leistungen durch die Generatoren des 5-Busnetzwerkes zusammen mit der Schätzung der Gradienten bezüglich der Leistungsvariablen unter Algorithmus 3.

Algorithmus 1 der relative Fehler für das 9-Busnetzwerk am schnellsten konvergiert und ein Fehler von $\epsilon_{\text{rel}} = 10^{-5}$ bereits nach etwa $k = 120$ Iterationen unterschritten wird. Die schnellere Konvergenz des 9-Busnetzwerkes gegenüber dem kleineren 5-Busnetzwerk kann mutmaßlich durch die geringere Anzahl der aktiven Beschränkungen im Optimum erklärt werden. Der relative Fehler für das 5-Busnetzwerk unterschreitet die Grenze von 10^{-5} nach etwa $k = 360$ Iterationen, während das 30-Busnetzwerk als das größte Netzwerk einen relativen Fehler von $\epsilon_{\text{rel}} = 2,06 \cdot 10^{-3}$ nach $k = 500$ Iterationen aufweist.

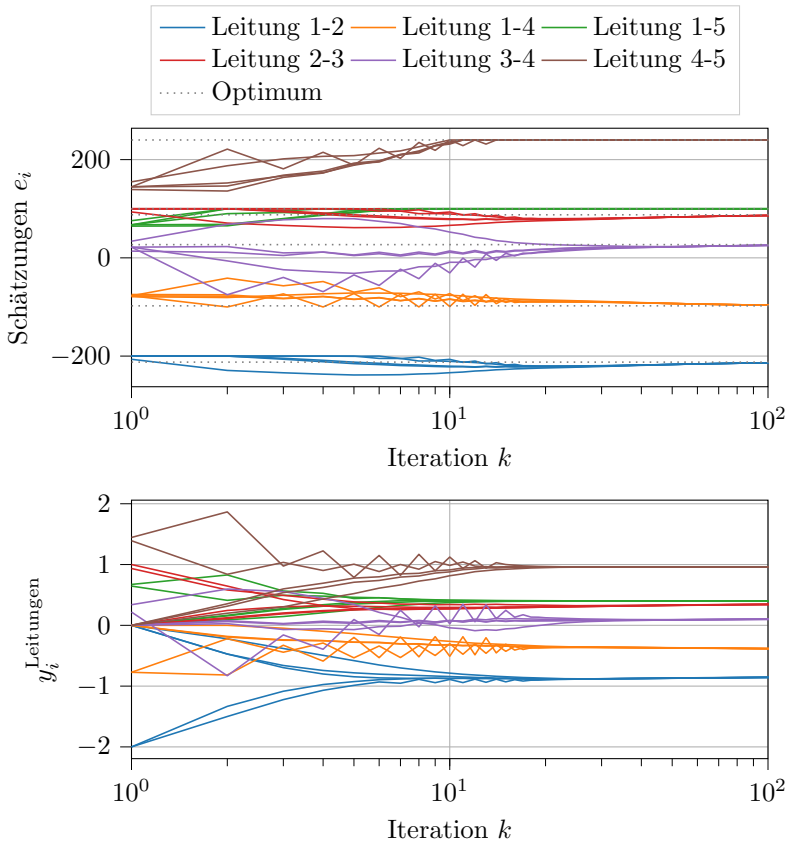


Abbildung 4.18: Verlauf der Schätzungen $e_i = \text{col}\left((e_{kl})_{(k,l) \in \bar{\mathcal{E}}^B}\right)$ bezüglich der Leitungen k - l zusammen mit der Schätzung der Gradienten bezüglich der Übertragungsvariablen für das 5-Busnetzwerk unter Algorithmus 3. Die Gradientenschritte werden auf die durch die Kapazitätsgrenzen erzeugte Nebenbedingungs Menge projiziert.

Bemerkenswert ist die um mehrere Faktoren geringere Konvergenzzeit des Algorithmus 3 im Vergleich zu den Algorithmen 1 und 2. Dieses Phänomen lässt sich zum einen auf die Verwendung von konstanten Schrittweiten und in diesem Zusammenhang der Schätzung des Gradienten des Gesamtsystems zurückführen. Darüber hinaus wird durch die Projektion ein effizientes Verfahren zur Einbindung der Nebenbedingungen verwendet,

das keine Erweiterung der Kostenfunktion durch Strafterme oder Lagrange-multiplikatoren benötigt. Der Nachteil dieser Art der Berücksichtigung der Nebenbedingungen liegt darin, dass ausschließlich globale Nebenbedingungen berücksichtigt werden können und somit die Privatsphäre der Agenten eingeschränkt wird.

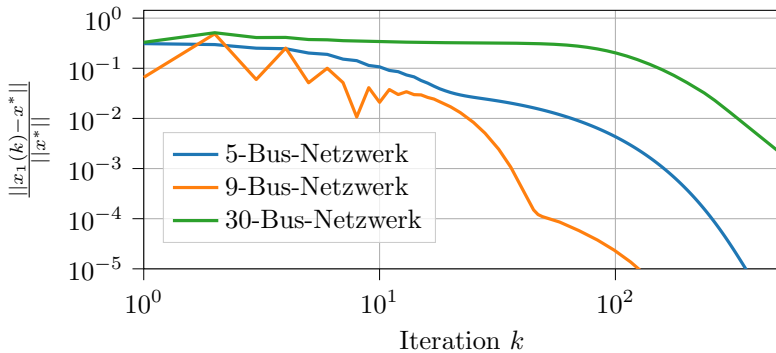


Abbildung 4.19: Verlauf der relativen Fehler unter Algorithmus 3 für 5-Bus-, 9-Bus- und 30-Busnetzwerke.

4.8 Fazit

Die Beiträge dieses Kapitels, siehe Abschnitt 4.3, bestehen aus der Vorstellung drei verschiedener Algorithmen zur Lösung von beschränkten, kooperativen und verteilten Optimierungsproblemen sowie deren Anwendung und Auswertung bezüglich eines Einsatzplanproblems. Eine Einordnung der Eigenschaften der vorgestellten Algorithmen in den Stand der Forschung erfolgt im Abschnitt 4.3 und ist in der Tabelle 4.1 sowie im Unterabschnitt 4.3.2 zusammengefasst.

Die Hauptbeiträge der Algorithmen der Beiträge 1 und 2 ist die Einbindung der lokalen Nebenbedingungen in Kombination mit einer verallgemeinerten Kommunikationsarchitektur: Bei beiden Algorithmen können lokale Nebenbedingungen berücksichtigt werden, sodass diese nur den jeweiligen Agenten bekannt sind und keinem weiteren Agenten des Systems. Für die Konvergenz reicht die Kommunikation über einen spalten-stochastisch gewichteten, zeitvarianten Kommunikationsgraphen aus. Diese Erweiterung wird durch die Kombination des Push-Sum-Konsensusprotokoll-basierten,

verteilten Gradientenabstiegsverfahren mit einem Straftermverfahren, siehe Algorithmus 1, bzw. mit einem dualen Ansatz, siehe Algorithmus 2, erreicht. Damit schließen die Algorithmen 1 und 2 die in Unterabschnitt 4.3.2 beschriebenen Forschungslücken bezüglich des ersten und zweiten Punktes, wie auch in den Beiträgen 1 und 2 beschrieben. Notwendig für die Konvergenz beider Verfahren ist unter anderem eine absinkende Schrittweitenfolge, welche die Länge des Gradientenschrittes steuert. Allerdings ist diese Verfahrensart für im Vergleich langsame Konvergenzzeiten bekannt.

Im Gegensatz zu den ersten beiden Algorithmen ist für den Konvergenzbeweis für Algorithmus 3 ein statischer, doppelt-stochastisch gewichteter Graph notwendig, der somit weniger allgemein ist. Da dieser Algorithmus eine Projektion des Gradientenschrittes auf die Nebenbedingungsmenge verwendet, müssen zusätzlich alle Nebenbedingungen global, d. h. allen Agenten, bekannt sein. Durch eine separierte Schätzung der Gradienten des Gesamtsystems sind konstante Schrittweiten für diesen Algorithmus zulässig. Im Vergleich zu einem ähnlichen Ansatz in der Arbeit [29] müssen weniger Algorithmusparameter optimal festgelegt werden. Damit erweitert der Algorithmus 3 den Stand der Forschung bezüglich der Lösung beschränkter, verteilter Optimierungsprobleme unter Verwendung konstanter Schrittweiten. Ein Vergleich der Simulationsergebnisse des Abschnittes 4.7 zeigt, dass unter Verwendung des Algorithmus 3 eine um etwa den Faktor 200 geringere Konvergenzzeit im Vergleich zu Algorithmus 2 erreicht wird.

Entsprechend muss bei der Entscheidung über die Wahl des Algorithmus in Bezug auf ein bestimmtes Problem abgewogen werden, ob eine allgemeinere Kommunikation und Wahrung der Privatsphäre so stark gewichtet ist, dass eine deutlich längere Konvergenzzeit in Kauf genommen werden kann.

5 Verteilte Lösung von Multiclusterspielen

5.1 Einleitung

Im vorherigen Kapitel werden Methoden vorgestellt, mit denen Optimierungsprobleme gelöst werden können, die auf ein Agentensystem verteilt sind. Dabei wird davon ausgegangen, dass die Agenten miteinander kooperieren, um ein gemeinsames Problem zu lösen. Es existieren allerdings auch viele Problemstellungen, in denen die Agenten egoistisch eine Aktion oder Strategie wählen, um die eigenen Kosten zu minimieren. Dazu gehören beispielsweise Energiemarktsituationen wie in [15] oder Energiemanagementprobleme wie in [16][17], bei denen Agenten bezüglich einer geteilten Ressource miteinander konkurrieren. Wird davon ausgegangen, dass die Kostenfunktionen der Agenten untereinander bezüglich einer gemeinsamen Optimierungsvariablen verkoppelt sind, hat die egoistische Wahl eines Agenten Einfluss auf die Kostenfunktionen der anderen Agenten. Diese Art von Problemen kann durch die nicht-kooperative Spieltheorie modelliert werden. Das Spiel gilt dann als gelöst, wenn sich ein stabiler Zustand zwischen den Spielern einstellt. Das Nash-Gleichgewicht beschreibt beispielsweise einen solchen Zustand. Für grundlegende Erläuterungen hierzu sei auf den Grundlagenabschnitt 2.4 verwiesen.

Die Identifikation des Nash-Gleichgewichts durch eine verteilte Methode, bei der die Agenten bzw. die konkurrierenden Spieler keine Information bezüglich ihrer Kostenfunktionen und Nebenbedingungen mit den Kontrahenten teilen müssen, ist bereits ein gut erforschtes Gebiet und es existieren effiziente, zeitdiskrete Algorithmen, bei deren Ausführung alle Agenten lokal Änderungen an ihren Aktionen vornehmen, bis sich das Nash-Gleichgewicht einstellt. Unterschieden wird hierbei unter anderem zwischen gradienten- [4][122], proximal-punkt- [123] und payoffbasierten [124][125] Verfahren zur Identifizierung von Nash-Gleichgewichtsstrategien sowie gradienten- [126], proximal-punkt- [5] und payoffbasierten [55] Verfahren für die Identifikation von Verallgemeinerten Nash-Gleichgewichten.

Multiclusterspiele stellen nun eine Kombination von kooperativer und nicht-kooperativer Optimierung in Agentennetzwerken dar, siehe hierzu auch die Abbildung 5.1. Dabei wird angenommen, dass die Agenten sogenannten Clustern zugeordnet sind. Innerhalb dieser Cluster arbeiten die Agenten zusammen, um ein kooperatives Optimum bezüglich der Gesamtkostenfunktion des Clusters, bestehend aus einer Kombination der Kostenfunktionen der Agenten des Clusters, zu erreichen. Mit den Agenten der anderen Cluster findet keine Kooperation statt. Aufgrund der angenommenen Kopplung der Clusterkostenfunktionen bezüglich der festzulegenden Optimierungsvariablen, hat die von den Agenten des jeweiligen Cluster bestimmte Clusterstrategie eine Auswirkung auf die Kosten der restlichen Cluster. Damit befinden sich die Cluster in einem nicht-kooperativen Spiel miteinander.

Die Cluster repräsentieren in diesem System somit Interessengemeinschaften, während die Agenten, die diesen Gemeinschaften zugehörig sind, ihre jeweiligen Aktionen im Sinne der optimalen Gesamtstrategie des Clusters ausführen. Das Multiclusterspiel gilt als gelöst, wenn die Cluster mit dem Nash-Gleichgewicht einen stabilen Zustand bezüglich der Wahl ihrer Strategien erreichen und sich gleichzeitig ein kooperatives Optimum bezüglich der Kombination der lokalen Kostenfunktionen innerhalb der Cluster einstellt.

Nachfolgend wird zunächst die Problemstellung der Multiclusterspiele mathematisch beschrieben, bevor im Abschnitt zum Stand der Forschung bereits veröffentlichte Algorithmen diskutiert werden. Anschließend wird gekennzeichnet, an welchen Stellen diese Forschungsfront durch dieses Kapitel erweitert wird. Der Beitrag 5 wird darauffolgend in Form eines Algorithmus detailliert vorgestellt. Im Simulationsabschnitt wird das Einsatzplanproblem zwischen Microgrids, beschrieben in Kapitel 3, unter Verwendung des zuvor beschriebenen Algorithmus gelöst. Dies stellt den Beitrag 6 dar.

5.2 Problemstellung

Es sei ein Agentensystem gegeben, das aus insgesamt N Agenten besteht, die in der Agentenmenge $\mathcal{I} = \{1, \dots, N\}$ enthalten sind. Diese Agenten sind aufgeteilt auf insgesamt H Cluster, die sich in der Clustermenge $\mathcal{H} = \{1, \dots, H\}$ befinden, wobei jeder Cluster h genau N^h Agenten be-

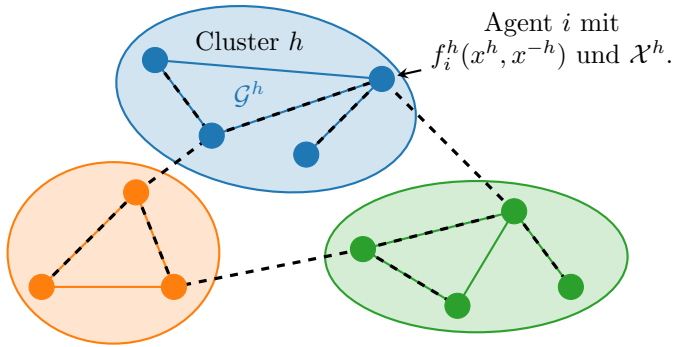


Abbildung 5.1: Multiclusternstruktur eines Agentensystems mit drei Clustern und elf Agenten. Die lokalen Kommunikationsgraphen \mathcal{G}^h verbinden die Agenten innerhalb der Cluster $h \in \{1, 2, 3\}$, während der globale Graph \mathcal{G} , hier schwarz gestrichelt dargestellt, die Agenten auch über die Cluster Grenzen hinaus verbindet. Der globale Graph kann dabei mit den lokalen Graphen überlappen. Beispielfähig wird hier von ungerichteten Kommunikationsverbindungen ausgegangen. Nur der Agent i des Clusters h kennt die lokale Kostenfunktion $f_i^h(x)$, während die Beschränkungsmenge \mathcal{X}^h allen Agenten des Clusters h bekannt ist.

inhaltet. Die Menge \mathcal{I}^h beschreibt die Menge der Agenten, die Cluster h zugeordnet sind. Es wird davon ausgegangen, dass kein Agent Teil von mehr als einem Cluster sein kann, sodass $\mathcal{I}^h \cap \mathcal{I}^l = \emptyset, \forall h, l \in \mathcal{H}, h \neq l$. Zusätzlich beinhaltet jeder Cluster mindestens einen Agenten, sodass $\mathcal{I}^h \neq \emptyset, \forall h \in \mathcal{H}$. In der Abbildung 5.1 ist beispielhaft ein so strukturiertes Agentensystem dargestellt.

Wie in der Einleitung zu diesem Kapitel beschrieben, koexistieren kooperatives und nicht-kooperatives Verhalten in diesem System: Während die Agenten innerhalb der jeweiligen Cluster zusammenarbeiten, um das Ziel des Clusters zu erreichen, befinden sich die Cluster selbst als virtuelle Spieler in einem nicht-kooperativen Spiel zueinander. Die Bezeichnung als virtuell bezieht sich auf die Tatsache, dass die Cluster selbst als Entität keine Entscheidungen treffen, sondern die Agenten, die den jeweiligen Clustern zugehörig sind, Aktionen im Sinne des Clusterziels durchführen. Folglich können alle Agenten des Clusters h die dem Cluster zugehörige Aktion $x^h \in \mathbb{R}^{d^h}$ anpassen, wobei d^h die Dimension der Aktion des Clusters h darstellt. Das Ziel des Clusters wird mathematisch als Minimierung der Funktion $F^h(x^h, x^{-h})$ ausgedrückt, welche sowohl von der eigenen Clusteraktion x^h als auch von der Aktion der restlichen Cluster

x^{-h} abhängt. Aufgrund der Kooperation der Agenten innerhalb der Cluster kann die Clusterkostenfunktion F^h als Mittelwert der lokalen Kostenfunktionen $f_i^h : \mathbb{R}^d \rightarrow \mathbb{R}$ mit $d = \sum_{h=1}^H d^h$ ausgedrückt werden²⁰. Somit gilt

$$F^h(x^h, x^{-h}) = \frac{1}{N^h} \sum_{i=1}^{N^h} f_i^h(x^h, x^{-h}). \quad (5.1)$$

Die Agenten jedes Clusters versuchen diese Funktion egoistisch zu minimieren, indem durch das optimale Festlegen der Clustervariablen x^h die beste Antwort auf die Aktion der Agenten der restlichen Cluster gefunden wird. Folglich legt jeder Cluster $h \in \mathcal{H}$ seine Aktion durch das Lösen des Optimierungsproblems

Problem 5 (*Multiclusterspiel*)

$$\begin{aligned} \min_{x^h} F^h(x^h, x^{-h}) &= \min_{x^h} \frac{1}{N^h} \sum_{i=1}^{N^h} f_i^h(x^h, x^{-h}), \\ \text{s.t. } x^h &\in \mathcal{X}^h, \end{aligned}$$

fest, bis sich ein stabiler Zustand bezüglich des gesamten Strategievektors des Spiels $x = \text{col}(x^h, x^{-h}) \in \mathbb{R}^d$ einstellt. Je nachdem, ob Beschränkungen für die Wahl der Clusteraktion existieren, gilt entweder $\mathcal{X}^h \subset \mathbb{R}^{d^h}$ oder $\mathcal{X}^h \equiv \mathbb{R}^{d^h}$.

Damit beschreibt die obige Problemstellung das nicht-kooperative Spiel $\Gamma(H, \mathcal{X}, \{F^h\}_{h \in \mathcal{H}})$ zwischen allen Clustern in \mathcal{H} , wobei unter Verwendung des kartesischen Produkts $\mathcal{X} = \mathcal{X}^1 \times \dots \times \mathcal{X}^H$ gilt. Dieses Spiel gilt als gelöst, wenn keiner der Spieler, d. h. der Cluster, einen Anreiz hat, von seiner aktuellen Strategie abzuweichen, bzw. jeder Spieler eine für sich optimale Antwort auf die Strategien der anderen gefunden hat. Wie in Abschnitt 2.4 beschrieben, wird ein solcher Zustand als Nash-Gleichgewicht des Spiels bezeichnet. In Bezug auf das Multiclusterproblem 5 ist der Vektor $x^{\text{NE}} \in \mathcal{X}$ ein Nash-Gleichgewicht des Spiels $\Gamma(H, \mathcal{X}, \{F^h\}_{h \in \mathcal{H}})$, wenn

$$F^h((x^h)^{\text{NE}}, (x^{-h})^{\text{NE}}) \leq F^h(x^h, (x^{-h})^{\text{NE}}), \forall x^h \in \mathcal{X}^h, \forall h \in \mathcal{H}, \quad (5.2)$$

gilt. In Unterabschnitt 5.2.2 wird diskutiert, wie sich die Lösung des kooperativen Problems innerhalb der Cluster in diese Formulierung

²⁰Hier wird der Mittelwert und nicht die Summe aus den lokalen Kostenfunktionen verwendet, um die Notation des Konvergenzbeweises zu vereinfachen. Durch den konstanten Faktor $\frac{1}{N^h}$ ändert sich zwar der Wert der Kostenfunktion im Vergleich zur Summe, der Lösungsvektor x^* selbst wird allerdings nicht beeinflusst.

integriert.

Das Mapping des Spiels, siehe auch Grundlagenabschnitt 2.4, ist eine Abbildung $M : \mathcal{X} \rightarrow \mathbb{R}^d$, die aus einer Vektorisierung der Gradienten der Spielerkostenfunktionen besteht, d. h.

$$M(x) = \text{col}((\nabla_h F^h(x))_{h \in \mathcal{H}}) = \text{col} \left(\left(\left(\frac{1}{N^h} \nabla_h \sum_{i=1}^{N^h} f_i^h(x) \right)_{h \in \mathcal{H}} \right) \right). \quad (5.3)$$

Anhand der Monotonieeigenschaften dieser Abbildung kann abgeleitet werden, ob ein Nash-Gleichgewicht existiert und, falls ja, ob das Nash-Gleichgewicht eindeutig ist. Damit spielt das Spielmapping eine entscheidende Rolle bei der Überprüfung der Lösungsbedingungen von Multiclusterspielen, die im nachfolgenden Unterabschnitt 5.2.2 diskutiert werden.

In Multiclusterproblemstellungen kennt der Agent i des Clusters h nur die eigene lokale Kostenfunktion f_i^h sowie im beschränkten Fall die Beschränkungsmenge \mathcal{X}^h des Clusters. Folglich müssen sich die Agenten innerhalb eines Clusters miteinander koordinieren, damit das tatsächliche Minimum des Mittelwertes der lokalen Kostenfunktionen erreicht werden kann. Zusätzlich benötigen die Agenten den aktuellen Gesamtstrategievektor $x \in \mathbb{R}^d$ des Spiels $\Gamma(H, \mathcal{X}, \{F^h\}_{h \in \mathcal{H}})$, um eine Antwort auf die Strategien der restlichen Cluster x^{-h} zu finden. Aus diesem Grund verwaltet jeder Agent eine Schätzung $x_i \in \mathbb{R}^d$ des Strategievektors x . Die Schätzung jedes Agenten $i \in \mathcal{I}^h$ in einem generischen Cluster h besteht aus der Schätzung bezüglich der Strategie $x_i^{(h)} \in \mathbb{R}^{d^h}$ des eigenen Clusters und der Schätzungen bezüglich der Strategien der anderen Cluster $x_i^{(-h)} \in \mathbb{R}^{d-d^h}$, sodass

$$x_i = \text{col} \left(x_i^{(1)}, \dots, x_i^{(h)}, \dots, x_i^{(H)} \right).$$

Die Schätzungen der einzelnen Agenten i können nun in dem Vektor $\mathbf{x} = \text{col}((x_i)_{i \in \mathcal{I}}) \in \mathbb{R}^{Nd}$ vektorisiert werden. Entsprechend wird der Vektor

$$\mathbf{x}^h = \text{col}((x_i)_{i \in \mathcal{I}^h}) \in \mathbb{R}^{N^h d},$$

in dem alle Schätzungen der Agenten des Clusters h aufeinander gestapelt werden und der Vektor $\mathbf{x}^{h(h)} = \text{col}((x_i^{(h)})_{i \in \mathcal{I}^h}) \in \mathbb{R}^{N^h d^h}$, der die

Schätzungen bezüglich der Clusterstrategie $x^{(h)}$ der Agenten des Clusters h vektorisiert, definiert. Analog ist der Vektor $\mathbf{x}^{h(-h)}$ festgelegt. Folglich wird differenziert zwischen Schätzungen bezüglich des Clusters h , gekennzeichnet durch das geklammerte Superskript (h) und der Zugehörigkeit zu Cluster h , gekennzeichnet durch das einfache Superskript h .

Unter Verwendung dieser Definitionen kann nun die Problemstellung aus Problem 5 unter Berücksichtigung der verteilten Information und damit des zu lösenden Schätzungsproblems erweitert werden. Es ergibt sich

Problem 6 (*Multiclusterspiel mit Schätzungsproblem*)

$$\min_{\mathbf{x}} \frac{1}{N^h} \sum_{i=1}^{N^h} f_i^h \left(x_i^{(h)}, x_i^{(-h)} \right), \quad (5.4a)$$

$$\text{s.t. } \mathbf{x}^{h(h)} \in \mathcal{X}^h, \quad (5.4b)$$

$$(I_{Nn} - \mathbf{W})\mathbf{x} = 0, \quad (5.4c)$$

wobei $\mathcal{X}^h := \{\mathbf{x}^{h(h)} | x_i^{(h)} \in \mathcal{X}^h, \forall i \in \mathcal{I}^h\}$. Der Gesamtstrategievektor \mathbf{x} muss entsprechend in der Menge \mathcal{X} liegen, die durch

$$\mathcal{X} := \{\mathbf{x} \in \mathbb{R}^{Nd} | \mathbf{Q}\mathbf{x} \in \mathcal{X}^1 \times \mathcal{X}^2 \times \dots \times \mathcal{X}^H\} \quad (5.5)$$

definiert ist. Unter Verwendung der Notation $d^{<h} := \sum_{j=1}^{h-1} d^j$ und $d^{>h} := \sum_{j=h+1}^H d^j$ ist die Matrix \mathbf{Q} hierbei definiert durch

$$\begin{aligned} \mathbf{Q}_i^h &= [0_{d^h \times d^{<h}}, I_{d^h}, 0_{d^h \times d^{>h}}], \\ \mathbf{Q}^h &= \text{diag}((\mathbf{Q}_i^h)_{i \in \mathcal{I}^h}) \text{ und } \mathbf{Q} = \text{diag}((\mathbf{Q}^h)_{h \in \mathcal{H}}). \end{aligned} \quad (5.6)$$

Folglich gilt, dass $\mathbf{Q}_i^h x_i = x_i^{(h)}$, $\mathbf{Q}^h \text{col}((x_i)_{i \in \mathcal{I}^h}) = \mathbf{x}^{h(h)}$ und $\mathbf{Q}\mathbf{x} = \text{col}((\mathbf{x}^{h(h)})_{h \in \mathcal{H}})$. Entsprechend kann \mathbf{Q} als Zuordnungsmatrix verstanden werden, die bei der Multiplikation mit Vektor \mathbf{x} die Schätzung jedes Agenten bezüglich der Strategie des eigenen Clusters aus \mathbf{x} extrahiert.

Die Bedingung $(I_{Nn} - \mathbf{W})\mathbf{x} = \mathbf{0}$ mit $\mathbf{W} = W \otimes I_d$, unter Verwendung einer beliebigen doppelt-stochastischen Matrix W , stellt sicher, dass die Agenten sich auf einen Konsensus bezüglich ihrer Schätzungen des Gesamtstrategievektors einigen, d. h. $x_i = x_j, \forall i, j \in \mathcal{I}$. Durch das zusätzliche Schätzungsproblem erhöht sich die Dimension des Optimierungsproblems um den Faktor N .

Damit ein Konsensus erreicht werden kann, müssen die Agenten miteinander kommunizieren. Bezüglich der Kommunikation für die gradientenbasierte Lösung von Multiclusterspielen wird allgemein zwischen zwei verschiedenen Kommunikationsebenen unterschieden, wobei die Kommunikationsverbindungen auf jeder Ebene durch Graphen beschrieben werden, siehe auch Abbildung 5.1. Auf der oberen Ebene vernetzt ein in der vorliegenden Arbeit als global bezeichneter Graph die Agenten unabhängig ihrer Clusterzugehörigkeit. Entsprechend stellen alle Agenten des Systems die Knoten des Graphen dar und der Graph wird durch $\mathcal{G}\{\mathcal{V} \equiv \mathcal{I}, \mathcal{E}, W\}$, mit der Gewichtungsmatrix $W \in \mathbb{R}^{N \times N}$, beschrieben. Über diesen Graphen kommunizieren die Agenten $i \in \mathcal{I}$ miteinander, um einen Konsensus bezüglich ihrer Strategieschätzungen x_i zu erreichen. Entsprechend ist die Vernetzung und damit die Kommunikation über die Cluster Grenzen hinaus erforderlich, auch wenn die Cluster bezüglich ihrer Ziele nicht miteinander kooperieren.

Lokale Graphen $\mathcal{G}^h\{\mathcal{V}^h \equiv \mathcal{I}^h, \mathcal{E}^h, V^h\}$ mit Gewichtungsmatrizen $V^h \in \mathbb{R}^{N^h \times N^h}$ stellen die zweite Kommunikationsebene dar und verbinden die Agenten innerhalb des jeweiligen Clusters $h \in \mathcal{H}$ miteinander. Keiner dieser lokalen Graphen besitzt Kommunikationskanäle, die über die Grenzen des jeweiligen Clusters hinausgehen. Diese zweite Strukturebene ist notwendig für die Koordination der kooperierenden Agenten untereinander und stellt das Erreichen eines Konsensus bezüglich der Schätzung des Clustergradienten sicher, wie in der nachfolgenden Literaturdiskussion und dem anschließend vorgestellten Algorithmus erläutert wird. In Abbildung 5.1 wird auch deutlich, dass der globale Graph, der gestrichelt dargestellt ist, mit den lokalen Graphen innerhalb der Cluster überlappen kann.

Bevor sich der Betrachtung von möglichen Lösungen von Multiclusterspielen gewidmet wird, werden zunächst Pauschalannahmen getroffen, welche die behandelte Problemklasse weiter einschränken sollen:

Pauschalannahme 5.1 (*Differenzierbarkeit und Konvexität*) *Das Spiel $\Gamma(H, \mathcal{X}, \{F^h\}_{h \in \mathcal{H}})$ ist konvex und stetig differenzierbar: Alle lokalen Kostenfunktionen $f_i^h(x^h, x^{-h})$, $\forall i \in \mathcal{I}^h, \forall h \in \mathcal{H}$, sind stetig differenzierbar und konvex bezüglich x^h für jedes feste x^{-h} . Zusätzlich gilt für beschränkte Spiele, dass jede Strategiemenge $\mathcal{X}^h, \forall h \in \mathcal{H}$, konvex und kompakt ist.*

Darüber hinaus wird die folgende Pauschalannahme bezüglich des Mappings $M(x)$, definiert in Gleichung (5.3), getroffen:

Pauschalannahme 5.2 *Das Mapping $M(x) = \text{col}((\nabla_h F^h(x))_{h \in \mathcal{H}})$ des Multiclusterspiels $\Gamma(H, \mathcal{X}, \{F^h\}_{h \in \mathcal{H}})$ ist gleichmäßig monoton²¹ auf \mathbb{R}^d mit einer Konstanten $\mu > 0$ sowie lipschitzstetig auf \mathbb{R}^d mit einer Konstanten $L_0 > 0$.*

Ist die Forderung nach der gleichmäßigen Monotonie des Mappings erfüllt, existiert ein eindeutiges Nash-Gleichgewicht des Multiclusterspiels. Darüber hinaus ist diese Eigenschaft notwendig für den Nachweis der linearen Konvergenz des Algorithmus 4.

5.2.1 Übersicht über die Notation

Aufgrund der Einführung der zusätzlichen Hierarchieebene in Form der miteinander in Konkurrenz stehenden Clustern, verkompliziert sich die Notation im Vergleich zu den rein kooperativen Problemen im vorausgegangenen Kapitel 4. Bezüglich der Variablen des zu lösenden Problems muss deutlich gemacht werden, welcher Agent die Variable verwaltet, welchem Cluster der Agent zugeordnet ist und bezüglich welcher Aktion die Schätzung definiert ist. Aus diesem Grund sind in der folgenden Aufzählung alle relevanten Variablen mit ihren jeweiligen Bezeichnern und Dimensionen zusammengefasst:

- $x^h \in \mathbb{R}^{d^h}$ bzw. $x^{-h} \in \mathbb{R}^{d-d^h}$: Strategie des Clusters h bzw. der restlichen Cluster $-h$ bezüglich der Problemstellung 5.
- $x \in \mathbb{R}^d = \text{col}(x^h, x^{-h})$: Gesamtstrategie des Spiels $\Gamma(H, \mathcal{X}, \{F^h\}_{h \in \mathcal{H}})$ zwischen den Clustern bezüglich Problemstellung 5.
- $x_i^{(h)} \in \mathbb{R}^{d^h}$ bzw. $x_i^{(-h)} \in \mathbb{R}^{d-d^h}$: Schätzung des Agenten i bezüglich der Clusterstrategie x^h bzw. bezüglich der Strategien der restlichen Cluster x^{-h} .
- $x_i \in \mathbb{R}^d = \text{col}(x_i^{(h)}, x_i^{(-h)})$: Schätzung des Agenten i bezüglich der Gesamtstrategie x des Spiels zwischen den Clustern.
- $\mathbf{x} = \text{col}((x_i)_{i \in \mathcal{I}}) \in \mathbb{R}^{Nd}$: Vektorisierung der Schätzungen aller Agenten bezüglich der Gesamtstrategie x in einem gestapelten Vektor.

²¹Siehe Definition 2.2.5 in Kapitel 2.

- $\mathbf{x}^h = \text{col}((x_i)_{i \in \mathcal{I}^h}) \in \mathbb{R}^{N^h d}$: Vektorisierung der Schätzungen der Agenten des Clusters h bezüglich der Gesamtstrategie x in einem gestapelten Vektor.
- $\mathbf{x}^{h(h)} = \text{col}((x_i^{(h)})_{i \in \mathcal{I}^h}) \in \mathbb{R}^{N^h d^h}$: Vektorisierung der Schätzungen aller Agenten des Clusters h bezüglich der Strategie x^h des Clusters h in einem gestapelten Vektor.
- $\mathbf{x}^{h(-h)} = \text{col}((x_i^{(-h)})_{i \in \mathcal{I}^h}) \in \mathbb{R}^{N^h(d-d^h)}$: Vektorisierung der Schätzungen aller Agenten des Clusters h bezüglich der Strategie x^{-h} der Cluster $l \in \mathcal{H}/\{h\}$ in einem gestapelten Vektor.

5.2.2 Lösungsbedingungen von Multiclusterspielen

Wie beschrieben muss die Lösung des Multiclusterspiels aus Problem 5 zwei Bedingungen erfüllen: Zum einen muss das kooperative Optimum der lokalen Kostenfunktionen innerhalb der jeweiligen Cluster erreicht werden und zum anderen muss sich mit dem Nash-Gleichgewicht zwischen den Clustern ein stabiler Zustand einstellen. Bei Betrachtung der Definition des Nash-Gleichgewichts für ein Multiclusterspiel in Gleichung (5.2) lässt sich feststellen, dass bei als fest angenommen Strategien $x^{-h, \text{NE}}$ der Cluster $-h$ für die Nash-Gleichgewichtsstrategie von Spieler h

$$x^{h, \text{NE}} = \underset{x^h \in \mathcal{X}^h}{\text{argmin}} F^h(x^h, x^{-h, \text{NE}}) = \underset{x^h \in \mathcal{X}^h}{\text{argmin}} \frac{1}{N^h} \sum_{i=1}^{N^h} f_i^h(x^h, x^{-h, \text{NE}}) \quad (5.7)$$

gilt. Befinden sich die Cluster im Nash-Gleichgewicht zueinander, werden folglich per Definition die lokalen, kooperativen Probleme innerhalb der Cluster optimal gelöst. Daher reicht es aus, das Nash-Gleichgewicht als Lösungsprinzip des Multiclusterspiels zu betrachten. Entsprechend wird im Folgenden der Nash-Gleichgewichtsvektor x^{NE} durch x^* ausgedrückt und damit als Lösung des Multiclusterspiels bezeichnet.

Nach dem Grundlagenabschnitt 2.4 kann die Existenz eines Nash-Gleichgewichts in nicht-kooperativen Spielen durch die Betrachtung der Monotonieeigenschaft des Spiel mappings nachgewiesen werden. Da es sich bei dem Multiclusterspiel um ein nicht-kooperatives Spiel handelt und laut Pauschalannahme 5.2 das Mapping $M(x)$, siehe Definition in Gleichung (5.3), des Multiclusterspiels gleichmäßig monoton ist, existiert in

der so eingeschränkten Problemklasse ein eindeutiges Nash-Gleichgewicht x^* für die Problemstellung 5.

Komplizierter gestaltet sich die Betrachtung von Lösungen des um das Schätzungsproblem erweiterten Multiclusterspiels aus Problem 6. Zugehörig zu dieser Problemstellung ist das entsprechend erweiterte Mapping $M : \mathcal{X} \rightarrow \mathbb{R}^{\sum_{h=1}^H N^h d^h}$, definiert durch

$$M(\mathbf{x}) = \text{col} \left(\left(\mathbf{1}_{N^h} \otimes \frac{1}{N^h} \sum_{i=1}^{N^h} \nabla_h f_i^h \left(x_i^{(h)}, x_i^{(-h)} \right) \right)_{h \in \mathcal{H}} \right). \quad (5.8)$$

In nachfolgendem Lemma werden Verbindungen zwischen der Lösung der Problemstellung 5, der zugehörigen Variationsungleichung²² und der erweiterten Problemstellung 6 gezogen:

Lemma 5.2.1 *Es gelten die Pauschalannahmen 5.1 und 5.2. Die folgenden Aussagen sind dann zueinander äquivalent:*

1. *Es existiert eine eindeutige Lösung x^* der Variationsungleichung $VI(\mathcal{X}, M)$, d. h. $(x - x^*)^T M(x^*) \geq 0, \forall x \in \mathcal{X}$.*
2. *x^* ist das eindeutige Nash-Gleichgewicht des Multiclusterspiels $\Gamma(H, \mathcal{X}, \{F^h\}_{h \in \mathcal{H}})$ aus Problem 5.*
3. *$\mathbf{x}^* = \mathbf{1}_N \otimes x^*$ ist ein Fixpunkt der Abbildung $\mathbf{H}(\mathbf{x}) = \text{proj}_{\mathcal{X}} [\mathbf{x} - \mathbf{Q}^T M(\mathbf{x})]$.*
4. *$\mathbf{x}^* = \mathbf{1}_N \otimes x^*$ ist eine Lösung des erweiterten Multiclusterspiels aus Problem 6.*

Die Matrix \mathbf{Q} ist in (5.6) definiert. Der Beweis dieses Lemmas findet sich in Anhang B.1.

Anmerkung 5.2.1 Der dritte Punkt der obigen Äquivalenzdarstellung ist das Pendant der Lösungsbedingung der Multiclustersproblemstellung zu der Optimalitätsbedingung 5 für einfache, nicht-kooperative Spiele aus dem Grundlagenabschnitt 2.4.

Anmerkung 5.2.2 Der Fall von unbeschränkten Multiclusterspielen, d. h. $\mathcal{X}^h \equiv \mathbb{R}^{d^h}$, stellt eine Vereinfachung dar. In diesem gilt, dass der Vektor x^* das Nash-Gleichgewicht des Spiels aus Problem 5 ist, wenn für das

²²Für Definition und weitere Erklärungen von Variationsungleichungen siehe Grundlagenabschnitt 2.4

Spielmapping $M(x^*) = 0$ gilt [127]. Der Definition von $M(x)$ der Gleichung (5.3) entsprechend gilt, dass $\nabla_h F^h(x^*) = 0$ für alle $h \in \mathcal{H}$. Damit folgt für ein $\mathbf{x}^* = \mathbf{1}_N \otimes x^*$, dass auch das Mapping des erweiterten Spiels in \mathbf{x}^* gleich null ist $\mathbf{M}(\mathbf{x}^*) = \mathbf{0}$. Entsprechend lässt sich anhand der weiteren Argumentation aus Lemma 5.2.1 feststellen, dass wenn x^* ein Nash-Gleichgewicht des Problems 5 darstellt, \mathbf{x}^* eine stabile Lösung des erweiterten Problems 6 ist.

Zusammengefasst folgt aus Lemma 5.2.1, dass wenn die Schätzungen x_i der Agenten $i \in \mathcal{I}$ bezüglich des Gesamtstrategievektors x einen Konsensus erreichen, d.h. $x_i = x_j, \forall i, j \in \mathcal{I}$, und der Konsensusvektor dem Nash-Gleichgewicht x^* des Spiels $\Gamma(H, \mathcal{X}, \{F^h\}_{h \in \mathcal{H}})$ entspricht, dann kann das Problem 6 als gelöst betrachtet werden. Zusätzlich wird im dritten Punkt des Lemmas ein notwendiges und hinreichendes Kriterium definiert, mit dem der Vektor \mathbf{x}^* auf Optimalität überprüft werden kann. Diese Bedingung ist eine für das Schätzungsproblem erweiterte Version der Fixpunktoptimalitätsbedingung 5 aus dem Grundlagenabschnitt 2.4 für nicht-kooperative Spiele.

Diese Fixpunktoptimalitätsbedingung ist entscheidend für den Konvergenzbeweis eines entsprechenden Algorithmus. Bezüglich der Anwendung gestaltet sich allerdings problematisch, dass das in der Abbildung $\mathbf{H}(\mathbf{x})$ enthaltene Mapping $\mathbf{M}(\mathbf{x})$ des erweiterten Spiels 6 im Allgemeinen nicht gleichmäßig monoton ist, auch wenn dies für das Mapping $M(x)$ des ursprünglichen Spiels 5 ohne Schätzungsproblem gilt. Folglich können entsprechende Abschätzungen des Mappings im Konvergenzbeweis nicht durchgeführt werden und es müssen alternative Lösungswege gefunden werden.

5.2.3 Sonderfälle

Es existieren zwei relevante Sonderfälle, die an den Rändern des Definitionsbereiches von Multiclusterspielen liegen und sich zu bekannten Problemstellungen reduzieren. Dabei wird der unbeschränkte Fall untersucht, die Ergebnisse gelten allerdings analog für beschränkte Probleme.

Zunächst wird der Fall betrachtet, dass nur ein einziger Cluster im Agentensystem existiert, d.h. $\mathcal{H} = \{1\}$, der alle N Agenten beinhaltet. Da für diesen Cluster kein Gegenspieler existiert, fällt der nicht-kooperative Anteil des Multiclusterspiels weg und das Problem reduziert sich auf das aus Kapitel 4 bekannte, rein kooperative, verteilte Optimierungsproblem der

Form

$$\min_x F^1(x) = \min_x \sum_{i=1}^{N^1} f_i^1(x). \quad (5.9)$$

Ein weiterer Sonderfall entsteht, wenn es mehr als einen Cluster im Netzwerk gibt, $|\mathcal{H}| > 1$, allerdings jeder dieser Cluster nur einen einzigen Agenten umfasst, d. h. $N^h = 1, \forall h \in \mathcal{H}$. In diesem Fall entfällt der kooperative Teil des Multiclusterspiels, da sich die Clusterkostenfunktion auf eine einzige, dem jeweiligen Spieler bekannte Kostenfunktion reduziert. Entsprechend verbleibt nur das nicht-kooperative Spiel zwischen den Clustern, sodass die allgemeine Form eines Nash-Gleichgewichtsproblems

$$\min_{x^h} f_1^h(x^h, x^{-h}), \forall h \in \mathcal{H},$$

resultiert. Damit kann festgehalten werden, dass Multiclusterspiele sich sowohl als Kombination von nicht-kooperativen Spielen und verteilter, kooperativer Optimierung verstehen lassen sowie als Verallgemeinerung dieser Sonderfälle. Daraus folgt, dass Algorithmen für Multiclusterspiele auch klassische Nash-Gleichgewichtsprobleme und kooperative, verteilte Optimierungsprobleme²³ unter bestimmten Annahmen lösen können.

5.3 Stand der Forschung und Beiträge

Im Vergleich zur kooperativen, verteilten Optimierung, die in Kapitel 4 behandelt wird, ist die Forschung bezüglich Multiclusterspielen noch relativ jung. Die ersten Veröffentlichungen zur Lösung dieser Probleme beschäftigen sich mit zeitkontinuierlichen Nash-Gleichgewichtsproblemstellungen. Da allerdings zeitdiskrete Algorithmen im Fokus der vorliegenden Arbeit stehen, wird im Folgenden nur eine grobe Übersicht über zeitkontinuierliche Problemstellungen gegeben und anschließend detaillierter auf den zeitdiskreten Fall eingegangen.

5.3.1 Zeitkontinuierliche Algorithmen

Die Arbeiten [128] und [129] betrachten unbeschränkte Problemstellungen und schlagen gradientenbasierte Algorithmen vor, um das zeitkontinuierliche Multiclusterspiel zu lösen. Die Arbeiten aus [130] und [131] bauen

²³Für eine explizite Formulierung des Lösungsalgorithmus für kooperative Probleme siehe Abschnitt 4.6.

auf diesen Arbeiten auf, lösen jedoch die Problemstellung unter Verwendung eines payoffbasierten Ansatzes, bei dem die Agenten nur einzelne Werte ihrer lokalen Kostenfunktionen auswerten können und explizit der Gradient nicht zur Verfügung steht. Auch bei diesen Arbeiten werden unbeschränkte Problemstellungen betrachtet. In [132] wird ein gradientenbasiertes Verfahren vorgestellt, bei dem die Strategiemengen durch Projektion der Gradientenschritte auf diese Mengen im Optimierungsverfahren berücksichtigt werden können.

Zusätzlich existieren Arbeiten für Multiclusterspiele, bei denen die Nebenbedingungen der Cluster untereinander verkoppelt sind und somit das Verallgemeinerte Nash-Gleichgewicht den zu identifizierenden, stabilen Zustand darstellt²⁴. In den Artikeln [133][134] werden gradientenfreie Algorithmen vorgestellt, die entsprechend beschränkte Problemstellungen lösen können. Für beide Methoden werden die Agenten der Cluster in Leader und Follower eingeteilt. Im Gegensatz dazu basiert der Ansatz aus [135] auf einer Leader-freien Struktur und ermöglicht die Identifikation des Generalisierten Nash-Gleichgewichts unter Verwendung von Gradienteninformationen.

5.3.2 Zeitdiskrete Algorithmen

Die Forschung bezüglich zeitdiskreter Multiclusterspiele ist insgesamt weniger stark fortgeschritten. Beispielsweise existieren zum aktuellen Zeitpunkt noch keine Algorithmen für Verallgemeinerte Nash-Gleichgewichte. Für Nash-Gleichgewichtsprobleme kann die veröffentlichte Literatur bezüglich folgender Eigenschaften unterschieden werden:

- gradientenbasierte und gradientenfreie Verfahren,
- unbeschränkte und beschränkte Problemstellungen,
- Leader-Follower und Leader-freie Hierarchien und
- doppelt-stochastische und spalten- bzw. reihen-stochastische Kommunikationsmatrizen.

Als Leader-Follower-Hierarchie wird in der vorliegenden Arbeit folgende Clusterstruktur bezeichnet: ein Agent in jedem Cluster $h \in \mathcal{H}$ stellt den

²⁴Zur Unterscheidung von Nash-Gleichgewichten und Generalisierten Nash-Gleichgewichten, siehe Abschnitt 2.4.

Clusterleader dar, der mit den Leadern der restlichen Cluster kommuniziert. Darüber hinaus findet keine inter-Cluster Kommunikation statt. Der Clusterleader kommuniziert mit den restlichen Followeragenten seines Clusters und zieht deren Schätzung in Richtung der Informationen, die der Leader von den restlichen Clusterleadern erhalten hat²⁵.

Die Beiträge der vorliegenden Arbeit bezüglich der Multiclusterspiele sind im Bereich der gradientenbasierten Verfahren verortet. Für diese Verfahren lässt sich unter Berufung auf bereits veröffentlichte Literatur ein gemeinsames Schema identifizieren, welches im Folgenden vorgestellt werden soll, bevor auch eine Übersicht über gradientenfreie Verfahren gegeben wird.

Schema gradientenbasierter Verfahren Klassische, konvexe Spiele $\Gamma(H, \mathcal{X}, \{F^h\}_{h \in \mathcal{H}})$ sind, wie in Unterabschnitt 5.2.3 festgestellt, äquivalent zu Multiclusterspielen, wenn die Cluster nur jeweils einen Agenten umfassen. Sind Gradienteninformationen in solchen Spielen verfügbar, kann das Nash-Gleichgewicht mit einem weithin bekannten Verfahren, einer Fixpunktiterationen bestehend aus projizierten Gradientenschritten, identifiziert werden. Jeder Spieler h berechnet hierbei den Gradienten seiner Kostenfunktion F^h , ausgewertet an dem Vektor der Gesamtstrategie $x(k)$, und führt anschließend den Schritt

$$x^h(k+1) = \text{proj}_{\mathcal{X}^h} [x^h(k) - \gamma \nabla_h F^h(x(k))] \quad (5.10)$$

aus. Bei geltenden Pauschalannahmen 5.1 und 5.2 sowie einer konstanten, positiven Schrittweite γ konvergieren die Strategien unter Anwendung dieses Schemas zu einem Nash-Gleichgewicht mit linearer Konvergenzrate, siehe hierzu beispielsweise [57].

Dieses Schema ist für allgemeine Multiclusterspiele mit $N^h > 1$ in dieser Form allerdings nicht anwendbar, da die Agenten nur ihre eigene Kostenfunktion f_i^h kennen und nicht die Gesamtkostenfunktion F^h des Clusters. Aus diesem Grund muss das obige Schema erweitert werden, indem in

²⁵Die hier beschriebene Leader-Follower-Hierarchie muss begrifflich zum Leader-Follower-Konsensus abgegrenzt werden: Als Leader-Follower-Konsensus wird in der Literatur, siehe beispielsweise [25][26][136], ein Verfahren bezeichnet, bei dem die Follower der Trajektorie eines (virtuellen) Leaders folgen, dessen Position oder Geschwindigkeit geschätzt werden muss. Dieser Ansatz weist große Übereinstimmungen mit dem *Gradient-Tracking*-Ansatz auf, der eine bedeutende Relevanz für dieses Kapitel hat, siehe nachfolgende Beschreibungen um die Gleichung (5.11).

einer zusätzlichen Aktualisierungsvorschrift der Gradient des Clusters geschätzt wird. Hierzu bietet sich das *Gradient-Tracking*-Verfahren an, das auch bereits im Kapitel 4 angesprochen wird. Dieses Verfahren basiert auf einer Methodik aus der Quelle [112], mit der ein zeitvariantes Signal in einem Agentensystem geschätzt werden kann. In Bezug auf die zu lösende Problemstellung kann der Gradient der lokalen Kostenfunktionen als ein solches Signal interpretiert werden. Die Schätzung wird in der Variablen $y_i^h \in \mathbb{R}^{d^h}$ gespeichert und von den jeweiligen Agenten $i \in \mathcal{I}^h$ über die Vorschrift

$$y_i^h(k+1) = \sum_{j=1}^{N^h} A_{ij}^h y_j^h(k) + \nabla_h f_i^h(x_i(k+1)) - \nabla_h f_i^h(x_i(k)) \quad (5.11)$$

aktualisiert, wobei A^h die Gewichtungsmatrix eines lokalen Graphen $\mathcal{G}^h\{\mathcal{I}^h, \mathcal{E}^h, A^h\}$ darstellt, der ausschließlich die Agenten in Cluster h verbindet, siehe zur Veranschaulichung die Grafik 5.1. Der Vektor $x_i(k)$ entspricht hierbei der aktuellen Schätzung des Agenten bezüglich der Gesamtstrategie des Spiels. Initialisiert werden die Variablen unter Verwendung der Startschätzungen $x_i(0)$ mit

$$y_i^h(0) = \nabla_h f_i^h(x_i(0)). \quad (5.12)$$

Unter Verwendung dieser Initialisierung gilt nach der Vorschrift (5.11) für die vektorisierten Schätzungen $y^h(k) = \text{col}((y_i^h(k))_{i \in \mathcal{I}^h})$ in den jeweiligen Zeitschritten k für die Schätzung im Cluster h

$$\mathbf{1}^T y^h(k) = \sum_{i=1}^{N^h} \nabla_h f_i^h(x_i(k)), \quad (5.13)$$

wenn die Matrix A^h doppelt-stochastisch gewichtet ist, d. h. $\mathbf{1}^T A^h = \mathbf{1}^T$. Das bedeutet, dass die Summe über die Schätzvariablen in jedem Zeitschritt k gleich der Summe der Gradienten der Kostenfunktionen der Clusteragenten an ihren unterschiedlichen Zustandsschätzungen $x_i(k)$ ist, siehe hierzu beispielsweise die Argumentation in [31]. Konvergieren nun die Schätzungen $x_i(k)$ für alle $i \in \mathcal{I}^h$ zu einem Konsensusvektor x_K , so konvergiert auch die Gradientenschätzung zu einem Konsensus y_K^h , für den folglich

$$\mathbf{1}^T y^h(\infty) = N^h y_K^h = \sum_{i=1}^{N^h} \nabla_h f_i^h(x_K), \quad (5.14)$$

gilt und somit $y_K^h = \frac{1}{N^h} \sum_{i=1}^{N^h} \nabla_h f_i^h(x_K)$, was dem Clustergradienten bezüglich Problem 5 entspricht²⁶.

Die beschriebene und motivierte Struktur der Gradientenschätzung in den jeweiligen Clustern aus Gleichung (5.11) wird von vielen verteilten Algorithmen zur Identifizierung von Nash-Gleichgewichten in Multiclusterspielen verwendet. In [24] wird die lineare Konvergenz eines entsprechend formulierten Verfahrens zur Lösung von unbeschränkten Problemstellungen²⁷ unter Verwendung von doppelt-stochastischen Kommunikationsmatrizen bewiesen, wobei die Kommunikation zwischen den Clustern ausschließlich über die Leader des Clusters abläuft. Wie auch in [34] untersucht, stellt diese Leaderstruktur eine Einschränkung der Allgemeinheit der Kommunikation zwischen den Clustern dar und kann die Konvergenzgeschwindigkeit negativ beeinflussen. In der eigenen Veröffentlichung [38] wird die Kommunikationsstruktur verallgemeinert: Sowohl für die Kommunikation innerhalb der Cluster als auch für die Kommunikation über die Cluster Grenzen hinaus werden gerichtete Graphen angenommen, die spalten- bzw. reihen-stochastisch gewichtet sind. Zusätzlich wird die Hierarchie innerhalb der Cluster aufgelöst, sodass alle Agenten gleichwertig sind und mehrere Agenten, nicht nur der Clusterleader, über die Cluster Grenzen hinaus kommunizieren können. Ähnliche Ergebnisse sind auch in [25] und [26] veröffentlicht. Dieser Algorithmus dient als Grundlage für den Algorithmus des Beitrages 5. Eine bisher offene Fragestellung ist die Einbeziehung von Nebenbedingungen in die gradientenbasierte Aktualisierungsstruktur unter Verwendung der Gradientenschätzung aus Gleichung (5.11) bei hierarchielosen Clustern.

Wie auch im zeitkontinuierlichen Bereich sind auch für diskrete Zeitschritte gradientenfreie Algorithmen entworfen worden, bei denen die Agenten keine Kenntnis bezüglich der funktionalen Form ihrer Kostenfunktionen besitzen. In diesen Fällen haben die Agenten die Möglichkeit, den aktuellen Wert ihrer Kostenfunktion an einem oder mehreren Abfragepunkten von einem Orakel zu erfahren. Solche Szenarien treten beispielsweise in Energiemarktsituationen auf, bei denen die Energiepreisfunktionen den Marktteilnehmer unbekannt sind und der Marktopera-

²⁶Die Überlegungen stellen lediglich eine Veranschaulichung des *Gradient-Tracking*-Prinzips dar. Bei rigorosen Konvergenzbeweisen müssen die Konvergenz der Strategien x_i^h und der *Gradient-Tracking*-Variablen y_i^h gemeinsam betrachtet werden.

²⁷In einer späteren Version der Veröffentlichung werden die Ergebnisse auf beschränkte Problemstellungen erweitert. Die Leader-Follower-Struktur bleibt weiterhin bestehen.

tor die Energiepreise auf Basis der Gebote festlegt, siehe beispielsweise [15]. In [27] ist ein solcher gradientenfreier Algorithmus für die Lösung von unbeschränkten Multiclusterspielen veröffentlicht. Die Einführung einer Einteilung in Leader- und Follower-Agenten ist nicht notwendig. Dieser Ansatz verwendet ebenfalls die *Gradient-Tracking*-Struktur aus Gleichung (5.11) zur Schätzung des Clustergradienten. Die Auswertung der lokalen Gradienten wird hierbei allerdings durch Gradientenschätzungen ersetzt, welche die lokalen Gradienten unter Verwendung des Kostenfunktionswertes, bereitgestellt vom Orakel, schätzt. Hierbei müssen pro Iteration die Kostenfunktionen an zwei Abfragepunkten ausgewertet werden. Zudem konvergiert der Algorithmus nur in einer Umgebung des Nash-Gleichgewichts, wobei der Fehler unter anderem proportional von der konstanten Schrittweite abhängt. Die Agenten innerhalb der Cluster sind über einen doppelt-stochastisch gewichteten Kommunikationsgraphen verbunden, um die Clusterstrategie- und Clustergradientenschätzung durchführen zu können.

In [36] wird ein gradientenfreier Algorithmus präsentiert, der fast sicher²⁸ zu dem tatsächlichen Nash-Gleichgewicht konvergiert und nicht nur in dessen Umgebung. Zusätzlich können in diesem Algorithmus Nebenbedingungen durch Projektionen auf die jeweilige Strategiemenge berücksichtigt werden. Im Gegensatz zu den bisher betrachteten Algorithmen wird hier allerdings nicht die *Gradient-Tracking*-Struktur verwendet, um den Clustergradienten zu schätzen. Vielmehr wird der lokale Gradient der Agenten unter Verwendung von Orakelinformationen an bestimmten Abfragepunkten geschätzt und ein *verteilter Gradientenabstieg* unter Verwendung der aktuellen Schätzung durchgeführt. Dadurch müssen die lokalen Kostenfunktionen in jeder Iteration vom Orakel nur an einer Stelle ausgewertet werden. Auch für diesen Algorithmus sind die Cluster leaderfrei und die Agenten kommunizieren über einen doppelt-stochastisch gewichteten Kommunikationsgraphen. Ein Informationsaustausch zwischen den Clustern ist bei diesen beiden Algorithmen unter der beschriebenen Informationsstruktur nicht notwendig, die Agenten müssen jedoch Abfragepunkte an das Orakel schicken und entsprechende Auswertungen der Kostenfunktionen erhalten können.

In Tabelle 5.1 ist eine Übersicht über die Eigenschaften der Algorithmen aus den besprochenen Quellen zusammen mit dem Beitrag 5 dieses

²⁸Es handelt bei diesem Algorithmus, wie auch bei dem Algorithmus aus [27] um einen stochastischen Prozess, bei dem die Konvergenz mit fast sicherer Wahrscheinlichkeit festgestellt werden kann.

Kapitels aufgeführt. Zusätzlich zu den bereits betrachteten Merkmalen sind dort Monotonieeigenschaften des Mappings $M(x)$, Eigenschaften der Schrittweite und Konvergenzrate, sofern identifiziert, angegeben. Es fällt auf, dass alle in der Tabelle vermerkten Algorithmen, mit Ausnahme der Veröffentlichung [36], ein gleichmäßig monotones Spielmapping $M(x)$ benötigen, damit Konvergenz des jeweiligen Algorithmus nachgewiesen werden kann. Für diese Algorithmen ist auch die Verwendung einer konstanten Schrittweite und der Nachweis einer linearen Konvergenzrate möglich. Der Algorithmus aus [36] mit der allgemeineren Anforderung der strengen Monotonie des Mappings verwendet eine zeitvariante, absinkende Schrittweitenfolge und es wird keine Konvergenzrate nachgewiesen. Darüber hinaus wird aus der Tabelle deutlich, dass der unbeschränkte Fall bisher weiter ausgearbeitet ist als Lösungsalgorithmen für beschränkte Probleme.

Bezüglich der betrachteten Eigenschaften und der Struktur sind die Algorithmen aus [25] und [26] gleich zur eigenen Veröffentlichung [38]. Es sei darauf hingewiesen, dass [38] zeitgleich mit der Arbeit [26] erschien, wobei von [26] bisher nur eine nicht begutachtete arXiv-Version existiert. Der Artikel [25] erschien etwa ein Jahr später.

Auch bei dem Artikel [24] handelt es sich um eine nicht begutachtete arXiv-Version. Zur Zeit der Erstellung des Beitrages 5 behandelte der dort veröffentlichte Algorithmus eine unbeschränkte Problemstellung. In einer aktuellen Version sind die Ergebnisse auf beschränkte Problemstellungen unter Verwendung einer Projektion erweitert worden. Im Vergleich zu Beitrag 5 bleibt allerdings der Unterschied bezüglich der Leader-Follower-Einschränkung des Algorithmus aus [24] bestehen.

Die Parallelität der genannten Arbeiten zu den eigenen Beiträgen und die damit verdeutlichte Forschungsaktivität zeigt die Bedeutung der Problemstellung der Multiclusterspiele.

Tabelle 5.1: Übersichtstabelle über Literatur zur zeitdiskreten Lösung von Multiclusterspielen als Nash-Gleichgewichtsproblem zur Einordnung des Beitrages 5. Legende: *¹: Konvergiert nur zu einer Umgebung des Optimums, *²: In einer späteren Version der Veröffentlichung wurden die Ergebnisse auf beschränkte Problemstellungen erweitert. +: doppelt-stochastisch, —: reihen-stochastisch, |: spalten-stochastisch

	[24]	[25]	[26]	[27]	[36]	[38]	Beitrag 5
grad.-basiert	✓	✓	✓	×	×	✓	✓
Beschränkt	×* ²	×	×	×	✓	×	✓
leader-frei	×	✓	✓	✓	✓	✓	✓
Gewichtung Komm.	+	—,	—,	+	+	—,	+
Monotonie $M(x)$	gleichm.	gleichm.	gleichm.	gleichm.	strikt	gleichm.	gleichm.
Schrittvariable	konstant	konstant	konstant	konstant	zeit-var.	konstant	konstant
∇f_i lipschitzstetig	✓	✓	✓	✓	✓	✓	✓
Konvergenzrate	linear	linear	linear	geom.* ¹	n.a.	linear	linear

Beitrag 5 ([34]) Dieser Beitrag befasst sich mit einem gradientenbasierten Algorithmus zur verteilten Lösung von beschränkten Nash-Gleichgewichtsproblemen in Multiclusterspielen. In diesem Algorithmus wird die Struktur aus der eigenen Veröffentlichung [38] durch eine Projektion erweitert, sodass nun die Nebenbedingungsmengen der Cluster berücksichtigt werden können. Zeitgleich zur Erstellung des Beitrags wurde die arXiv-Veröffentlichung [24] ebenfalls auf beschränkte Problemstellungen erweitert. Im Vergleich zu [24] besitzt dieser Beitrag eine allgemeinere Kommunikationsstruktur, da keine Einschränkung auf eine ausschließliche Leaderkommunikation zwischen den Clustern erfolgt.

Damit schließt dieser Algorithmus die im Stand der Forschung identifizierte und in Tabelle 5.1 verdeutlichte Forschungslücke bezüglich der gradientenbasierten Lösung von beschränkten Multiclusterspielen bei hierarchiefreien Clustern.

Für den Konvergenzbeweis wird davon ausgegangen, dass die Clusterbeschränkungsmengen allen Agenten des jeweiligen Clusters bekannt sind. Unter doppelt-stochastischer Gewichtung der Kommunikation kann eine lineare Konvergenzrate des Algorithmus bewiesen werden.

Beitrag 6 Der zweite Beitrag dieses Kapitels umfasst die simulative Überprüfung der Konvergenz des Algorithmus des Beitrages 5 in Anwendung auf das Einsatzplanproblem zwischen Microgrids, formuliert in Kapitel 3. Darüber hinaus wird die Lösung des Spiels zwischen den Microgrids diskutiert. Weitere Simulationen bezüglich einer ähnlichen Problemstellung finden sich in der Veröffentlichung [34].

5.4 Algorithmus 4: Lösung beschränkter Multiclusterspiele

In diesem Abschnitt wird ein Algorithmus vorgestellt, mit dem Multiclusterspiele der Form aus den Problemstellungen 5 bzw. 6 gelöst werden können. Zunächst wird der Algorithmus selbst im Detail beschrieben, bevor anschließend der Konvergenzbeweis geführt wird. Abschließend folgen Simulationsergebnisse des Algorithmus in Anwendung auf die Microgridproblemstellung aus Kapitel 3.

5.4.1 Algorithmus

Wie bereit im Abschnitt 5.3 zum Stand der Forschung beschrieben, besteht der Kern des im Folgenden vorgestellten Algorithmus aus der Kombination eines Fixpunkt-Iterationsschemas zur Lösung von klassischen Nash-Gleichgewichtsproblemen und der Schätzung der jeweiligen Clustergradienten durch das Tracking-Verfahren nach [112]. Zusätzlich muss, wie im Abschnitt über die Problemstruktur deutlich gemacht, eine zusätzliche Schätzung bezüglich der Strategievarenablen sowohl des eigenen Clusters, als auch der restlichen Cluster stattfinden. Als Erweiterung zur Veröffentlichung [38], welche die Grundlage für die Konstruktion dieses Algorithmus bildet, werden bei dem nachfolgenden Algorithmus die Beschränkungsmengen \mathcal{X}^h der Cluster $h \in \mathcal{H}$ durch eine Projektion berücksichtigt. Allerdings muss hierfür mit der doppelt-stochastischen Gewichtung eine weniger allgemeine Kommunikationsgewichtung angenommen werden.

Um die Schätzungen zu ermöglichen, ist der Austausch von Informationen unter den Agenten unerlässlich. Deswegen sind die Agenten durch zwei verschiedene Kommunikationsstrukturen miteinander verbunden: Einen globalen Graphen $\mathcal{G}\{\mathcal{V} \equiv \mathcal{I}, \mathcal{E}, W\}$ mit Gewichtungsmatrix $W \in \mathbb{R}^{N \times N}$, der die Agenten des Gesamtsystems unabhängig der jeweiligen Clusterzugehörigkeit vernetzt, und lokale Graphen $\mathcal{G}^h\{\mathcal{V}^h \equiv \mathcal{I}^h, \mathcal{E}^h, V^h\}$ mit Gewichtungsmatrizen $V^h \in \mathbb{R}^{N^h \times N^h}$, welche die Agenten innerhalb der jeweiligen Cluster $h \in \mathcal{H}$ vernetzt. Dargestellt sind diese Graphen beispielhaft in Abbildung 5.1 zu Beginn des Kapitels.

Um die Schätzungen der Agenten über die Iterationen hinweg aneinander anzunähern und damit die Konsensusnebenbedingung (5.4c) der Problemstellung 6 zu erfüllen, wird ein Konsensusverfahren verwendet, siehe hierzu auch den Grundlagenabschnitt 2.5. Das Nash-Gleichgewicht wird durch eine gradientenbasierte Fixpunktiteration mit Projektion auf die jeweilige Beschränkungsmenge des Clusters ermittelt, während in einer zusätzlichen Iteration der Gesamtgradient des Clusters geschätzt wird. Zusammengekommen ergibt sich der Algorithmus 4.

Algorithmus 4 Projizierte *Gradient-Tracking*-Methode für beschränkte Multiclusterspiele

Initialisierung: $y_i^h(0) = \nabla_h f_i^h(x_i(0))$, $x_i(0) = \hat{x}_i(0) = x_0$, $\forall i \in \mathcal{I}^h, \forall h \in \mathcal{H}$.

Iteriere bis zur Konvergenz: Jeder Agent i berechnet

$$\hat{x}_i(k) = \sum_{j=1}^N W_{ij} x_j(k), \quad (5.15a)$$

$$x_i^{(h)}(k+1) = \text{proj}_{\mathcal{X}^h} \left[\hat{x}_i^{(h)}(k) - \alpha y_i^h(k) \right], \quad (5.15b)$$

$$x_i^{(-h)}(k+1) = \hat{x}_i^{(-h)}(k), \quad (5.15c)$$

$$y_i^h(k+1) = \sum_{j=1}^{N^h} V_{ij}^h y_j^h(k) + \nabla_h f_i^h(x_i(k+1)) - \nabla_h f_i^h(x_i(k)). \quad (5.15d)$$

Die Initialisierung der Variablen erfolgt wie im Algorithmus 4 beschrieben. Besonders wichtig ist hierbei die Initialisierung von $y_i^h(k)$, da ansonsten nicht der korrekte Clustergradient geschätzt werden kann, siehe hierfür Diskussion in Abschnitt 5.3.

In jeder Iteration startet der Algorithmus in Gleichung (5.15a) zunächst mit einem Konsensusschritt. Hierbei aktualisieren die Agenten i ihre eigene Schätzung $x_i(k)$ bezüglich der Gesamtstrategie des Spiels durch die Bildung einer gewichteten Summe aus den übermittelten Schätzungen ihrer direkten Nachbarn, festgelegt durch den globalen Kommunikationsgraphen, und ihrer eigenen Schätzung. Anschließend wird bezüglich der aktualisieren Strategieschätzung des eigenen Clusters $x_i^{(h)}$ ein Gradientenschritt durchgeführt, der das Ziel hat die Clusterkostenfunktion zu minimieren. Da allerdings der Clustergradient selbst nicht zur Verfügung steht, wird die aktuelle Schätzung dieses Gradienten, gespeichert in $y_i^h(k)$, verwendet, um den Schritt, gesteuert über die Schrittweite $\alpha > 0$, durchzuführen. Durch eine Projektion auf die Beschränkungsmenge \mathcal{X}^h des Clusters wird sichergestellt, dass die Nebenbedingungen eingehalten werden. Da die Agenten nur die Strategie des eigenen Clusters anpassen dürfen, werden in Gleichung (5.15c) die Ergebnisse des Konsensusverfahrens bezüglich der Schätzung der restlichen Cluster $-h$ übernommen. In Gleichung (5.15d) wird schließlich die Aktualisierung der Cluster-

gradientenschätzung $y_i^h(k)$ vorgenommen. Hierzu erfolgt zunächst ein Konsensusschritt über die jeweiligen lokalen Graphen \mathcal{G}^h unter Verwendung der gewichteten Kommunikationsmatrix V^h , um die Schätzung der Agenten des Clusters aneinander anzupassen. Anschließend werden die lokalen Gradienten an den agenteneigenen Strategieschätzungen zu den Zeitpunkten $k + 1$ und k ausgewertet und deren Differenz zu dem Konsensusergebnis addiert. Diese Aktualisierungsgleichung repräsentiert die Schätzungsgleichung für zeit-variante Signale, veröffentlicht in [112]. Wie im Abschnitt 5.3 zum Abschnitt zum Stand der Forschung motiviert, kann bei entsprechender Initialisierung der Schätzung unter der Annahme, dass bezüglich der Strategieschätzungen ein Konsensus x_K erreicht wird, die Existenz des Grenzwertes $\lim_{k \rightarrow \infty} y_i^h(k) = \frac{1}{N^h} \sum_{i=1}^{N^h} \nabla_h f_i^h(x_K)$ angenommen werden.

Unter Definition der Vektoren bzw. Matrizen

$$\begin{aligned} \mathbf{y}^h(k) &= \text{col}((y_i^h(k))_{i \in \mathcal{I}^h}) \in \mathbb{R}^{d^h N^h}, \\ \mathbf{y}(k) &= \text{col}((\mathbf{y}^h(k))_{h \in \mathcal{H}}) \in \mathbb{R}^{\sum_{h=1}^H d^h N^h}, \\ \mathbf{G}^h(k) &= \text{col}((\nabla_h f_i^h(x_i(k))_{i \in \mathcal{I}^h})) \in \mathbb{R}^{d^h N^h}, \\ \mathbf{G}(k) &= \text{col}((\mathbf{G}^h(k))_{h \in \mathcal{H}}) \in \mathbb{R}^{\sum_{h=1}^H d^h N^h}, \\ \mathbf{W} &= W \otimes \mathbf{I}_d \in \mathbb{R}^{Nd \times Nd}, \\ \mathbf{V} &= \text{diag}((V^h \otimes \mathbf{I}_{d^h})_{h \in \mathcal{H}}) \in \mathbb{R}^{\sum_{h=1}^H d^h N^h}, \end{aligned}$$

kann der Algorithmus 4 in vektorisierter Form angegeben werden:

$$\hat{\mathbf{x}}(k) = \mathbf{W} \mathbf{x}(k), \quad (5.16a)$$

$$\mathbf{x}(k+1) = \text{proj}_{\mathcal{X}} [\hat{\mathbf{x}}(k) - \alpha \mathbf{Q}^T \mathbf{y}(k)], \quad (5.16b)$$

$$\mathbf{y}(k+1) = \mathbf{V} \mathbf{y}(k) + \mathbf{G}(k+1) - \mathbf{G}(k), \quad (5.16c)$$

wobei die Menge \mathcal{X} in Gleichung (5.5) und die Matrix \mathbf{Q} in (5.6) definiert ist. Die letzte Gleichung (5.16c) kann in die Aktualisierungsgleichungen

$$\mathbf{y}^h(k) = \mathbf{V}^h \mathbf{y}^h(k) + \mathbf{G}^h(k+1) - \mathbf{G}^h(k), \quad (5.17)$$

für jeden Cluster $h \in \mathcal{H}$ aufgeteilt werden.

5.4.2 Konvergenz

Nachdem nun der Algorithmus vorgestellt ist, kann die Konvergenz des Algorithmus besprochen werden. Zunächst werden hierfür Annahmen bezüglich der Kommunikationsstrukturen angeführt. Eine grundlegende Voraussetzung für das Erreichen eines Konsensus bezüglich Strategie- und Gradientenschätzung, ist die korrekte Gewichtung der lokalen und des globalen Graphen. Bezüglich dieser Graphen werden die folgenden Annahmen getroffen:

Annahme 5.4.1 *Die lokalen Graphen $\mathcal{G}^h\{\mathcal{I}^h, \mathcal{E}^h, V^h\}$, welche die Agenten innerhalb der jeweiligen Cluster h verbinden, sind ungerichtet und verbunden für alle $h \in \mathcal{H}$. Die zugehörigen nicht-negativen, gewichteten Adjazenzmatrizen V^h haben positive Diagonaleinträge $V_{ii}^h > 0, \forall i \in \mathcal{I}^h, \forall h \in \mathcal{H}$, und die Gewichte sind so gewählt, dass die Matrizen jeweils doppelstochastisch sind, d. h. es gilt, $\mathbf{1}^T V^h = \mathbf{1}^T$ und $V^h \mathbf{1} = \mathbf{1}$ für alle $h \in \mathcal{H}$.*

Annahme 5.4.2 *Der globale Graph $\mathcal{G}\{\mathcal{I}, \mathcal{E}, W\}$, der die Agenten unabhängig von der Clusterzugehörigkeit verbindet, ist ungerichtet und verbunden. Die zugehörige nicht-negative, gewichtete Adjazenzmatrix W hat positive Diagonaleinträge $W_{ii} > 0, \forall i \in \mathcal{I}$, und die Gewichte sind so gewählt, dass die Matrix doppelstochastisch ist, d. h. es gilt $\mathbf{1}^T W = \mathbf{1}^T$ und $W \mathbf{1} = \mathbf{1}$.*

Unter diesen Annahmen bestehen die folgenden bekannten und wichtigen Normungleichungen, siehe beispielsweise [95]: Es existieren Skalare $\sigma \in (0, 1)$ und $\sigma_V = \max\{\text{col}((\sigma_V^h)_{h \in \mathcal{H}})\} \in (0, 1)$, sodass für alle $x \in \mathbb{R}^N$ und $y \in \mathbb{R}^{N^h}$ die Ungleichungen

$$\|Wx - \mathbf{1}_N \bar{x}\| \leq \sigma \|x - \mathbf{1}_N \bar{x}\|, \quad (5.18)$$

$$\|V^h y - \mathbf{1}_{N^h} \bar{y}\| \leq \sigma_V \|y - \mathbf{1}_{N^h} \bar{y}\|, \quad (5.19)$$

gelten, wobei $\bar{x} = \frac{1}{N} \mathbf{1}_N^T x$ und $\bar{y} = \frac{1}{N^h} \mathbf{1}_{N^h}^T y$.

Bezüglich der Gradienten der lokalen Kostenfunktionen $f_i^h(x^h, x^{-h})$ wird neben der Pauschalannahme 5.1 der Differenzierbarkeit und Konvexität bei konstanten x^{-h} gefordert, dass

Annahme 5.4.3 *Die Gradienten $\nabla_h f_i^h(x^h, x^{-h})$ der Kostenfunktionen aller Agenten $i \in \mathcal{I}^h, h \in \mathcal{H}$, sind Lipschitzstetig. Es existiert folglich eine Konstante $L_1 > 0$, sodass für $x = \text{col}(x^h, x^{-h})$ und $y = \text{col}(y^h, y^{-h})$*

$$\|\nabla_h f_i^h(x^h, x^{-h}) - \nabla_h f_i^h(y^h, y^{-h})\| \leq L_1 \|x - y\| \quad (5.20)$$

gilt, für alle $x, y \in \mathbb{R}^d$ und $i \in \mathcal{I}^h, h \in \mathcal{H}$.

Präliminarien

Bevor die Konvergenz des oben beschriebenen Algorithmus unter Beweis gestellt werden kann, müssen zunächst einige Definitionen getätigt werden, welche die Notation der Beweisführung erleichtern sollen. Zunächst sei der Vektor $\bar{\mathbf{y}}^h(k) \in \mathbb{R}^{N^h d^h}$ durch

$$\bar{\mathbf{y}}^h(k) = \mathbf{1}_{N^h} \otimes \frac{1}{N^h} \sum_{i=1}^{N^h} \mathbf{y}_i^h(k) = \mathbf{1}_{N^h} \otimes \frac{1}{N^h} \sum_{i=1}^{N^h} \nabla_h f_i^h(x_i(k))$$

definiert. Dieser Vektor beschreibt den Mittelwert der *Gradienten-Tracking*-Variablen $\mathbf{y}_i^h(k)$ des Clusters h zum Zeitpunkt k , der N^h mal gestapelt wird. Die zweite Äquivalenz zur Summe der lokalen Gradienten folgt aus der korrekten Initialisierung und Aktualisierung nach Algorithmus 4.

Es sei $\bar{\mathbf{y}}(k) = \text{col}((\bar{\mathbf{y}}^h(k))_{h \in \mathcal{H}})$ definiert als ein Vektor, der die entsprechende Mittelwertsvektoren aller Cluster untereinander stapelt. Unter Verwendung der Zuordnungsmatrix

$$\mathbf{R}^h = \frac{1}{N^h} \mathbf{1}_{N^h} \mathbf{1}_{N^h}^T \otimes \mathbf{I}_{d^h} \text{ und } \mathbf{R} = \text{diag}((\mathbf{R}^h)_{h \in \mathcal{H}}),$$

wobei $\dim \mathbf{R}^h = N^h d^h \times N^h d^h$ und $\dim \mathbf{R} = \sum_{h=1}^H N^h d^h \times \sum_{h=1}^H N^h d^h$, kann die Beschreibung der Mittelwertsvektoren vereinfacht als

$$\bar{\mathbf{y}}^h(k) = \mathbf{R}^h \mathbf{y}^h(k) \text{ und } \bar{\mathbf{y}}(k) = \mathbf{R} \mathbf{y}(k)$$

angegeben werden.

Um die Konvergenz des Algorithmus 4 bzw. der vektorisierten Form aus den Gleichungen (5.16) zu beweisen, müssen einige Hürden genommen werden. Zum einen müssen drei unterschiedliche Dynamiken berücksichtigt werden: Die Konsensdynamik bezüglich der Schätzungen des Gesamtstrategievektors in Gleichung (5.15a), der eigentliche Optimierungsprozess in Form des projizierten Gradientenabstieges in Gleichung (5.15b) und die Schätzungsdynamik der Gradienten innerhalb der Cluster in Gleichung (5.15d). All diese Dynamiken sind miteinander verkoppelt. Zusätzlich muss die Schrittweite α so gewählt werden, dass eine stabile Konvergenz zum Nash-Gleichgewicht möglich ist.

Ein weiteres Problem besteht in der Tatsache, dass zwar nach Pauschalanahme 5.2 das Mapping $M(x)$, definiert in Gleichung (5.3), als gleichmäßig monoton angenommen wird, aber das erweiterte Mapping $\mathbf{M}(x)$

aus Gleichung (5.8) des Schätzproblems diese Annahme nicht erfüllt. Allerdings ist es möglich, das nachfolgende Lemma bezüglich der Lipschitzstetigkeit des erweiterten Mappings anzuführen, dessen Beweis in [4] zu finden ist:

Lemma 5.4.1 ([4]) *Unter der Pauschalannahme 5.2 ist das erweiterte Mapping $\mathbf{M}(\mathbf{x})$ lipschitzstetig mit einer Konstanten $L > 0$, sodass $0 < \mu \leq L \leq L_0$. μ ist hierbei die Konstante bezüglich der gleichmäßigen Monotonie des Mappings $M(x)$ und L_0 die Konstante der Lipschitzstetigkeit aus Pauschalannahme 5.2.*

Bevor nun im nächsten Abschnitt auf den tatsächlichen Konvergenzbeweis eingegangen werden kann, werden im nachfolgenden Lemma noch einige für den eigentlichen Beweis notwendige Abschätzungen getroffen.

Lemma 5.4.2 *Seien die Pauschalannahmen und die Annahmen 5.4.1, 5.4.2 gegeben. $\mathbf{x}^* = \mathbf{1}_N \otimes x^*$ sei ein Fixpunkt der Abbildung $\mathbf{H}(\mathbf{x}) = \text{proj}_{\mathcal{X}}[\mathbf{x} - \alpha \mathbf{Q}^T \mathbf{M}(\mathbf{x})]$. Dann gilt die Ungleichung*

$$\|\hat{\mathbf{x}}(k) - \mathbf{x}^* - \alpha \mathbf{Q}^T \mathbf{M}(\mathbf{x}(k)) + \alpha \mathbf{Q}^T \mathbf{M}(\mathbf{x}^*)\| \leq \sqrt{\rho(A(\alpha))} \|\mathbf{x}(k) - \mathbf{x}^*\|$$

mit

$$A(\alpha) = \begin{bmatrix} \frac{\alpha^2 L_0^2}{N} - \frac{2\alpha\mu}{N} + 1 & \alpha^2 \frac{LL_0}{\sqrt{N}} + \alpha \left(L + \frac{\sigma L_0}{\sqrt{N}} \right) \\ \alpha^2 \frac{LL_0}{\sqrt{N}} + \alpha \left(L + \frac{\sigma L_0}{\sqrt{N}} \right) & \alpha^2 L^2 + 2\alpha\sigma^2 L + \sigma^2 \end{bmatrix}. \quad (5.21)$$

Darüber hinaus existiert eine positive Konstante α_A , die von den Parametern N, L, L_0, μ und $\sigma \in (0, 1)$ nach Gleichung (5.18) abhängig ist. Zusätzlich gilt für jedes $\alpha \in (0, \alpha_A)$ die Abschätzung $\sqrt{\rho(A(\alpha))} < 1$.

Der Beweis dieses Lemmas findet sich in Anhang B.2.

Hauptergebnis

Sei nun x^* das Nash-Gleichgewicht des Multiclusterspiels $\Gamma(H, \mathcal{X}, \{F^h\}_{h \in \mathcal{H}})$ und sei $\mathbf{x}^* = \mathbf{1}_N \otimes x^*$. Nach Lemma 5.2.1 ist \mathbf{x}^* dann die Lösung des durch das Schätzungsproblem erweiterten Multiclusterspiels aus Problem 6. Um nun die Konvergenz des beschriebenen Algorithmus zu beweisen, muss letztendlich gezeigt werden, dass die Distanz zwischen den Schätzungen \mathbf{x} und der Lösung \mathbf{x}^* zu null konvergiert, d. h. $\lim_{k \rightarrow \infty} \|\mathbf{x}(k) - \mathbf{x}^*\| = 0$. Um dies nachweisen zu können, muss

zusätzlich der kombinierte Vektor der Clustergradientenschätzungen $\mathbf{y}(k)$ zu dem Mittelwert der Gradienten der lokalen Kostenfunktionen konvergieren, d. h. $\lim_{k \rightarrow \infty} \|\mathbf{y}(k) - \bar{\mathbf{y}}(k)\| = 0$. Wie in Algorithmus 4 erkannt werden kann, hängen die Aktualisierungsgleichungen der Schätzungen $\mathbf{x}(k)$ von $\mathbf{y}(k)$ ab und vice versa. Wie im nachfolgenden Satz gezeigt wird, kann die Entwicklungen der beschriebenen Normen unter Algorithmus 4 als lineares Ungleichungssystem dargestellt werden, was die Beschreibung der Optimierungsdynamik entscheidend vereinfacht:

Satz 5.4.1 *Es gelten die Pauschalannahmen sowie die Annahmen 5.4.1, 5.4.2 und 5.4.3. Sei $\mathbf{x}^* = \mathbf{1}_N \otimes x^*$ und damit ein Fixpunkt der Abbildung $\mathbf{H}(\mathbf{x}) = \text{proj}_{\mathcal{X}} [\mathbf{x} - \alpha \mathbf{Q}^T \mathbf{M}(\mathbf{x})]$, siehe Lemma 5.2.1. Basierend auf Algorithmus 4 gilt das lineare Ungleichungssystem*

$$\tau(k+1) \leq A_\tau(\alpha)\tau(k) \quad (5.22)$$

für den Vektor

$$\tau(k) = \begin{bmatrix} \|\mathbf{x}(k) - \mathbf{x}^*\| \\ \|\mathbf{y}(k) - \bar{\mathbf{y}}(k)\| \end{bmatrix} \quad (5.23)$$

mit der positiven Matrix

$$A_\tau(\alpha) = \begin{bmatrix} \sqrt{\rho(A(\alpha))} & \alpha \\ a_{21}(\alpha) & \sigma_V + \alpha a_{22} \end{bmatrix}. \quad (5.24)$$

Darüber hinaus existiert eine positive Schrittweite $\bar{\alpha}$, die von den Problemparametern $N, L, L_0, L_1, \mu, \sigma_V$ und σ abhängt, sodass $\rho(A_\tau(\alpha)) < 1$ für jedes $\alpha \in (0, \bar{\alpha})$.

Der Beweis dieses Satzes findet sich in Anhang B.3.

Anmerkung 5.4.1 Die Matrix $A(\alpha)$ in Element $A_{\tau,11}$ ist in Gleichung (5.21) in Lemma 5.4.2 definiert. Ebenso findet sich im Beweis dieses Lemmas eine Spezifizierung der oberen Schranke α_A , sodass für $\alpha \in (0, \alpha_A)$ $\rho(A(\alpha)) < 1$ gilt.

Nach diesen Vorüberlegungen kann nun im nachfolgenden Theorem die Konvergenz des Algorithmus bewiesen werden.

Theorem 5.4.1 *Es gelten die Pauschalannahmen dieses Kapitels sowie die Annahmen 5.4.1, 5.4.2 und 5.4.3. Dann existiert ein eindeutiges Nash-Gleichgewicht x^* bezüglich des Problems 5 und $\mathbf{x}^* = \mathbf{1}_N \otimes x^*$ ist die Lösung des erweiterten Problems 6.*

Zusätzlich gilt für den Schätzungsvektor $\mathbf{x}(k)$ unter den Aktualisierungen nach Algorithmus 4, dass

$$\lim_{k \rightarrow \infty} \mathbf{x}(k) = \mathbf{x}^*, \quad (5.25)$$

wenn $0 < \alpha < \bar{\alpha}$, wobei in Satz 5.4.1 bewiesen ist, dass unter gegebenen Annahmen immer eine solche obere Schranke $\bar{\alpha} \neq 0$ existiert. Die Konvergenz erfolgt linear mit der Rate $O(\rho(A_\tau(\alpha))^k)$, wobei $\rho(A_\tau(\alpha)) < 1$ für $0 < \alpha < \bar{\alpha}$.

Beweis. Wenn das Spielmapping $M(x)$ aus Gleichung (5.3) die Pauschalannahme 5.2 erfüllt, d. h. es gleichmäßig monoton ist, dann hat die dem Problem 5 zugehörige Variationsungleichung $\text{VI}(\mathcal{X}, M(x))$ eine eindeutige Lösung [57]. Nach Lemma 5.2.1 existiert in diesem Fall auch eine Lösung für das erweiterte Multiclusterspiel, beschrieben in Problem 6, und diese Lösung ist eindeutig.

Nach Lemma 5.2.1 ist der Vektor \mathbf{x}^* ein Fixpunkt der Abbildung $\mathbf{H}(\mathbf{x}) = \text{proj}_{\mathcal{X}}[\mathbf{x} - \alpha \mathbf{Q}^T \mathbf{M}(\mathbf{x})]$. Daher kann der Satz 5.4.1 angewendet werden und die Entwicklung der für die Konvergenz relevanten Normen kann als das lineare Ungleichungssystem aus Gleichung (5.22) beschrieben werden. Zusätzlich gilt dann für den spektralen Radius $\rho(A_\tau(\alpha)) < 1$, wenn α in dem im Satz angegebenen Intervall gewählt wird, wodurch das lineare System stabil wird. Folglich gilt $\lim_{k \rightarrow \infty} \tau(k+1)/\tau(k) < \rho(A_\tau(\alpha)) < 1$, was die lineare Konvergenz mit Rate $O(\rho(A_\tau(\alpha))^k)$ zum gewünschten Zustand $\text{col}(0, 0)$ beweist. Damit ist die Beweisführung abgeschlossen. \square

Anmerkung 5.4.2 Nach Theorem 5.4.1 erreichen alle lokalen Schätzungen $x_i(k)$, $\forall i \in \mathcal{I}$ einen Konsensus, der die Lösung des Multiclusterspiels darstellt.

5.4.3 Fazit

Mit dem in diesem Abschnitt vorgestellten Algorithmus ist es möglich, beschränkte Multiclusterspiele mit einer linearen Konvergenzrate unter Verwendung von Gradienteninformationen zu lösen. Die Nebenbedingungen der jeweiligen Cluster werden mit einer Projektion in den Optimierungsprozess eingebunden. Hierfür müssen die Beschränkungsmengen allen Agenten des betreffenden Clusters bekannt sein. Der resultierende Algorithmus kann als eine Erweiterung einer Standardmethode zur Identifizierung von Nash-Gleichgewichten in nicht-kooperativen Spielen verstanden werden, die durch eine Gradientenschätzung bezüglich des Cluster-

gradienten erweitert wird. Diese ist notwendig, da auch das lokale, kooperative Optimum innerhalb der Cluster verteilt berechnet werden soll. Der Algorithmus konvergiert nachweislich unter Annahme von doppeltstochastischen Kommunikationsgraphen sowie der gleichmäßigen Monotonie des Spiel mappings. Der Stabilitätsbereich des Verfahrens durch die Wahl der Schrittweite eingeschränkt. Diese kann konstant und beliebig klein sein. Eine obere Schranke, die von Problemparametern abhängt, beschränkt zusätzlich den Stabilitätsbereich.

5.5 Simulationen: Einsatzplanproblem zwischen Microgrids

Während in den vorherigen Abschnitten die theoretische Konvergenz des Algorithmus im Vordergrund steht, soll in diesem Abschnitt die Anwendung auf ein spezifisches Multiclusterspiel, das Einsatzplanproblem zwischen Microgrids aus Kapitel 3, erfolgen.

Entsprechend beläuft sich das Ziel dieses Abschnittes auf die Konvergenzanalyse des Algorithmus 4 in Anwendung auf das Einsatzplanproblem aus Kapitel 3. Im Vordergrund steht hierbei die Konvergenzgeschwindigkeit in Bezug auf ein spezifisches Szenario sowie die Diskussion der resultierenden, bezüglich des Nash-Gleichgewichts optimalen Strategie der Microgrids.

5.5.1 Szenario und Setup

Die Problemformulierung des Einsatzplanproblems als Multiclusterspiel ist in Kapitel 3 in der Problemstellung (3.16) zu finden. Es wird empfohlen, vor der Lektüre dieses Abschnittes die Problemformulierung in Kapitel 3 zu lesen. Das in diesem Abschnitt behandelte Szenario des Einsatzplanproblems besteht aus insgesamt drei Microgrids, die jeweils fünf Busse umfassen, siehe Abbildung 5.2. Diese Busnetzwerke innerhalb der Microgrids sind an das 5-Bus-Beispielnetzwerk von Pandapower²⁹, siehe auch [119], angelehnt und übernehmen die Anordnung der Busse und Verbindungsleitungen sowie teilweise die Generatoren und deren Parametrierung aus diesem Beispielnetzwerk. In dieses System sind zusätzlich Speicher

²⁹Pandapower ist eine Python-basierte open-source Bibliothek, mit der Energienetze modelliert, analysiert und optimiert werden können, siehe [118].

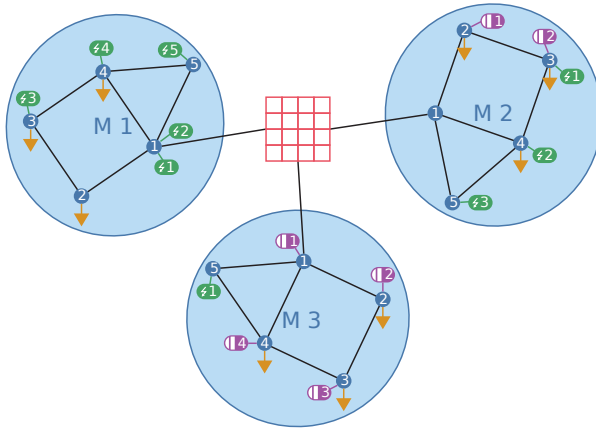


Abbildung 5.2: Setup für Simulationen: Drei Microgrids mit jeweils fünf Busen und drei Verbrauchern sowie jeweils gleichem Übertragungsnetz. Das erste Microgrid (M 1) besitzt fünf Generatoren, das zweite (M 2) drei Generatoren und zwei Speicher und das dritte (M 3) einen Generator und vier Speicher. Alle Microgrids sind mit dem roten Hauptnetz verbunden.

eingebraucht, die wie in Kapitel 3 vorgestellt modelliert sind. Die Parametrierung der Kostenfunktionen und der Nebenbedingungen des Multiclusterproblems sind dem Anhang B.4 zu entnehmen. Als Planungshorizont für das Einsatzplanproblem wird ein Tag angesetzt, der in 24 Zeitslots der Länge einer Stunde eingeteilt wird. Es gilt somit $t \in \mathcal{T} = \{1, \dots, T = 24\}$.

Die Verbraucher befinden sich bei allen drei Microgrids an den Bussen zwei, drei und vier. Der jeweilige Verbrauch pro Zeitslot kann der Tabelle B.2 aus dem Anhang B.4 entnommen werden und ist dem Tagesverbrauch von privaten Haushalten nachempfunden, siehe hierzu auch die Abbildung 5.3: In den frühen Morgenstunden ist der Verbrauch gering und steigt bis mittags leicht an. Danach fällt der Verbrauch wieder ab, bevor ab etwa 15 Uhr wieder ansteigt und schließlich um 19 Uhr der höchste Punkt der Verbrauchskurve erreicht ist. Im weiteren Verlauf des Abends verringert sich der Verbrauch wieder bis das Nachtniveau der frühen Morgenstunden erreicht ist.

In Tabelle 5.2 ist die Größe des Optimierungsproblems in Form der Anzahl der festzulegenden Optimierungsvariablen und der Anzahl der Beschränkungen, sowohl getrennt für jedes Microgrid, als auch aufsummiert für das Gesamtproblem angegeben. Microgrid 1 muss die Variablen sei-

Tabelle 5.2: Anzahl der Variablen und Beschränkungen des Optimierungsproblem, aufgeteilt auf die Microgrids (MG) sowie die Gesamtanzahl. Die Anzahl der Optimierungsvariablen entspricht der Problemstellung 5, das Schätzungsproblem aus 6 wird hier nicht betrachtet. Zur Berechnung der Beschränkungen sind jeweils einzelne (Un-)gleichungen verwendet.

	MG 1	MG 2	MG3	gesamt
# Variablen	288	288	288	864
# Beschränkungen	696	702	708	2106

ner fünf Generatoren festlegen, Microgrid 2 die seiner drei Generatoren und zwei Speicher und Microgrid 3 die Variablen seines Generators und die der vier Speicher. Zusätzlich muss jedes Microgrid jeweils die sechs Übertragungsleitungen sowie die bezogene Leistung vom Hauptnetz berücksichtigen. Diese Variablen müssen in jedem Zeitslot festgelegt werden. Entsprechend ergeben sich für jedes Microgrid 288 festzulegende Optimierungsvariablen und die Dimension der Gesamtstrategie des Multiclustersproblems beträgt 864. Für die Angabe der Anzahl der Beschränkungen wird jede Ungleichung und jede Gleichung als eine eigene Beschränkung angenommen.

Damit die Lösung dieser Problemstellung durch Algorithmus 4 effizient erfolgt, muss die Schrittweite $\alpha > 0$ möglichst optimal festgelegt werden. Das Ergebnis einer entsprechenden Rastersuche ergibt eine Schrittweite $\alpha = 3,8$.

Für die Kommunikation werden zufällige globale und lokale Kommunikationsgraphen erstellt und gewichtet, sodass die Annahmen 5.4.1 und 5.4.2 erfüllt sind. Die Kommunikationsgraphen sind in Anhang B.5 abgebildet.

Die Startwerte der Strategieschätzungen x_i der Agenten $i \in \mathcal{I}$ werden sämtlich mit dem Nullvektor initialisiert, während die Schätzungen y_i^h bezüglich des Gradienten des Clusters $h \in \mathcal{H}$ mit dem entsprechenden Gradienten $\nabla_h f_i^h$ der lokalen Kostenfunktion, ausgewertet an der Stelle $x_i(0)$, initialisiert werden. Dies entspricht der geforderten Initialisierung aus Algorithmus 4.

Die Clusterkostenfunktionen setzen sich aus den quadratischen Kostenfunktionen der Speicher, Generatoren und Leitungen zusammen. Dazu addiert werden die Kosten für den Bezug von Leistung aus dem Hauptnetz. Letztere sind bezüglich der clustereigenen Strategie $p^h(t)$, d. h. der im

Zeitslot t vom Cluster h bezogenen Leistungsmenge aus dem Hauptnetz, ebenfalls quadratisch. Somit sind die Clusterkostenfunktionen $F^h(x^h, x^{-h})$ differenzierbar und gleichmäßig konvex bezüglich der Clusterstrategie x^h . Die Nebenbedingungs Menge besteht aus linearen Gleichungs- und Ungleichungsbedingungen. Alle Optimierungsvariablen des Problems sind durch eine obere und untere Schranke begrenzt, wobei die Schranke selbst teil der Beschränkungs Menge ist. Folglich ist die Pauschalannahme 5.1 erfüllt. Darüber hinaus lässt sich zeigen, dass das Mapping (5.3) des Multiclusterspiels unter beschriebener Festlegung der lokalen Kostenfunktionen gleichmäßig monoton ist, d. h. die Definition 2.2.3 erfüllt. Entsprechend ist auch die Pauschalannahme 5.2 gegeben.

5.5.2 Analyse des Nash-Gleichgewichts des Einsatzplanspiels

Zunächst werden die Nash-Gleichgewichtsstrategien der Microgrids des betrachteten Szenarios bezüglich des Einsatzplanproblems besprochen, bevor detaillierter auf die Speicherstrategien des dritten Microgrids, welches einen Generator und vier Speicher beinhaltet, eingegangen wird.

In der Abbildung 5.3 sind die Bilanzen der Microgrids, d. h. die Balance zwischen produzierter, gespeicherter, gekaufter und verbrauchter Leistung, für jeden Zeitslot im Nash-Gleichgewicht als Balkendiagramm dargestellt. Die Grafik 5.3 (a) zeigt die Ergebnisse für das erste Microgrid, die Grafik 5.3 (b) die des zweiten Microgrids und Grafik 5.3 (c) stellt die Nash-Gleichgewichtsstrategie des dritten Microgrids dar. Zusätzlich ist die Preisfunktion des Hauptnetzes, die das Kopplungselement der Microgrids repräsentiert, in den Grafiken abgebildet.

Auf eine Darstellung der Strategien bezüglich der Übertragungsleitungen, mit Ausnahme der bezogenen Leistung von dem Hauptnetz, wird hier verzichtet. Grund hierfür ist, dass diese für die akkumulierte Betrachtung auf Microgridebene, wie der Fall in Abbildung 5.3, keine Rolle spielen: Die Leitungen verbinden ausschließlich Busse innerhalb der Microgrids, d. h. es fließt keine Leistung über das Bilanzgebiet der Microgrids hinaus oder hinein. Befinden sich somit alle Busse in Balance, wie durch die Balancebeschränkung gefordert, summieren sich die Leistungsflüsse bei Bilanzierung über ein Microgrid zu null. Eine Ausnahme stellt der Leistungsbezug von dem Hauptnetz dar, über den Leistung in das Microgrid eingebracht werden kann und der entsprechend Eingang in die Grafik findet. Für eine Betrachtung der Auslastung der Leitungen innerhalb von

Microgrid 3 sei auf die Abbildung 5.4 verwiesen.

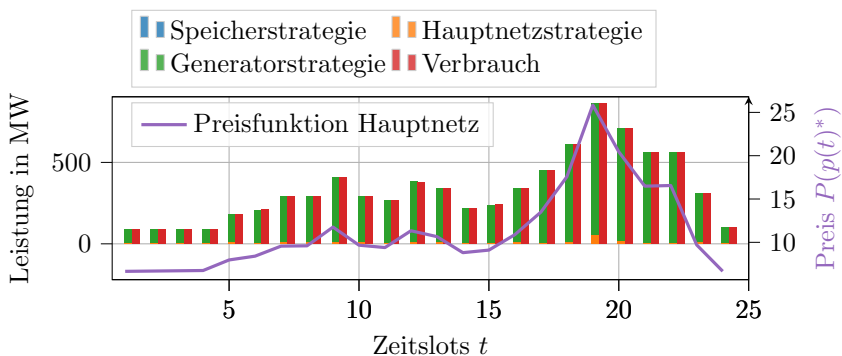
Microgrid 1, das mit insgesamt fünf Generatoren über die meisten dezentralen Produktionsmöglichkeiten im Netzwerk verfügt, wird im Nash-Gleichgewicht fast autark betrieben: Ausschließlich in den Zeiten von Verbrauchsspitzen wird Leistung vom Hauptnetz zugekauft und selbst dann nur wenig, weil auch die restlichen Microgrids zu diesen Zeitpunkten größere Leistungsmengen vom Hauptnetz einkaufen und der Preis pro Leistungseinheit entsprechend ansteigt.

Microgrid 2 enthält insgesamt drei Generatoren und zwei Speicher. Diese Speicher werden in Zeitslots, in denen der eigene Verbrauch niedrig ist, vor allem in $t \in \{1, \dots, 6\}$, durch Überproduktion der eigenen Generatoren aufgeladen und dann in Zeiten von hohem Verbrauch, vor allem in den Zeitslots $t \in \{18, 19, 20\}$, entladen, sodass weniger Leistung zu einem hohen Preis in diesen Zeitslots von dem Hauptnetz eingekauft werden muss. Zum Ende des Zeithorizonts werden die Speicher wieder geladen, um den gleichen Ladezustand wie zum Beginn des Zeithorizonts herzustellen.

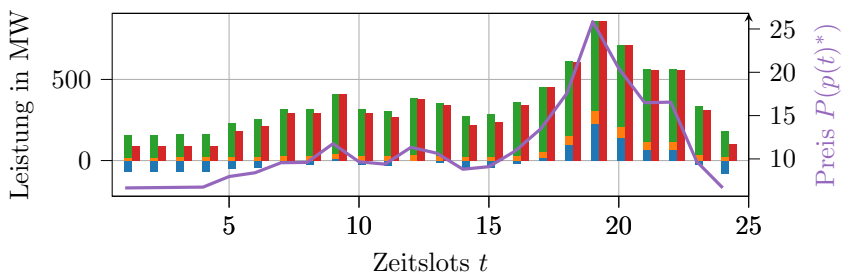
Ähnliches lässt sich für das Microgrid 3 beobachten, das über einen Generator und vier Speicher verfügt. Hierbei werden die Speicher allerdings stärker beansprucht. In der Abbildung 5.4a sind die Ladungen der einzelnen Speicher des Microgrids 3 dargestellt, die sich unter der entsprechenden Nash-Gleichgewichtsstrategie ergeben. Die Füllstände der Speicher sind in Prozent der Speicherkapazität angegeben und repräsentieren die Speicherladung zu Beginn des jeweiligen Zeitslots t . In schwarz ist für jeden Speicher zusätzlich die Ladungsänderung des jeweiligen Zeitslots t angegeben. Die entsprechende Ladung des nächsten Zeitslots $t + 1$ ergibt sich aus der Ladung des Zeitslots t plus oder minus der eingespeisten oder abgezogenen Leistung und unter Berücksichtigung der Leakageverluste. Deutlich erkannt werden können die Leakageverluste beim Speicher 3: In Zeitslot 16 wird der Speicher bis zum Kapazitätslimit aufgeladen, in Zeitslot 17 ist allerdings nicht mehr die komplette Ladung vorhanden. Es fällt auf, dass die Speicher 1 und 3 deutlich stärker beansprucht werden als die Speicher 2 und 4. Das liegt zum einen daran, dass die Kostenfunktionen der Speicher 1 und 3 kostengünstigere Parameter aufweisen als die Kostenfunktionen der Speicher 2 und 4, siehe hierzu die Tabelle B.2b in Anhang B.4. Darüber hinaus befindet sich vor allem der Speicher 3 an einer strategisch günstigen Position, da der den Speicher 3 verwendende Bus selbst einen Verbraucher besitzt und die beiden anderen Verbraucher über eine einzige Übertragungsleitung erreichbar sind. Letztere besitzen zwar selbst die Speicher 2 und 4, deren Verwendung allerdings, wie bereits

festgestellt, teurer ist.

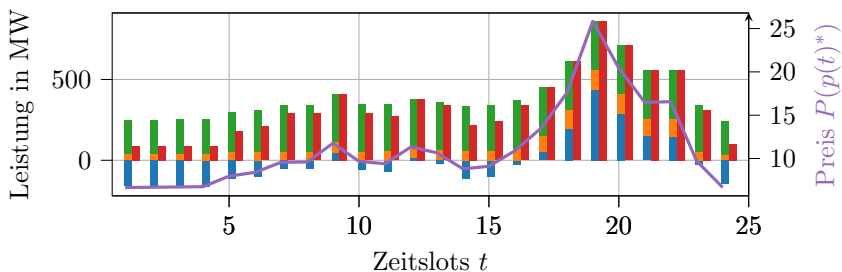
In Abbildung 5.4b sind die Strategien bezüglich der Übertragungsleitungen des Microgrids 3 dargestellt. Die Angaben sind in Prozent bezogen auf die jeweilige Kapazitätsbeschränkung. Wie erkannt werden kann befinden sich die bezüglich ihrer Kapazität am stärksten ausgelasteten Übertragungsleitungen zwischen den Bussen 1-4, 1-5 und 3-4. Bei einem Blick in die Tabelle B.1 des Anhangs B.4 fällt auf, dass diese Übertragungsleitungen die geringsten Kapazitäten besitzen. Zusätzlich befinden sich diese Leitungen an strategisch wichtigen Positionen. Über die Leitung 1-5 wird Leistung von dem Generator des Subnetzwerkes in das restlichen Netzwerk eingespeist. Die einzige weitere Übertragungsleitung für Leistung dieses Generators ist die Verbindung 4-5, die zwar nicht bis zur Kapazitätsgrenze, aber dennoch stark beansprucht wird. An Bus 1 befindet sich die Schnittstelle zum Hauptnetzwerk. Über die Leitung 1-4 kann somit, neben der von dem Generator über die Leitung 1-5 übertragenen Leistung, auch die Leistung, die vom Hauptnetz gekauft wird, an den Verbraucher am Bus 4 weitergegeben werden. Die Leitung 3-4 ermöglicht schließlich die Versorgung der Verbraucher bzw. der Speicher des Busses 3 mit von dem Generator des Netzwerkes erzeugten Leistung.



(a) Strategien des Microgrids 1.

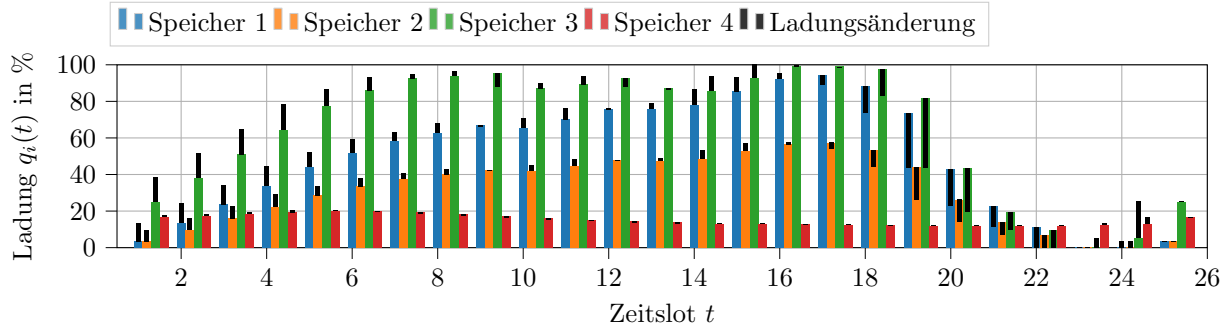


(b) Strategien des Microgrids 2 .

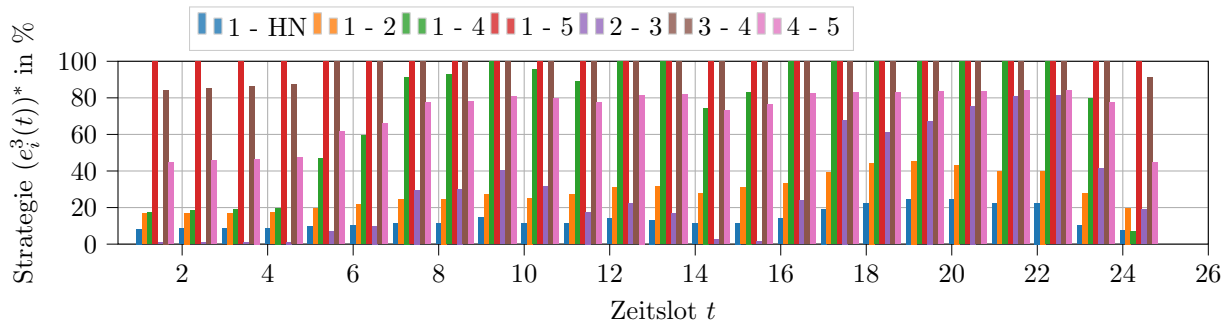


(c) Strategien des Microgrids 3.

Abbildung 5.3: Betrachtung der Bilanzen der Microgrids $h \in \{1, 2, 3\}$ im Nash-Gleichgewicht $(x^h)^*$ als Balkendiagramme über die Zeitslots $t \in \{1, \dots, 24\}$ zusammen mit der Preisfunktion $P(p(t)^*) = b_P \sum_{h=1}^H (p^h(t))^*$ des Hauptnetzes. Die Strategien der einzelnen Generatoren, Speicher und Verbrauch sind für jedes Microgrid aufsummiert als entsprechender Balken dargestellt.



(a) Ladung und Ladungsänderung der Speicher. Ladungsänderung entsprechen negativer Speicherstrategie $-(s_i^3(t))^*$.



(b) Leitungsstrategien für alle Leitungen innerhalb des Microgrids sowie Verbindung zum Hauptnetz (HN).

Abbildung 5.4: Speicher- und Leitungsstrategien für Microgrid 3 über die Zeitslots $t \in \{1, \dots, 24\}$ in Prozent der jeweiligen Speicher- bzw. Leitungskapazität.

In der Abbildung 5.5 sind die Generator-, Speicher- und Hauptnetzstrategien sowie der Verbrauch aller Microgrids aggregiert als Balkendiagramm über die Zeitslots t dargestellt. Es wird deutlich, dass die Microgrids hauptsächlich sich selbst mit ihren eigenen Generatoren versorgen, d. h. es wird deutlich mehr Leistung mit den dezentralen Generatoren produziert, als von dem Hauptnetz eingekauft wird. Mit steigender Leistungsnachfrage steigt auch die Nachfrage bezüglich des Leistungsbezugs vom Hauptnetz, da die Kosten der Generatoren mit der produzierte Leistungsmenge quadratisch ansteigen und es wirtschaftlicher ist, Leistung zu zukaufen. Durch den Einsatz von Speichern kann die Abhängigkeit vom Hauptnetz zusätzlich reduziert werden. In Zeiten geringen Verbrauchs wird billig Leistung produziert bzw. eingekauft und in die zur Verfügung stehenden Speicher eingespeist. In Zeiten hohen Verbrauchs werden die Speicher wieder geleert, sodass höhere Kosten durch große Produktionsmengen der dezentralen Generatoren bzw. ein Zukauf von Leistung aus dem Hauptnetz bei hohen Preisen vermieden werden können.

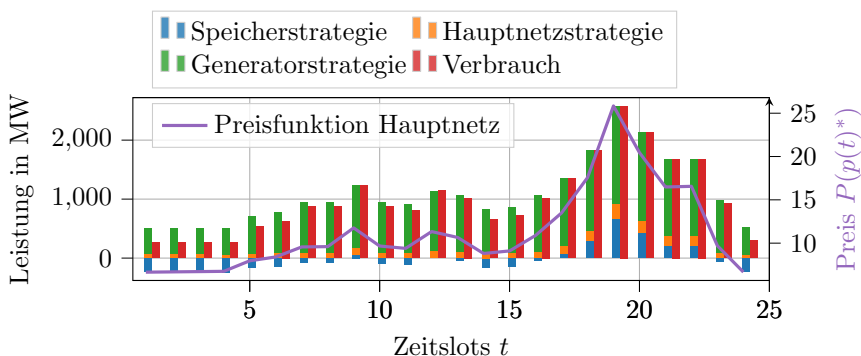


Abbildung 5.5: Aggregierte Strategien x^* aller Microgrids $h \in \{1, 2, 3\}$ über die Zeitslots $t \in \{1, \dots, 24\}$ zusammen mit der Preisfunktion $P(p(t)^*) = b_P \sum_{h=1}^H (p^h(t))^*$. Hierfür sind die Strategien bezüglich aller Generatoren, Speicher und des Leistungsbezuges aus dem Hauptnetz für die jeweiligen Zeitslots aufsummiert.

5.5.3 Konvergenz

In der Abbildung 5.6 sind Konvergenzverläufe dargestellt, die sich unter Anwendung des Algorithmus 4 auf das Einsatzplanproblem zwischen den

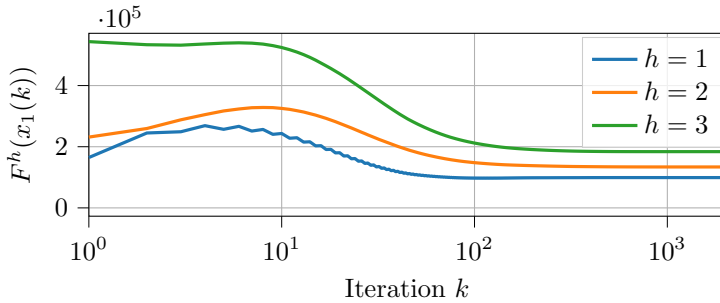
drei Microgrids ergeben.

In der Darstellung 5.6a ist die Konvergenz der Clusterkostenfunktionen F^h , d. h. der Summe der lokalen Kostenfunktionen der Agenten des Clusters, dargestellt. Ausgewertet werden diese Kostenfunktionen an der Schätzung des ersten Agenten, d. h. dem Bus 1, des jeweiligen Clusters bzw. Microgrids h zum jeweiligen Iterationsschritt k . In der Iteration $k = 0$ haben alle Kostenfunktionen den Wert 0, da jede Schätzung mit dem Nullvektor initialisiert wird. Durch diese Initialisierungswahl sind allerdings einige Nebenbedingungen verletzt, vor allem die Balancegleichungen an denjenigen Bussen, an denen die Verbraucher angeschlossen sind. Im ersten Iterationsschritt wird auf die Clusterbeschränkungsmenge projiziert, sodass alle Beschränkungen des jeweiligen Clusters eingehalten werden. Die durch die Projektion festgelegten Strategien sind allerdings keineswegs optimal, sondern müssen durch kontinuierlichen Informationsaustausch, Schätzung der Gradienten innerhalb der Cluster und die jeweilige Gradientenschritte in Richtung des Optimums angepasst werden. Ausgehend von der ersten Projektion erhöhen sich zunächst die Kostenfunktionen für die Microgrids 1 und 2 in Folge der Strategieänderungen der jeweils anderen Microgrids. Über den Iterationsverlauf lässt sich insgesamt aber ein deutliches Abfallen der Werte der Kostenfunktionen der Cluster bis zum Erreichen des Nash-Gleichgewichts feststellen. Wie aus der Grafik 5.6a erkannt werden kann, verändern sich die Kostenfunktionen ab der Iteration $k = 800$ nur noch in geringem Maße. Es ergeben sich die finalen Werte $F^1 = 99092,23$, $F^2 = 133830,55$ und $F^3 = 183937,17$, gerundet auf die zweite Nachkommastelle. Somit kann das Microgrid 1 seine Verbraucher am kostengünstigsten versorgen, während bei Reduktion der Anzahl der dezentralen Generatoren die Kosten steigen, siehe Microgrid 2 und 3. Durch die zusätzlichen Stromspeicher dieser Microgrids kann zwar vorhandene Leistung zwischen den Zeitslots transferiert werden, allerdings können hierdurch die fehlenden dezentralen Generatoren nicht ersetzt werden, auch da die Abhängigkeit vom Hauptnetz bei fehlenden dezentralen Produktionsmöglichkeiten zunimmt. Die beste Wahl ist ohne Frage ein Microgrid, bei dem eine ausreichend große Anzahl von dezentralen Generatoren und Speichern vorhanden ist.

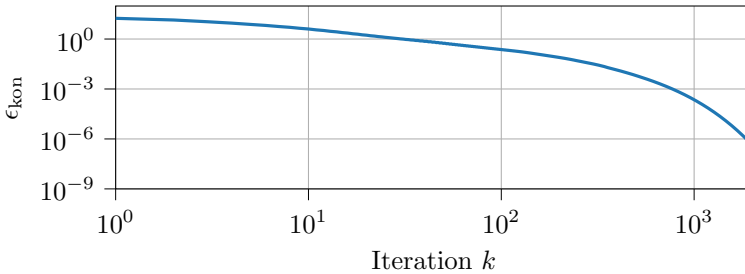
In der Abbildung 5.6b ist der Verlauf des Konsensusfehlers $\epsilon_{\text{kon}} = \sum_{i=1}^N \frac{\|x_1(k) - x_i(k)\|}{\|x_1(k)\|}$ über die Iterationen hinweg dargestellt. Hierbei handelt es sich um die relative, normierte Differenz zwischen der Schätzung des ersten Agenten des Clusters $h = 1$ und den Schätzungen der restlichen Agenten, aufsummiert über alle Agenten des Systems. Es kann erkannt werden, dass, ausgehend von einem hohen relativen Fehler von 17,52 nach

der ersten Iteration, der Konsensusfehler schnell abnimmt. Die Marke 10^{-3} wird nach etwa $k = 800$ Iterationen unterschritten, die Marke von 10^{-6} nach etwa $k = 1500$ Iterationen.

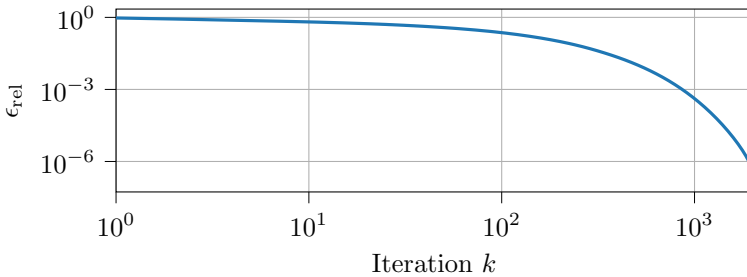
Ähnliche Beobachtungen können für den relativen Fehler $\epsilon_{\text{rel}} = \frac{\|x_1(k) - x^*\|}{\|x^*\|}$ bezüglich des Nash-Gleichgewichtvektors x^* gemacht werden, der ebenfalls bezüglich der Schätzung des ersten Agenten $i = 1$ des Systems berechnet wird. Da der Konsensusfehler zwischen den Schätzungen der Agenten in gleichem Maße abnimmt, kann vor allem gegen Ende des betrachteten Iterationsintervalls davon ausgegangen werden, dass sich ein unwesentlich abweichender Fehler für die Schätzungen der restlichen Agenten einstellt. Der relative Fehler unterschreitet die Genauigkeit von 10^{-3} nach etwa $k = 900$ Iterationen und nach $k = 1550$ wird ein Fehler von 10^{-6} unterschritten.



(a) Verlauf der Clusterkostenfunktionen $F^h(x_1(k))$.



(b) Verlauf des Konsensusfehlers $\epsilon_{\text{kon}} = \sum_{i=1}^N \frac{\|x_1(k) - x_i(k)\|}{\|x_1(k)\|}$.



(c) Verlauf des relativen Fehlers $\epsilon_{\text{rel}} = \frac{\|x_1(k) - x^*\|}{\|x^*\|}$.

Abbildung 5.6: Konvergenz der Clusterkostenfunktionen F^h , $h \in \mathcal{H}$, des Konsensusfehlers ϵ_{kon} und des relativen Fehlers ϵ_{rel} unter Algorithmus 4 in Anwendung auf das beschriebene Szenario. Logarithmische Skalierung der x-Achse für die Darstellung der Clusterkostenfunktion, doppelt-logarithmische Skalierung für die Darstellung der Fehlerkonvergenz.

5.6 Fazit

In Beitrag 5 dieses Kapitels wird der Algorithmus 4 vorgestellt, der auf dem *Gradient-Tracking*-Verfahren beruht, das durch eine Projektion erweitert wird, sodass nun beschränkte Multiclusterspiele gelöst werden können. Dabei wird von doppelt-stochastischen Gewichtung der Kommunikationsgraphen ausgegangen. Mit diesem Algorithmus ist es erstmals möglich, beschränkte Multiclustersprobleme gradientenbasiert unter Verwendung einer konstanten Schrittweite in einer zeitdiskreten Umgebung zu lösen. Es wird lineare Konvergenz des Algorithmus zum Nash-Gleichgewicht des Multiclusterspiels nachgewiesen und gezeigt, dass der Algorithmus für beliebig kleine Schrittweiten bzw. unter Einhaltung einer problemabhängigen oberen Schranke konvergiert, insofern die getroffenen Annahmen eingehalten werden.

Der Beitrag 6 umfasst schließlich die Anwendung des Algorithmus 4 auf das Einsatzplanproblem zwischen Microgrids aus dem Kapitel 3. Es wird gezeigt, dass der Algorithmus 4 ein Szenario von drei miteinander konkurrierenden Microgrids, die verschieden konfiguriert sind, effizient lösen kann. Das sich einstellende Nash-Gleichgewicht wird hierbei ausführlich besprochen.

6 Fazit

6.1 Zusammenfassung

In der vorliegenden Arbeit werden vier verschiedene Algorithmen vorgestellt, mit denen Optimierungsprobleme gelöst werden können, die auf einem Agentensystem verteilt formuliert sind. Drei dieser Algorithmen sind für die verteilte Lösung von kooperativen Problemstellungen und ein Algorithmus zur verteilten Lösung von Multiclusterspielen formuliert. Letztere bestehen aus einer hierarchischen Verkopplung von nicht-kooperativen und kooperativen Problemen. Alle Algorithmen bauen jeweils auf einem unbeschränkten Basisalgorithmus aus dem Stand der Forschung auf, der zur Berücksichtigung von Nebenbedingungen erweitert wurde. Die verwendeten Basisalgorithmen bestehen aus einem Konsensusprotokoll zur Angleichung von Agentenschätzungen aneinander sowie einem Aktualisierungsschritt unter Verwendung von Gradienteninformationen bezüglich der jeweiligen Agentenkostenfunktion.

In Algorithmus 1 aus Abschnitt 4.4 wird für jede lokale Ungleichungsbeschränkung eine Straffunktion definiert, die zu der jeweiligen Kostenfunktion hinzu addiert wird. Für das resultierende Verfahren zur Lösung von beschränkten Problemstellungen müssen die absinkende Schrittweitenfolge sowie die aufsteigende Strafparameterfolge zur optimalen Konvergenz aufeinander abgestimmt werden.

Um die aufwendige Wahl einer optimalen Strafparameterfolge zu vermeiden, werden in Algorithmus 2 aus Abschnitt 4.5 die lokalen Ungleichungsnebenbedingungen durch Lagrangemultiplikatoren zu den Agentenkostenfunktionen hinzugefügt sowie globale Nebenbedingungsmengen durch eine Projektion der lokalen Gradientenschritte berücksichtigt. Auch hier muss eine optimale, absinkende Schrittweitenfolge im Vorfeld definiert werden.

Um auch die notwendige Wahl der Schrittweite als absinkende Folge aufzuheben, wird in Algorithmus 3 aus Abschnitt 4.6 ein beschränktes Verfahren präsentiert, bei dem der Gradientenschritt über eine konstante Schrittweite gesteuert wird. Globale Nebenbedingungen werden über eine

Projektion berücksichtigt.

Mit Algorithmus 4 aus Abschnitt 5.4 kann schließlich eine beschränkte Multiclusterproblemstellung gelöst werden, bei der die Agenten, die sich innerhalb des selben Clusters befinden, zusammenarbeiten, während sich die Cluster selbst zueinander in einem nicht-kooperativen Spiel befinden. Für den gradientenbasierten Aktualisierungsschritt werden konstante Schrittweiten verwendet. Zusätzlich konvergiert der Algorithmus nachweislich linear zu seinem Optimum.

Alle in der vorliegenden Arbeit vorgestellten Algorithmen tragen zum Erreichen des in der Einleitung definierten **Ziels 1** bei, in dem die Erweiterung des Forschungsstandes bezüglich der Lösung von beschränkten, verteilten Problemstellungen gefordert wird. Der simulative Vergleich der kooperativen Algorithmen 1 - 3 in Anwendung auf das Einsatzplanproblem innerhalb der Microgrids aus Kapitel 3 hat gezeigt, dass Algorithmus 3 mit der konstanten Schrittweite deutlich geringere Konvergenzzeiten aufweist, als die Algorithmen 1 und 2. Eine Konvergenzvoraussetzung dieses Algorithmus ist allerdings die Annahme, dass die Nebenbedingungsmenge allen Agenten bekannt ist. Folglich können keine privaten, lokalen Nebenbedingungsfunktionen und -mengen existieren. Dies stellt eine Einschränkung der Anwendbarkeit des Algorithmus dar. Darüber hinaus werden für diesen Algorithmus statische, doppel-stochastische gewichtete Kommunikationsgraphen, und damit eine weniger allgemeine Kommunikation, als für die Algorithmen 1 und 2 gefordert, benötigt.

In Kapitel 3 wird ein Einsatzplanproblem zwischen Microgrids als Multiclusterproblem formuliert. Bei diesem suchen die aktiven Bestandteile der Microgrids, d. h. die Generatoren und Speicher, kooperativ nach einem optimalen Einsatzplan, um den Bedarf der Verbraucher im Microgrid zu decken. Die Microgrids selbst konkurrieren hierbei nicht-kooperativ bezüglich des Strompreises des Hauptnetzwerks, mit dem jedes Microgrid durch Übertragungsleitungen verbunden ist. Diese Problemstellung kann mit dem Algorithmus 4 gelöst werden. Die Simulationsergebnisse sind in Abschnitt 5.5 zu finden. Entsprechend ist hiermit das gesetzte **Ziel 2** erreicht.

Die Algorithmen 1 und 2 basieren auf dem Push-Sum-Konsensusprotokoll zur Steuerung des Kommunikationsprozesses. Ein Vorteil dieses Protokolls ist, dass für das Erreichen eines Konsensus eine spalten-stochastische Gewichtung der Kommunikationsmatrix bei zeitvarianten Verbindungen zwischen den Agenten ausreicht. Diese Eigen-

schaft bleibt auch für die verteilten Algorithmen bestehen, die auf Basis dieser Struktur entwickelt werden. Damit lassen sich die Algorithmen 1 und 2 einfach auf eine große Menge an Kommunikationsgraphen, seien es gerichtete oder ungerichtete, anpassen, was dem in der Einleitung gesetzten **Ziel 3** entspricht. Die Algorithmen 3 und 4 hingegen benötigen die weniger allgemeine doppelt-stochastische Gewichtung der Kommunikationsmatrix sowie statische Kommunikationskanäle zwischen den Agenten.

6.2 Ausblick

In den Simulationen hat sich Algorithmus 4 bzw. dessen auf die kooperative Problemstellung angepasste Sonderform in Algorithmus 3 als besonders effizient erwiesen, siehe die Abschnitte 4.7 und 5.5. Der unbeschränkte Basisalgorithmus dieser Verfahren ist das sogenannte *Gradient-Tracking*, siehe [28][95], bei dem neben der Konsensus- und Gradientenaktualisierung eine zusätzliche Iteration zur Schätzung des Gradienten des gesamten Agentensystems bzw. des Agentensystems innerhalb der Cluster benötigt wird. Berücksichtigt werden Nebenbedingungen hier durch die Projektion des Gradientenschrittes auf eine globale bzw. im Cluster verfügbare Nebenbedingungsmenge.

Nachteilig bei diesem Verfahren ist zum einen, dass aufgrund der Projektion keine lokalen Nebenbedingungen verarbeitet werden können, was entsprechend die Privatsphäre der Agenten einschränkt. Eine mögliche Erweiterung für diesen Algorithmus wäre, nach Vorbild des Algorithmus 2, die lokalen Ungleichungsbedingungen über einen dualen Ansatz zu berücksichtigen. Vorläufig durchgeführte Testsimulationen eines so erweiterten Algorithmus sehen vielversprechend aus. Der Konvergenzbeweis muss allerdings noch auf diese Erweiterung angepasst werden.

Zusätzlich konnte der Konvergenzbeweis für diesen Algorithmus bisher nur für doppelt-stochastisch gewichtete Kommunikationsgraphen durchgeführt werden. In der Veröffentlichung [38] ist bereits eine Erweiterung einer unbeschränkten Multiclusterproblemstellung für eine Kombination aus reihen- und spalten-stochastischer Gewichtung erfolgt. Allerdings steht auch hier der Schritt zur Verwendung dieser Gewichtung in dem Algorithmus 4 zur Lösung der beschränkten Problemstellung noch aus. Das Gleiche gilt für die Erweiterung zur Verarbeitung von zeitvarianten Graphen.

Bei der in der vorliegenden Arbeit betrachteten Klasse von Multiclusterspielen sind die Nebenbedingungen der Agenten unabhängig von den Aktionen der restlichen Spieler bzw. Agenten im Netzwerk. Entsprechend wird als Lösungskonzept das Nash-Gleichgewicht verfolgt. In vielen Problemstellungen ergibt sich allerdings eine Verkopplung der Nebenbedingungen der Spieler durch die zu wählende Strategie. Als Beispiel kann eine Erweiterung des in Kapitel 3 beschriebenen Einsatzplanproblems zwischen Microgrids betrachtet werden, bei der die maximal erzeugte Leistung des Hauptnetzes begrenzt ist. Folglich wird die Kaufstrategie eines Microgrids i durch die Strategien der restlichen Microgrids $-i$ beschränkt.

Das Lösungskonzept eines solchen Spiels ist beispielsweise das Verallgemeinerte Nash-Gleichgewicht. Lässt sich eine verkoppelnde Nebenbedingung separiert von den restlichen, unverkoppelten Beschränkungen der Spieler darstellen, kann der Ansatz aus [55] zur Berücksichtigung dieser Verkopplung verwendet werden. Hierbei wird ein zusätzlicher Spieler definiert, der für die Einhaltung der verkoppelnden Nebenbedingung zuständig ist. In dem Artikel ist unter gegebenen Annahmen bewiesen, dass das Nash-Gleichgewicht eines so erweiterten Spiels einem Verallgemeinerten Nash-Gleichgewicht entspricht. Eine mögliche Zusammenführung dieser Strategie und des in vorliegender Arbeit beschriebenen Lösungsalgorithmus 4 ist noch zu untersuchen. Durch diese Erweiterung ließe sich der resultierende Lösungsalgorithmus auf eine deutlich größere Klasse von Problemen anwenden. Möglich wird hierdurch auch der Aufbau eines peer-to-peer Marktes, auf dem die Microgrids untereinander über Leistungskontingente verhandeln können [17].

Bei allen in der vorliegenden Arbeit präsentierten Algorithmen wird genau ein Kommunikationsschritt und ein Gradientenschritt zur Aktualisierung der Aktions- bzw. Strategieschätzungen der Agenten durchgeführt. Diese strikte Abfolge kann, je nach Anwendung, zum einen den Konvergenzvorgang verlangsamen und zum anderen nicht energieeffizient sein. Letzteres kann beispielsweise dann der Fall sein, wenn die Kommunikation deutlich mehr Energie benötigt als die lokalen Aktualisierungen. In diesem Fall kann es von Vorteil sein, mehrere Gradientenschritte hintereinander auszuführen, bevor das Ergebnis der Aktualisierung mit den Nachbarn geteilt wird. Die Veröffentlichung [137] beschäftigt sich mit der Fragestellung, wie die Aufteilung von Kommunikation und Berechnung für einen verteilten Gradientenabstieg im optimalen Fall aussehen soll und welche theoretischen Garantien möglich sind. Die Ergebnisse sehen vielversprechend aus, sodass eine Anwendung auf die Algorithmen der vorliegenden Arbeit überdacht werden kann. Damit ließe

sich potentiell die Flexibilität und Anpassbarkeit der Algorithmen auf eine bestimmte Problemstellung erhöhen und somit sind auch potentiell geringere Konvergenzzeiten möglich.

Letztlich handelt es sich bei der gewählten Anwendung, dem Einsatzplanproblem zwischen Microgrids, um eine Abstraktion der Realität. Damit eine verteilte Lösung mit dem aktuellen Forschungsstand möglich ist, werden in der vorliegenden Arbeit sowie in relevanter Literatur verschiedene Vereinfachungen vorgenommen. Dazu gehören die ausschließlich durch maximale Kapazitäten modellierte Übertragungsleitungen, die als quadratisch angenommenen Kostenfunktionen der aktiven Bestandteile der Microgrids sowie die Abstraktion der in Realität komplexeren Marktsituation zwischen den Microgrids. Um realitätsnähere Problemstellungen zu lösen, müssen entsprechend die Annahmen der Algorithmen und damit die Anwendbarkeit bezüglich spezieller Problemstellungen verallgemeinert werden. Dazu gehört beispielsweise die Forderung der gleichmäßigen/strengen Konvexität der Kostenfunktionen bei kooperativen Problemen sowie die Annahme der gleichmäßigen Monotonie für den Beweis des Algorithmus bezüglich der Multiclusterspiele.

A Anhang zum Kapitel 4

A.1 Unterstützende Theoreme für Konvergenzbeweis von Algorithmus 1

Theorem A.1.1 ([18]) *Die Folgen $\{z_i(k)\}_k$, $i \in \mathcal{I}$, seien durch den Algorithmus 1 generiert und es gelten die Annahmen 4.4.1, 4.4.2, 4.4.3 und 4.4.6. Dann gelten folgende Aussagen:*

(a) $\lim_{k \rightarrow \infty} \|z_i(k+1) - \bar{x}(k)\| = 0$ für alle $i \in \mathcal{I}$.

(b) Wenn $\{b_k\}$ eine nicht ansteigende, positive, skalare Folge mit $\sum_{k=1}^{\infty} b_k \alpha_k \|\nabla f_i(z_i(k+1)) + r_k \nabla \Psi_i(z_i(k+1))\|_1 < \infty$ für alle $i \in \mathcal{I}$ ist, dann gilt $\sum_{k=0}^{\infty} b_k \|z_i(k+1) - \bar{x}(k)\| < \infty$ für alle $i \in \mathcal{I}$, wobei $\|\cdot\|_1$ die l^1 -Norm in \mathbb{R}^d ist.

Theorem A.1.2 [116] *Seien z_k, β_k, ξ_k und ζ_k nicht-negative Folgen, so dass*

$$z_{k+1} \leq z_k(1 + \beta_k) - \zeta_k + \xi_k.$$

Dann existiert der endliche Grenzwert $\lim_{k \rightarrow \infty} z_k$ und $\sum_{k=1}^{\infty} \zeta_k < \infty$ auf $\{\sum_{k=1}^{\infty} \beta_k < \infty, \sum_{k=1}^{\infty} \xi_k < \infty\}$.

A.2 Beweis von Lemma 4.4.1

Der nachfolgende Beweis wird aus [55] zitiert:

Beweis. Nach Annahme 4.4.4 gilt

$$\begin{aligned} \|\mathbf{q}(k, \bar{x}(k))\| &= \left\| \frac{1}{N} \left[\sum_{i=1}^N \nabla f_i(z_i(k+1)) - \sum_{i=1}^N \nabla f_i(\bar{x}(k)) \right] \right\| \\ &\leq \frac{l}{N} \sum_{i=1}^N \|z_i(k+1) - \bar{x}(k)\|. \end{aligned}$$

Sei $q(k) = \frac{l}{N} \sum_{i=1}^N \|z_i(k+1) - \bar{x}(k)\|$. Nun wird die Reihe

$$\sum_{k=1}^{\infty} b_k \alpha_k \|\nabla f_i(z_i(k+1)) + r_k \nabla \Psi_i(z_i(k+1))\|_1$$

betrachtet. Wenn $b_k = \alpha_k r_k^3$, dann gilt

$$\sum_{k=1}^{\infty} b_k \alpha_k \|\nabla f_i(z_i(k+1)) + r_k \nabla \Psi_i(z_i(k+1))\|_1 < \infty,$$

da $\nabla f_i(z_i(k+1))$ und $\nabla \Psi_i(z_i(k+1))$ beschränkt sind und die Annahme 4.4.6 für α_k und r_k gilt. Daher kann Theorem A.1.1 verwendet werden, um die Schlussfolgerung

$$\sum_{k=0}^{\infty} \alpha_k r_k^3 q(k) = l \sum_{k=0}^{\infty} \frac{\alpha_k r_k^3}{N} \sum_{i=1}^N \|z_i(k+1) - \bar{x}(k)\| < \infty$$

zu ziehen. Analog lässt sich $\sum_{k=0}^{\infty} \alpha_k r_k^3 p(k) < \infty$ zeigen. \square

A.3 Beweis von Theorem 4.4.1

Der nachfolgende Beweis wird aus [55] zitiert:

Beweis. Zunächst wird gezeigt, dass $\lim_{k \rightarrow \infty} \Psi(\bar{x}(k)) = 0$. Das bedeutet, dass sich für $k \rightarrow \infty$ alle Grenzwerte $\{\bar{x}(k)\}$ innerhalb der Beschränkungsmenge $\mathcal{C} = \{z \in \mathbb{R}^d \mid g_i(z) \leq 0, i \in \mathcal{I}\}$ befinden. Unter Verwendung des Mittelwertsatzes der Differentialrechnung, der Beziehung (4.15) und unter Verwendung der Notation

$$\nabla \tilde{F}(k, \bar{x}(k)) = \nabla F(\bar{x}(k)) + r_k \nabla \Psi(\bar{x}(k)) + \mathbf{q}(k, \bar{x}(k)) + r_k \mathbf{p}(k, \bar{x}(k)) \quad (\text{A.1})$$

ergibt sich für ein $\theta \in [0, 1]$ und $\tilde{x}(k) = \bar{x}(k) - \theta \alpha_k \nabla \tilde{F}(k, \bar{x}(k))$

$$\begin{aligned} \Psi(\bar{x}(k+1)) &= \Psi(\bar{x}(k)) - \alpha_k \langle \nabla \Psi(\bar{x}(k)), \nabla \tilde{F}(k, \bar{x}(k)) \rangle \\ &\quad + \alpha_k [\langle \nabla \Psi(\bar{x}(k)), \nabla \tilde{F}(k, \bar{x}(k)) \rangle - \langle \nabla \Psi(\tilde{x}(k)), \nabla \tilde{F}(k, \bar{x}(k)) \rangle]. \end{aligned} \quad (\text{A.2})$$

Nach Anmerkung 4.4.1 und der Cauchy-Schwarz Ungleichung, ergibt sich

$$\begin{aligned}
& \langle \nabla \Psi(\bar{x}(k)), \nabla \tilde{F}(k, \bar{x}(k)) \rangle - \langle \nabla \Psi(\tilde{x}(k)), \nabla \tilde{F}(k, \bar{x}(k)) \rangle \\
& \leq \| \nabla \tilde{F}(k, \bar{x}(k)) \| \| \nabla \Psi(\bar{x}(k)) - \nabla \Psi(\tilde{x}(k)) \| \\
& \leq \| \nabla \tilde{F}(k, \bar{x}(k)) \| mN \| \bar{x}(k) - \tilde{x}(k) \| \\
& \leq \theta mN \alpha_k \| \nabla \tilde{F}(k, \bar{x}(k)) \|^2,
\end{aligned} \tag{A.3}$$

wobei m der Konstanten aus Anmerkung 4.4.1 entspricht.

Unter Verwendung von Anmerkung 4.4.1, Lemma 4.4.1 und der Cauchy-Schwarz Ungleichung resultiert

$$\begin{aligned}
& \| \nabla \tilde{F}(k, \bar{x}(k)) \|^2 \leq \| \nabla F(\bar{x}(k)) \|^2 \\
& + r_k^2 \| \nabla \Psi(\bar{x}(k)) \|^2 + \| \mathbf{q}(k, \bar{x}(k)) \|^2 + r_k^2 \| \mathbf{p}(k, \bar{x}(k)) \|^2 \\
& + 2 \| \nabla F(\bar{x}(k)) \| r_k \| \nabla \Psi(\bar{x}(k)) \| + 2 \| \nabla F(\bar{x}(k)) \| \| \mathbf{q}(k, \bar{x}(k)) \| \\
& + 2 r_k \| \mathbf{q}(k, \bar{x}(k)) \| \| \mathbf{p}(k, \bar{x}(k)) \| + 2 r_k \| \nabla \Psi(\bar{x}(k)) \| \| \mathbf{q}(k, \bar{x}(k)) \| \\
& + 2 r_k^2 \| \nabla \Psi(\bar{x}(k)) \| \| \mathbf{p}(k, \bar{x}(k)) \| + 2 \| \nabla F(\bar{x}(k)) \| \| \mathbf{p}(k, \bar{x}(k)) \| \\
& \leq k_1 \left(\frac{1}{2} + q(k) + r_k p(k) \right) + k_2 \left(\frac{1}{2} r_k^2 + r_k q(k) + r_k^2 p(k) \right) \\
& + k_3 r_k + q^2(k) + r_k^2 p^2(k) + 2 r_k p(k) q(k) := h_0(k)
\end{aligned} \tag{A.4}$$

mit den positiven Konstanten k_1, k_2 , und k_3 . Nach der Definition von $\nabla \tilde{F}(k, \bar{x}(k))$ in (A.1) und der erneuten Anwendung der Cauchy-Schwarz Ungleichung ergibt sich schließlich für positive Konstanten k_4 und k_5

$$\begin{aligned}
& \langle \nabla \Psi(\bar{x}(k)), \nabla \tilde{F}(k, \bar{x}(k)) \rangle \geq \langle \nabla \Psi(\bar{x}(k)), \nabla F(\bar{x}(k)) \rangle + r_k \| \nabla \Psi(\bar{x}(k)) \|^2 \\
& - k_4 \| \mathbf{q}(k, \bar{x}(k)) \| - k_5 r_k \| \mathbf{p}(k, \bar{x}(k)) \| \\
& \geq \| \nabla \Psi(\bar{x}(k)) \| (- \| \nabla F(\bar{x}(k)) \| + r_k \| \nabla \Psi(\bar{x}(k)) \|) \\
& - k_4 q(k) - k_5 r_k p(k) \\
& \geq \| \nabla \Psi(\bar{x}(k)) \| \mathbb{1}_{\{ \| \nabla \Psi(\bar{x}(k)) \| \neq 0 \}} - k_4 q(k) - k_5 r_k p(k),
\end{aligned} \tag{A.5}$$

wobei die erste Ungleichung aus der Beschränkung von $\| \nabla \Psi \|$ und die letzte Ungleichung aus den Tatsachen, dass $0 < r_k \rightarrow \infty$ wenn $k \rightarrow \infty$ und der Beschränkung von $\| \nabla F \|$, folgt.

Durch Einsetzen von (A.3)-(A.5) in (A.2) folgt

$$\Psi(\bar{x}(k+1)) \leq \Psi(\bar{x}(k)) - \alpha_k \| \nabla \Psi(\bar{x}(k)) \| \mathbb{1}_{\{ \| \nabla \Psi(\bar{x}(k)) \| \neq 0 \}} + h(k),$$

wobei $h(k) = a_k^2(h_0(k) + k_4q(k) + k_5r_kp(k))$. Aufgrund der Wahl von α_k und r_k nach Annahme 4.4.6, Lemma 4.4.1 und Anmerkung 4.4.2 ergibt sich $\sum_{k=0}^{\infty} h(k) < \infty$. Aus dieser Erkenntnis und unter Verwendung des bekannten Theorems A.1.2 zu Folgen nicht-negativer Variablen erschließt sich, dass der Grenzwert $\lim_{k \rightarrow \infty} \Psi(\bar{x}(k))$ existiert, endlich ist und dass $\sum_{k=0}^{\infty} \alpha_k \|\nabla \Psi(\bar{x}(k))\|^2 < \infty$. Es folgt, dass $\liminf_{k \rightarrow \infty} \|\nabla \Psi(\bar{x}(k))\| = 0$, da nach Annahme 4.4.6 $\sum_{k=0}^{\infty} \alpha_k = \infty$ gilt. Dies impliziert die Existenz einer Teilfolge $\{k_t\} \subseteq \{k\}$, sodass $\lim_{t \rightarrow \infty} \|\nabla \Psi(\bar{x}(k_t))\| = 0$. Unter Annahme 4.4.5 folgt, dass $\|\nabla \Psi(z)\| = 0$ dann und nur dann gilt, wenn $\Psi(z) = 0$. Zusammen lässt sich hieraus schließen, dass $\lim_{t \rightarrow \infty} \Psi(\bar{x}(k_t)) = 0$. Es gilt somit

$$\lim_{k \rightarrow \infty} \Psi(\bar{x}(k)) = 0. \quad (\text{A.6})$$

Als nächstes wird der Zusammenhang $F(\bar{x}(k+1)) = F_k(\bar{x}(k+1)) - r_k \Psi(\bar{x}(k+1)) = F_{k+1}(\bar{x}(k+1)) - r_{k+1} \Psi(\bar{x}(k+1))$ verwendet um, unter Berücksichtigung von $\Psi(\bar{x}(k)) = o(1)$ wenn $k \rightarrow \infty$ (siehe (A.6)), $r_{k+1} - r_k = o(\alpha_k)$, $\alpha_k \rightarrow 0$ wenn $k \rightarrow \infty$ und unter Verwendung des Mittelwertsatzes der Differentialrechnung, auf

$$\begin{aligned} F_{k+1}(\bar{x}(k+1)) &= F_k(\bar{x}(k+1)) + (r_{k+1} - r_k) \Psi(\bar{x}(k+1)) \\ &= F_k(\bar{x}(k) - \alpha_k \nabla \tilde{F}(k, \bar{x}(k))) + o(\alpha_k) \\ &= F_k(\bar{x}(k)) - \alpha_k \langle \nabla F_k(\bar{x}(k)), \nabla \tilde{F}(k, \bar{x}(k)) \rangle + o(\alpha_k) \\ &\quad + \alpha_k \langle \nabla F_k(\bar{x}(k)) - \nabla F_k(x'(k)), \nabla \tilde{F}(k, \bar{x}(k)) \rangle, \end{aligned} \quad (\text{A.7})$$

zu schließen, wobei $x'(k) = \bar{x}(k) - \beta \alpha_k \nabla \tilde{F}(k, \bar{x}(k))$ für ein $\beta \in [0, 1]$ gilt. Nach Annahme 4.4.4 existiert ein $l_1 > 0$, sodass

$$\begin{aligned} &\langle \nabla F_k(\bar{x}(k)) - \nabla F_k(x'(k)), \nabla \tilde{F}(k, \bar{x}(k)) \rangle \\ &\leq l_1 \alpha_k r_k \|\nabla \tilde{F}(k, \bar{x}(k))\|^2 \leq l_1 \alpha_k r_k h_0(k), \end{aligned}$$

wobei die erste Ungleichung aus der Cauchy-Schwartz Ungleichung und die letzte Ungleichung aus (A.4) folgt. Folglich ergibt sich unter Verwendung der Cauchy-Schwartz Ungleichung und der Definition von $\nabla \tilde{F}(k, \bar{x}(k))$ in

(A.1) aus (A.7), dass

$$\begin{aligned}
F_{k+1}(\bar{x}(k+1)) &\leq F_k(\bar{x}(k)) \\
&\quad - \alpha_k \|\nabla F_k(\bar{x}(k))\|^2 - \alpha_k \langle \nabla F(\bar{x}(k)), \mathbf{q}(k, \bar{x}(k)) \rangle \\
&\quad - \alpha_k r_k \langle \nabla \Psi(\bar{x}(k)), \mathbf{q}(k, \bar{x}(k)) \rangle - \alpha_k r_k \langle \nabla F(\bar{x}(k)), \mathbf{p}(k, \bar{x}(k)) \rangle \\
&\quad - \alpha_k r_k^2 \langle \nabla \Psi(\bar{x}(k)), \mathbf{p}(k, \bar{x}(k)) \rangle + l_1 a_k^2 r_k h_0(k) + o(\alpha_k) \\
&\quad \leq F_k(\bar{x}(k)) - \alpha_k (\|\nabla F_k(\bar{x}(k))\|^2 + o(1)) \\
&\quad + \alpha_k q(k) \|\nabla F(\bar{x}(k))\| + \alpha_k r_k q(k) \|\nabla \Psi(\bar{x}(k))\| \\
&\quad + \alpha_k r_k p(k) \|\nabla F(\bar{x}(k))\| + \alpha_k r_k^2 p(k) \|\nabla \Psi(\bar{x}(k))\| + l_1 a_k^2 r_k h_0(k) \\
&\quad = F_k(\bar{x}(k)) - \alpha_k (\|\nabla F_k(\bar{x}(k))\|^2 + o(1)) + h_1(k),
\end{aligned}$$

wobei

$$\begin{aligned}
h_1(k) &= \alpha_k q(k) \|\nabla F(\bar{x}(k))\| + \alpha_k r_k q(k) \|\nabla \Psi(\bar{x}(k))\| \\
&\quad + \alpha_k r_k p(k) \|\nabla F(\bar{x}(k))\| + \alpha_k r_k^2 p(k) \|\nabla \Psi(\bar{x}(k))\| + l_1 a_k^2 r_k h_0(k).
\end{aligned}$$

Damit gilt $\sum_{k=0}^{\infty} h_1(k) < \infty$ nach Annahmen 4.4.3, 4.4.6 und Lemma 4.4.1 (mit Anmerkung 4.4.2). Aus Theorem A.1.2 kann nun auf die Aussage

$$F_k(\bar{x}(k)) \text{ hat einen Grenzwert f\u00fcr } k \rightarrow \infty. \quad (\text{A.8})$$

geschlossen werden.

Zus\u00e4tzlich folgt $\sum_{k=1}^{\infty} \alpha_k \|\nabla F_k(\bar{x}(k))\|^2 < \infty$, was nach Annahme 4.4.6 $\liminf_{k \rightarrow \infty} \|\nabla F_k(\bar{x}(k))\| = 0$ impliziert. Sei die Teilfolge $\{k_t\} \subseteq \{k\}$ gew\u00e4hlt, sodass $\lim_{t \rightarrow \infty} \|\nabla F_{k_t}(\bar{x}(k_t))\| = 0$. Aufgrund der Konvexit\u00e4t von F_k auf \mathbb{R}^d f\u00fcr alle k impliziert der letzte Grenzwert, dass $\lim_{t \rightarrow \infty} [F_{k_t}(\bar{x}(k_t)) - F_{k_t}^*] = 0$, wobei $F_{k_t}^* = \min_{x \in \mathbb{R}^d} F_{k_t}(x)$. Nach Satz 4.4.1 und $r_k \rightarrow \infty$ l\u00e4sst sich auf $\lim_{t \rightarrow \infty} F_{k_t}(\bar{x}(k_t)) - F^* = 0$ schließen, woraus sich zusammen mit (A.8) $\lim_{k \rightarrow \infty} F_k(\bar{x}(k)) = F^*$ ergibt und somit jeder Grenzwert von $\bar{x}(k)$ eine L\u00f6sung des Problems 3 f\u00fcr $\mathcal{X} \equiv \mathbb{R}$ ist. Unter Berufung auf Theorem A.1.1(a) schlie\u00dft sich die Beweisf\u00fchrung. \square

A.4 Beweis von Lemma 4.5.1

Beweis. 1. Teil:

Durch Einsetzen von (4.29) in (4.30) ergibt sich

$$\Phi(t, s) = \text{diag}(y(k+1))^{-1} W(k)W(k-1)\dots W(s)\text{diag}(y(s)). \quad (\text{A.9})$$

Sei $W'(k, s) = W(k)W(k-1)\dots W(s)$. Dann gilt für den Spezialfall $s = 0$, dass

$$\Phi(s, 0)\mathbf{1} = \text{diag}(y(k+1))^{-1} W'(k, 0)\mathbf{1} = \mathbf{1}, \quad (\text{A.10})$$

wobei ausgenutzt wird, dass $y(k+1) = W'(k, 0)\mathbf{1}$ nach Aktualisierungsgleichung (4.27a) gilt und somit der Vektor $y(k+1)$ aus den Reihensummen der Matrix $W'(k, 0)$ besteht. Entsprechend normiert sich $\text{diag}(y(k+1))^{-1} W'(k, 0)$ zu $\mathbf{1}$. Damit ist $\Phi(k, 0)$ für alle $k \geq 0$ nach Definition 2.5.4 reihen-stochastisch. Aus dieser Erkenntnis folgt direkt nach

$$\Phi(k, 0)\mathbf{1} = Q(k)\Phi(k-1, 0)\mathbf{1} = Q(k)\mathbf{1} = \mathbf{1} \quad (\text{A.11})$$

die reihen-stochastische Eigenschaft für alle $Q(k)$ mit $k \geq 0$. Da die Multiplikation von reihen-stochastischen Matrizen wieder eine reihen-stochastische Matrix ergibt und $\Phi(k, s)$ nach Gleichung (4.30) aus der Multiplikation der Matrizen $Q(k)\dots Q(s)$ besteht, ist auch $\Phi(k, s)$ für alle $s \leq k$ und $k \geq 0$ reihen-stochastisch.

2. Teil:

Für den zweiten Teil wird das Korollar 1 aus [138] verwendet. Unter Annahme 4.5.2 zusammen mit den Ergebnissen aus dem 1. Teil folgt demnach, dass

$$\lim_{k \rightarrow \infty} \Phi(k, s) = \mathbf{1}\phi_s^T \quad (\text{A.12})$$

mit einem positiven Vektor ϕ_s , da unter jedem endlichen s eine unendliche Folge von reihen-stochastischen Matrizen $Q(s), Q(s+1), \dots$ erzeugt wird und der Startzeitpunkt somit beliebig ist. \square

A.5 Beweis von Lemma 4.5.2

Beweis. Nach Lemma 4.5.1 ist $\mathbf{1}\phi_s^T$ der Grenzwert für $\Phi(k, s)$ für $k \rightarrow \infty$. Die Elemente $+\phi_{s,j} - \phi_{s,j}$ werden zur linken Seite des Ausdrucks addiert und anschließend die Dreiecksungleichung angewendet. Es resultiert

$$\left| \Phi(k, s)_{ij} - \frac{1}{N} \sum_{i=1}^N \Phi(k, s)_{ij} \right| \leq |\Phi(k, s)_{ij} - \phi_{s,j}| + \left| \frac{1}{N} \sum_{i=1}^N \Phi(k, s)_{ij} - \phi_{s,j} \right| \quad (\text{A.13})$$

Die Matrizen $\Phi(k, s)$ konvergieren für $k \rightarrow \infty$ auf statische Matrizen, deren Reihen gleich dem Vektor ϕ_s sind. Es gilt somit für

$\lim_{k \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \Phi(k, s)_{ij} = \frac{1}{N} \sum_{i=1}^N \mathbf{1} \phi_s^T = \phi_s^T$. Entsprechend konvergieren für $k \rightarrow \infty$ beide Summanden in der obigen Gleichung nach null und können daher durch entsprechende Konstanten $C\lambda^{k-1}$, mit $C > 0$, $\lambda \in (0, 1)$ nach Gleichung (4.31) abgeschätzt werden, siehe hierzu [139]. \square

A.6 Beweis von Lemma 4.5.3

Beweis. Nach Definition von $\epsilon_i^x(k)$ in Gleichung (4.32) gilt für die Norm

$$\|\epsilon_i^x(k)\| = \left\| \text{proj}_{\mathcal{X}} \left[z_i(k) - \alpha_k \frac{\nabla_z \mathcal{L}_i(z_i(k), \mu_i(k))}{y_i(k+1)} \right] - z_i(k) \right\| \quad (\text{A.14})$$

$$= \left\| \text{proj}_{\mathcal{X}} \left[z_i(k) - \alpha_k \frac{\nabla_z \mathcal{L}_i(z_i(k), \mu_i(k))}{y_i(k+1)} \right] - \text{proj}_{\mathcal{X}} [z_i(k)] \right\| \quad (\text{A.15})$$

$$\leq \alpha_k \left\| \frac{\nabla_z \mathcal{L}_i(z_i(k), \mu_i(k))}{y_i(k+1)} \right\| \leq \alpha_k \frac{L_x}{y_i(k+1)}. \quad (\text{A.16})$$

Die erste Gleichung gilt aufgrund der Tatsache, dass nach Aktualisierungsgleichung (4.27c) $x_i(k) \in \mathcal{X}$, $\forall i \in \mathcal{I}$ gilt. Aufgrund der Aktualisierung (4.27b) gilt $z_i(k) = \sum_{j=1}^N Q(k)_{ij} x_j(k)$, wobei nach Lemma 4.5.1 für alle Reihensummen $\sum_{j=1}^N Q(k)_{ij} = 1$, $\forall i \in \mathcal{I}$ gilt. Daher liegen alle $z_i(k)$ innerhalb der konvexen Hülle, die durch $x_i(k)$, $i \in \mathcal{I}$ aufgespannt wird. Folglich gilt $z_i(k) \in \mathcal{X}$, $\forall i \in \mathcal{I}$, da $x_i(k) \in \mathcal{X}$.

Die Ungleichung der nächsten Zeile folgt aus der Nicht-Ausdehnungseigenschaft der Projektion, siehe Gleichung (2.15). Die letzte Ungleichung folgt aus der Anmerkung 4.5.2 bezüglich der Beschränkung der Gradienten. Analog ergibt sich die Abschätzung der Norm $\|\epsilon_i^{\mu}(k)\|$. \square

A.7 Beweis von Lemma 4.5.4

Beweis. Teil 1):

Durch das Zurückführen der Gleichung (4.34a) auf den Initialisierungswert $x_i(0)$ kann $x_i(k)$ durch

$$\begin{aligned} x_i(k) &= \sum_{j=1}^N \Phi(k-1, 0)_{ij} x_j(0) + \sum_{s=0}^{k-2} \sum_{j=1}^N \Phi(k-1, s+1)_{ij} \epsilon_j^x(s) \\ &\quad + \epsilon_j^x(k-1). \end{aligned}$$

angegeben werden. Dieser Ausdruck wird in $\|x_i(k) - 1/N \sum_{i=1}^N x_i(k)\|$ eingesetzt und die Dreiecksungleichung angewendet. Unter Verwendung der Ergebnisse von Lemma 4.5.2 ergibt sich dann

$$\begin{aligned} \|x_i(k) - \bar{x}(k)\| &\leq C\lambda^k \sum_{j=1}^N \|x_j(0)\| + \sum_{s=0}^{k-2} C_s \lambda_s^{k-s-2} \sum_{j=1}^N \|\epsilon_j^x(s)\| \\ &\quad + \|\epsilon_i^x(k-1)\| + \frac{1}{N} \sum_{i=1}^N \|\epsilon_i^x(k-1)\|. \end{aligned}$$

Nun wird der Grenzwert für $k \rightarrow \infty$ für jeden Term einzeln betrachtet: Der Term $C\lambda^k \sum_{j=1}^N \|x_j(0)\|$ konvergiert im Grenzwert nach 0, da $\lambda \in (0, 1)$ und $\|x_j(0)\|, j \in \mathcal{I}$ als endlich angenommen werden kann.

Für die Angabe des Grenzwertes des Terms $\sum_{s=0}^{k-2} C_s \lambda_s^{k-s-2} \sum_{j=1}^N \|\epsilon_j^x(s)\|$ wird das Lemma 7 aus [100] verwendet. Dieses Lemma besagt, dass für eine positive, skalare Sequenz γ_k mit dem Grenzwert $\lim_{k \rightarrow \infty} \gamma_k = 0$ der Grenzwert

$$\lim_{k \rightarrow \infty} \sum_{s=0}^k \beta^{k-s} \gamma_s = 0$$

gilt, wobei $\beta \in (0, 1)$. Durch die Implikation (4.37) ist bekannt, dass unter geltender Annahme 4.5.4 der Grenzwert der positiven, skalaren Sequenz $\|\epsilon_j^x(k)\|$ für $k \rightarrow \infty$ für alle $j \in \mathcal{I}$ gleich 0 ist. Damit kann das Lemma 7 aus [100] auf den betrachteten Term angewendet werden, indem $k' = k - 2$ substituiert wird und folglich der Grenzwert

$$\lim_{k' \rightarrow \infty} \sum_{s=0}^{k'} C_s \lambda_s^{k'-s} \sum_{j=1}^N \|\epsilon_j^x(s)\| = 0$$

gilt. Nach Implikation (4.37) konvergieren die Terme $\|\epsilon_i^x(k-1)\|$ und $\frac{1}{N} \sum_{i=1}^N \|\epsilon_i^x(k-1)\|$ ebenfalls nach 0, was den Beweis für den ersten Teil abschließt.

Teil 2):

Es gilt

$$\begin{aligned} \sum_{k=0}^{\infty} \alpha_k \|x_i(k) - \bar{x}(k)\| &\leq \sum_{k=0}^{\infty} \alpha_k a_k + \sum_{k=0}^{\infty} \alpha_k b_k \\ &\quad + \sum_{k=0}^{\infty} \alpha_k \left(\|\epsilon_i^x(k-1)\| + \frac{1}{n} \sum_{i=1}^N \|\epsilon_i^x(k-1)\| \right) \end{aligned} \quad (\text{A.17})$$

mit den Folgen

$$a_k = \sum_{j=1}^N \left| \Phi(k-1, 0)_{ij} - \frac{1}{N} \sum_{i=1}^N \Phi(k-1, 0)_{ij} \right| \|x_j(0)\|$$

und

$$b_k = \sum_{s=0}^{k-2} \sum_{j=1}^N \left| \Phi(k-1, s+1)_{ij} - \frac{1}{N} \sum_{i=1}^N \Phi(k-1, s+1)_{ij} \right| \|\epsilon_j^x(s)\|.$$

Nach Lemma 4.5.2 kann die Folge a_k durch eine Folge a'_k beschränkt werden:

$$a_k \leq C\lambda^k \sum_{j=1}^N \|x_j(0)\| = a'_k.$$

Die unendliche Reihe $\sum_{k=0}^{\infty} a'_k$ konvergiert, da es sich um eine geometrische Reihe mit $0 < \lambda < 1$ handelt. Nach dem Majorantenkriterium konvergiert dann auch die Reihe $\sum_{k=0}^{\infty} a_k$, da $0 \leq a_k \leq a'_k, \forall k$ gilt.

Nach Annahme 4.5.4 ist α_k eine positive, nicht ansteigende Folge. Daher existiert ein $0 < K < \infty$, sodass $\alpha_k \leq K$. Da folglich

$$\sum_{k=0}^{\infty} \alpha_k a_k \leq K \sum_{k=0}^{\infty} a_k < \infty \quad (\text{A.18})$$

gilt, konvergiert die Reihe $\sum_{k=0}^{\infty} \alpha_k a_k$.

Die Folge b_k lässt sich unter Verwendung von Lemma 4.5.2 durch

$$b_k \leq \sum_{s=0}^{k-2} C_s \lambda_s^{k-s-2} \sum_{j=1}^N \|\epsilon_j^x(s)\|$$

abschätzen. Durch Summierung über den Zeitindex k und multiplizieren mit der Schrittweite α_k resultiert

$$\sum_{k=0}^{\infty} \alpha_k b_k \leq \sum_{k=0}^{\infty} \left(\sum_{s=0}^{k-2} C_s \lambda_s^{k-s-2} \sum_{j=1}^N \alpha_s \|\epsilon_j^x(s)\| \right),$$

wobei ausgenutzt wird, dass α_k nicht ansteigt und somit $\alpha_s \leq \alpha_k$ für $s \leq k$ gilt. Es sei nun

$$\gamma_s = \sum_{j=1}^N \alpha_s \|\epsilon_j^x(s)\|.$$

Durch Implikation (4.38) ist bekannt, dass $\sum_{k=1}^{\infty} \gamma_k < \infty$. Nach Lemma 7 aus [100] gilt

$$\sum_{k=0}^{\infty} \sum_{s=0}^{k-2} \lambda^{k-s-2} \gamma_s < \infty.$$

Durch die Anwendung auf den vorliegenden Fall ergibt sich

$$\sum_{k=0}^{\infty} \alpha_k b_k < \infty.$$

Schließlich gilt

$$\sum_{k=0}^{\infty} \alpha_k \|\epsilon_i^x(k-1)\| \leq \sum_{k=0}^{\infty} \alpha_{k-1} \|\epsilon_i^x(k-1)\| < \infty,$$

wobei wieder die Implikation (4.38) verwendet wird, zusammen mit der Annahme, dass die Folge α_k nicht ansteigt. Damit gilt

$$\sum_{k=0}^{\infty} \alpha_k \|x_i(k) - \bar{x}(k)\| < \infty,$$

was den Beweis abschließt. \square

A.8 Beweis von Satz 4.5.1

Beweis. Durch Einsetzen von $x_i(k+1)$ und $\mu_i(k+1)$ aus den Gleichungen (4.27c) und (4.27d), Anwendung der Nicht-Ausdehnungseigenschaft des Projektionsoperators, der Tatsache, dass $x^* \in \mathcal{X}$ und $\mu^* \in \mathcal{M}_i$ sowie durch das Expandieren der quadratischen Norm ergibt sich

$$y_i(k+1) \|x_i(k+1) - x^*\|^2 + \|\mu_i(k+1) - \mu_i^*\|^2 \quad (\text{A.19a})$$

$$\leq y_i(k+1) \|z_i(k) - x^*\|^2 + \|\mu_i(k) - \mu_i^*\|^2 \quad (\text{A.19b})$$

$$- 2\alpha_k \nabla_z \mathcal{L}_i(z_i(k), \mu_i(k))^T (z_i(k) - x^*) \quad (\text{A.19c})$$

$$+ 2\alpha_t \nabla_{\mu} \mathcal{L}_i(z_i(k), \mu_i(k))^T (\mu_i(k) - \mu_i^*) \quad (\text{A.19d})$$

$$+ \frac{\alpha_k^2}{y_i(k+1)} \|\nabla_z \mathcal{L}_i(z_i(k), \mu_i(k))\|^2 \quad (\text{A.19e})$$

$$+ \alpha_k^2 \|\nabla_{\mu} \mathcal{L}_i(z_i(k), \mu_i(k))\|^2. \quad (\text{A.19f})$$

Als Nächstes wird über die linke und rechte Seite von $i = 1$ bis N summiert und jede Zeile separat analysiert.

Unter Verwendung von Jensens Ungleichung³⁰ folgt aufgrund von $z(k) = Q(k)x(k)$ und der reihen-stochastischen Eigenschaft von $Q(k)$, festgestellt in Lemma 4.5.1, dass

$$\left\| \sum_{j=1}^N Q_{ij}(k)(x_j(k) - x^*) \right\|^2 \leq \sum_{j=1}^N Q_{ij}(k) \|x_j(k) - x^*\|^2.$$

Dementsprechend lässt sich für den ersten Summanden in Gleichung (A.19b) die obere Schranke

$$\begin{aligned} & \sum_{i=1}^N y_i(k+1) \sum_{j=1}^N \frac{W_{ij}(k)y_j(k)}{y_i(k+1)} \|x_j(k) - x^*\|^2 \\ & \leq \sum_{j=1}^N y_j(k) \|x_j(k) - x^*\|^2 \sum_{i=1}^N W_{ij}(k) = \sum_{i=1}^N y_i(k) \|x_i(k) - x^*\|^2 \end{aligned}$$

angeben, wobei in der ersten Zeile die Berechnung der Elemente von $Q(k)$ eingesetzt wird, und in der zweiten Zeile die Summen neu angeordnet sowie die spalten-stochastische Eigenschaft von $W(k)$ nach Annahme 4.5.3 verwendet wird. Damit gilt

$$\sum_{i=1}^N (\text{A.19b}) \leq \sum_{i=1}^N (y_i(k) \|x_i(k) - x^*\|^2 + \|\mu_i(k) - \mu_i^*\|^2).$$

Für Zeile (A.19c) ergibt sich aufgrund der Konvexität von $\mathcal{L}_i(z_i(k), \mu_i(k))$ für feste $\mu_i(k)$, dass

$$-\nabla_z \mathcal{L}_i(z_i(k), \mu_i(k))^T (z_i(k) - x^*) \leq \mathcal{L}_i(x^*, \mu_i(k)) - \mathcal{L}_i(z_i(k), \mu_i(k)).$$

In Zeile (A.19d) hängt $\mathcal{L}_i(z_i(k), \mu_i(k))$ affin von $\mu_i(k)$ bei festen $z_i(k)$ ab und daher gilt

$$\nabla_{\mu} \mathcal{L}_i(z_i(k), \mu_i(k))^T (\mu_i(k) - \mu_i^*) = \mathcal{L}_i(z_i(k), \mu_i(k)) - \mathcal{L}_i(z_i(k), \mu_i^*).$$

Durch kombinieren dieser Resultate, dem Hinzuaddieren von $+\mathcal{L}_i(x^*, \mu_i^*) - \mathcal{L}_i(x^*, \mu_i^*)$ und $+\mathcal{L}_i(\bar{x}(k), \mu_i^*) - \mathcal{L}_i(\bar{x}(k), \mu_i^*)$ sowie dem Summieren von

³⁰Jensen's Ungleichung besagt, dass für eine konvexe Funktion ϕ und positive Gewichte a_i mit $\sum_{i=1}^N a_i = 1$ die Ungleichung $\phi(\sum_{i=1}^N a_i x_i) \leq \sum_{i=1}^N a_i \phi(x_i)$ gilt.

$i = 1$ bis N , ergibt sich

$$\begin{aligned} \sum_{i=1}^N (\text{A.19c}) + (\text{A.19d}) &\leq -2\alpha_k (\mathcal{L}(\bar{x}(k), \mu^*) - \mathcal{L}(x^*, \mu^*)) \\ &\quad + \mathcal{L}(x^*, \mu^*) - \mathcal{L}(x^*, \mu(k)) \\ &\quad - 2\alpha_k \sum_{i=1}^N (\mathcal{L}_i(z_i(k), \mu_i^*) - \mathcal{L}_i(\bar{x}(k), \mu_i^*)). \end{aligned}$$

Die letzte Zeile in obigem Ausdruck lässt sich noch weiter abschätzen:

$$\begin{aligned} &- 2\alpha_k \sum_{i=1}^N (\mathcal{L}_i(z_i(k), \mu_i^*) - \mathcal{L}_i(\bar{x}(k), \mu_i^*)) \\ &\leq 2\alpha_k \sum_{i=1}^N |\mathcal{L}_i(z_i(k), \mu_i^*) - \mathcal{L}_i(\bar{x}(k), \mu_i^*)| \leq 2\alpha_k L_x \sum_{i=1}^N \|z_i(k) - \bar{x}(k)\| \\ &\leq 2\alpha_k L_x \sum_{i=1}^N \sum_{j=1}^N Q_{ij}(k) \|x_j(k) - \bar{x}(k)\| \leq 2\alpha_k L_x N \sum_{i=1}^N \|x_i(k) - \bar{x}(k)\|. \end{aligned}$$

Hierbei wird die L_x -Lipschitzstetigkeit der Lagrangefunktion für feste μ_i^* nach Anmerkung 4.5.3, die Dreiecksungleichung und die Tatsache, dass $0 \leq Q_{ij}(k) < 1$ gilt, verwendet. Damit folgt

$$\begin{aligned} \sum_{i=1}^N (\text{A.19c}) + (\text{A.19d}) &\leq -2\alpha_k (\mathcal{L}(\bar{x}(k), \mu^*) - \mathcal{L}(x^*, \mu^*)) \\ &\quad + \mathcal{L}(x^*, \mu^*) - \mathcal{L}(x^*, \mu(k)) + 2\alpha_k L_x N \sum_{i=1}^N \|x_i(k) - \bar{x}(k)\|. \end{aligned}$$

Es existiert ein k_0 , sodass für alle $k > k_0$ die Ungleichung $y_i(k) \geq \frac{Nw_i}{2} > \frac{w_i}{2}$ gilt, da $\lim_{k \rightarrow \infty} y_i(k) = w_i$. Unter Verwendung der Aussagen der Anmerkung 4.5.2 bezüglich der Beschränkung der Gradienten folgt

$$\sum_{i=1}^N (\text{A.19e}) + (\text{A.19f}) \leq L_x^2 \alpha_k^2 \sum_{i=1}^N \frac{1}{w_i} + \alpha_k^2 \sum_{i=1}^N L_{\mu_i}^2.$$

In Kombination der obigen Resultate schließt sich die Beweisführung. \square

A.9 Äquivalenz zwischen kooperativen Einsatzplanproblem unter Leistungsgleichung und -ungleichung

Es wird das folgende Problem betrachtet:

$$\begin{aligned} \min_x F(x) &= \min_x \sum_{i=1}^N f_i(x), \\ \text{s.t. } g_i - \sum_{j \in \bar{\mathcal{N}}_i^-} e_{ij} + \sum_{j \in \bar{\mathcal{N}}_i^+} e_{ji} &\geq v_i, \forall i \in \mathcal{I}, \\ \underline{g}_i &\leq g_i \leq \bar{g}_i, \forall i \in \mathcal{I}, \\ -\underline{e}_{ij} &\leq e_{ij} \leq \bar{e}_{ij}, \forall i, j \in \bar{\mathcal{E}}^{\text{B}}. \end{aligned}$$

mit $x = \text{col}(g, e)$, $g = \text{col}((g_i)_{i \in \mathcal{I}})$ und $e = \text{col}((e_{ij})_{(i,j) \in \bar{\mathcal{E}}^{\text{B}}})$ sowie

$$f_i(x) = f_i^{\text{G}}(g_i) + f_i^{\text{L}}(e), \quad (\text{A.20})$$

wobei $f_i^{\text{G}}(g_i) = a_{\text{G},i} g_i^2 + b_{\text{G},i} g_i + c_{\text{G},i}$ und $f_i^{\text{L}}(e) = \sum_{j \in \mathcal{N}_i} \frac{1}{2} a_{\text{L},i} e_{ij}^2$.

Im Folgenden wird unter Verwendung der KKT-Bedingungen gezeigt, dass im Optimum x^* immer die Gleichung $g_i^* - \sum_{j \in \bar{\mathcal{N}}_i^-} e_{ij}^* + \sum_{j \in \bar{\mathcal{N}}_i^+} e_{ji}^* = v_i$ erfüllt ist für alle $i \in \mathcal{I}$ und somit die obige Problemstellung äquivalent zu Problem (4.51) ist. Ohne Einschränkung der allgemeinen Gültigkeit wird davon ausgegangen, dass sich an jedem Bus ein einzelner Generator befindet.

Die Lagrangefunktion für das Problem lautet unter Verwendung der Multiplikatoren $\lambda = \text{col}(\bar{\lambda} \in \mathbb{R}^N, \underline{\lambda} \in \mathbb{R}^N)$, $\gamma = \text{col}(\bar{\gamma} \in \mathbb{R}^{|\bar{\mathcal{E}}^{\text{B}}|}, \underline{\gamma} \in \mathbb{R}^{|\bar{\mathcal{E}}^{\text{B}}|})$ und $\mu \in \mathbb{R}^N$:

$$\mathcal{L}(x, \lambda, \gamma, \mu) = F(x) + \underline{\lambda}^T (g - \bar{g}) + \bar{\lambda}^T (\bar{g} - g) \quad (\text{A.21})$$

$$+ \underline{\gamma}^T (-\bar{e} - e) + \bar{\gamma}^T (e - \bar{e}) \quad (\text{A.22})$$

$$+ \sum_{i=1}^N \mu_i (v_i - g_i + \sum_{j \in \bar{\mathcal{N}}_i^-} e_{ij} - \sum_{j \in \bar{\mathcal{N}}_i^+} e_{ji}). \quad (\text{A.23})$$

Da das Problem konvex ist und die Slater Bedingung, siehe Satz 2.3.1, erfüllt, ist $x^* = \text{col}(g^*, e^*)$ dann und nur dann eine Lösung des obigen Problems, wenn die folgenden KKT Bedingungen erfüllt sind:

$$\left. \frac{d\mathcal{L}(x, \lambda, \gamma, \mu)}{dg_i} \right|_{g^*} = \left. \frac{dF(x)}{dg_i} \right|_{g^*} - \underline{\lambda}_i^* + \bar{\lambda}_i^* - \mu_i^* = 0, \forall i \in \mathcal{I}, \quad (\text{A.24a})$$

$$\left. \frac{d\mathcal{L}(x, \lambda, \gamma, \mu)}{de_{ij}} \right|_{e_{ij}^*} = \left. \frac{dF(x)}{de_{ij}} \right|_{e_{ij}^*} - \underline{\gamma}_{(ij)}^* + \bar{\gamma}_{(ij)}^* + \mu_i^* = 0, \forall (i, j) \in \bar{\mathcal{E}}^B, \quad (\text{A.24b})$$

$$\underline{\lambda}_i^*(g_i - g_i^*) = 0, \underline{\lambda}_i^* \geq 0, \forall i \in \mathcal{I}, \quad (\text{A.24c})$$

$$\bar{\lambda}_i^*(g_i^* - \bar{g}_i) = 0, \bar{\lambda}_i^* \geq 0, \forall i \in \mathcal{I}, \quad (\text{A.24d})$$

$$\underline{\gamma}_{(ij)}^*(-\bar{e}_{ij} - e_{ij}^*), \underline{\gamma}_{(ij)}^* \geq 0, \forall (i, j) \in \bar{\mathcal{E}}^B, \quad (\text{A.24e})$$

$$\bar{\gamma}_{(ij)}^*(e_{ij}^* - \bar{e}_{ij}), \bar{\gamma}_{(ij)}^* \geq 0, \forall (i, j) \in \bar{\mathcal{E}}^B, \quad (\text{A.24f})$$

$$\mu_i^*(v_i^* - g_i^* + \sum_{j \in \bar{\mathcal{N}}_i^-} e_{ij}^* - \sum_{j \in \bar{\mathcal{N}}_i^+} e_{ji}^*) = 0, \mu_i^* \geq 0, \forall i \in \mathcal{I}. \quad (\text{A.24g})$$

Hierbei werden die Lagrangemultiplikatoren $\underline{\gamma}_{(ij)}, \bar{\gamma}_{(ij)}$ für die oberen und unteren Beschränkungen der Leitung e_{ij} verwendet, d. h. für die Kanten $(i, j) \in \bar{\mathcal{E}}^B$.

Gelte nun im Optimum die strikte Ungleichung $v_i - g_i^* + \sum_{j \in \bar{\mathcal{N}}_i^-} e_{ij}^* - \sum_{j \in \bar{\mathcal{N}}_i^+} e_{ji}^* < 0$. Es folgt nach (A.24g), dass $\mu_i^* = 0$.

Für die Generatoren kann davon ausgegangen werden, dass $\underline{g}_i \geq 0$ ist. Da $\left. \frac{dF(x)}{dg_i} \right|_{g^*} \geq 0$ im Bereich $[\underline{g}_i, \bar{g}_i]$ gilt, ist nach (A.24a) $\bar{\lambda}_i^* = 0$ und $\underline{\lambda}_i^* > 0$.

Es folgt aus (A.24c), dass $g_i^* = \underline{g}_i$ sein muss.

Da $\left. \frac{d\mathcal{L}(x, \lambda, \gamma, \mu)}{de_{ij}} \right|_{e_{ij}^*} < 0$ im Intervall $[-\bar{e}_{ij}, 0)$ gilt, muss in diesem Intervall

$\underline{\gamma}_{(ij)}^* = 0$ sein. Andererseits ist $\left. \frac{d\mathcal{L}(x, \lambda, \gamma, \mu)}{de_{ij}} \right|_{e_{ij}^*} > 0$ im Intervall $(0, \bar{e}_{ij}]$ und somit $\bar{\gamma}_{(ij)}^* = 0$. In Kombination ergibt sich $e_{ij}^* = 0$.

Werden diese Ergebnisse in die strikte Ungleichungsannahme eingesetzt ergibt sich

$$v_i < \underline{g}_i = g_i^*, \forall i \in \mathcal{I}. \quad (\text{A.25})$$

Dies widerspricht allerdings der Lösbarkeit des Problems, da der Bedarf unter den unteren Kapazitätsgrenzen der Generatoren liegen muss, was physikalisch nicht möglich ist. Entsprechend muss die Annahme $v_i - g_i^* + \sum_{j \in \bar{\mathcal{N}}_i^-} e_{ij}^* - \sum_{j \in \bar{\mathcal{N}}_i^+} e_{ji}^* < 0$ verworfen werden. Da zusätzlich $v_i - g_i^* +$

$\sum_{j \in \vec{N}_i^-} e_{ij}^* - \sum_{j \in \vec{N}_i^+} e_{ji}^* \leq 0$ eine Nebenbedingung des Problems darstellt, muss die Gleichungsbedingung im Optimum gelten.

A.10 Parameterkonfiguration für Simulationen

In den nachfolgenden Tabellen A.1, A.2, A.3 und A.4 sind die Parameterkonfigurationen für die Szenarien aufgeführt, für die im Abschnitt 4.7 die Simulationen durchgeführt werden. Aus Gründen der Übersichtlichkeit wird für das 30-Busssystem eine explizite Aufführung der Buskonfigurationen verzichtet und es sei auf die Darstellung des Netzwerkes in Abbildung 4.5c verwiesen.

Tabelle A.1: Parameter für 5-Busnetzwerk. Alle Werte bezogen auf die Einheit MW.

(a) Bus Konfiguration.			(b) Leitungsparameter.		
	Gen.	Verbrauch		$a_{L,i}$	\bar{e}_i
Bus 1	G 1, G 2	0	1 - 2	0.01	400
Bus 2	-	300	1 - 3	0.01	100
Bus 3	G 3	300	1 - 5	0.01	100
Bus 4	G 4	400	2 - 3	0.01	100
Bus 5	G 5	0	2 - 4	0.01	100
			4 - 5	0.01	240

(c) Generatorparameter.					
	$a_{G,i}$	$b_{G,i}$	$c_{G,i}$	\underline{g}_i	\bar{g}_i
Generator 1	0,2	6	0	0	40
Generator 2	0,047	7	0	0	170
Generator 3	0,038	10	0	0	520
Generator 4	0,145	11	0	0	200
Generator 5	0,008	5	0	0	600

Tabelle A.2: Parameter für 9-Busnetzwerk. Alle Werte bezogen auf die Einheit MW.

(a) Bus Konfiguration.			(b) Leitungsparameter.		
	Gen.	Verbrauch		$a_{L,i}$	\bar{e}_i
Bus 1	G 1	0	1 - 3	0.01	400
Bus 2	G 2	0	2 - 8	0.01	400
Bus 3	-	0	3 - 6	0.01	100
Bus 4	-	90	4 - 5	0.01	100
Bus 5	-	0	4 - 9	0.01	100
Bus 6	-	100	5 - 6	0.01	240
Bus 7	-	0	6 - 7	0.01	400
Bus 8	-	125	7 - 8	0.01	100
Bus 9	-	0	8 - 9	0.01	100

(c) Generatorparameter.					
	$a_{G,i}$	$b_{G,i}$	$c_{G,i}$	g_i	\bar{g}_i
Generator 1	0,11	5	0	10	300
Generator 2	0,085	1,2	0	10	270

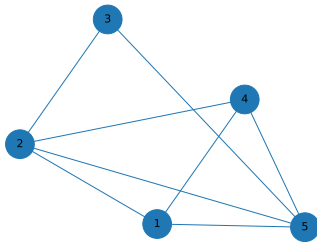
Tabelle A.3: Parameter für Leitungen des 30-Bussystems. Alle Werte bezogen auf die Einheit MW.

	$a_{L,i}$	\bar{e}_i			
			9 - 21	0.01	100
0 - 1	0.01	400	11 - 12	0.01	100
0 - 2	0.01	400	11 - 13	0.01	100
1 - 3	0.01	100	11 - 14	0.01	100
1 - 4	0.01	100	11 - 15	0.01	100
1 - 5	0.01	100	13 - 14	0.01	100
2 - 3	0.01	240	14 - 17	0.01	100
3 - 5	0.01	400	14 - 22	0.01	100
3 - 11	0.01	100	15 - 16	0.01	100
4 - 6	0.01	100	17 - 18	0.01	100
5 - 6	0.01	100	18 - 19	0.01	100
5 - 7	0.01	100	20 - 21	0.01	100
5 - 8	0.01	100	21 - 23	0.01	100
5 - 9	0.01	100	22 - 23	0.01	100
5 - 27	0.01	100	23 - 24	0.01	100
7 - 27	0.01	100	24 - 25	0.01	100
8 - 9	0.01	100	24 - 26	0.01	100
8 - 10	0.01	100	26 - 27	0.01	100
9 - 16	0.01	100	26 - 28	0.01	100
9 - 19	0.01	100	26 - 29	0.01	100
9 - 20	0.01	100	28 - 29	0.01	100

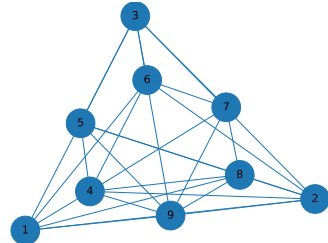
Tabelle A.4: Parameter für Generator des 30-Busnetzwerkes. Alle Werte bezogen auf die Einheit MW.

	$a_{G,i}$	$b_{G,i}$	$c_{G,i}$	\underline{g}_i	\bar{g}_i
Generator 1	0,02	2	0	0	80
Generator 2	0,0175	1,75	0	0	50
Generator 3	0,0625	1	0	0	55
Generator 4	0,00834	3,25	0	0	30
Generator 5	0,025	3	0	0	40

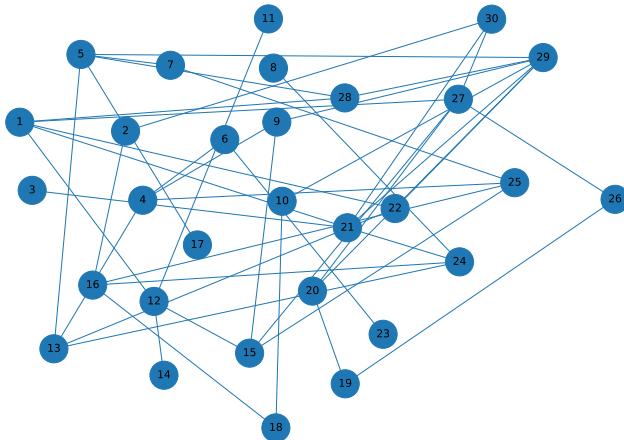
A.11 Kommunikationsgraphen für Busnetzwerke



(a) 5-Buskommunikationsgraph.



(b) 9-Buskommunikationsgraph.



(c) 30-Buskommunikationsgraph.

Abbildung A.1: Für die Simulationen verwendete zufällig erstellte Kommunikationsgraphen.

B Anhang zum Kapitel 5

B.1 Beweis von Lemma 5.2.1

Beweis. Teil 1) \iff 2): Die Äquivalenz zwischen den Lösungen der Variationsungleichung und dem konvexen Nash-Gleichgewichtsproblem ist unter den gegebenen Annahmen in Satz 2.4.1 aus dem Grundlagenabschnitt 2.4 belegt. Da x^* eine Lösung der Variationsungleichung VI(\mathcal{X}, M) darstellt, ist der Vektor ebenfalls ein Fixpunkt der Abbildung $H(x) = \text{proj}_{\mathcal{X}}[x - M(x)]$, d. h. $x^* = \text{proj}_{\mathcal{X}}[x^* - M(x^*)]$. Damit x_{opt}^h ein kooperatives Optimum innerhalb des Clusters h ist, muss für ein konstantes x^{-h} gelten, dass $x_{\text{opt}}^h = \text{proj}_{\mathcal{X}^h}[x_{\text{opt}}^h - \nabla_h F^h(x_{\text{opt}}^h, x^{-h})]$. Die Vektoren $x^{h,*}, \forall h \in \mathcal{H}$, sind wiederum Fixpunkte der entsprechenden Abbildungen, wenn der Vektor $x^* = \text{col}((x^{h,*})_{h \in \mathcal{H}})$ ein Fixpunkt der Abbildung $H(x)$ ist. Ist daher x^* ein Nash-Gleichgewicht des nicht-kooperativen Spiels zwischen den Clustern, wird das kooperative Optimum innerhalb der Cluster ebenfalls erreicht.

Teil 1) \iff 3):

Der Definition entsprechend erfüllt \mathbf{x}^* die Konsensusbedingung $(\mathbf{I}_{Nn} - \mathbf{W})\mathbf{x}^* = \mathbf{0}$, sodass die Schätzungen aller Agenten gleich dem Vektor x^* sind. \mathbf{x}^* ist ein Fixpunkt der Abbildung $\mathbf{H}(\mathbf{x})$ ist, wenn $\mathbf{H}(\mathbf{x}^*) - \mathbf{x}^* = \mathbf{0}$. Aus der Definition von \mathbf{Q} und der Abbildung $\mathbf{M}(\mathbf{x})$ sowie der Löschung von identischen und Nulleinträgen des Vektors $\mathbf{H}(\mathbf{x}^*) - \mathbf{x}^*$ folgt, dass sich die Fixpunktbedingung bezüglich der Abbildung $\mathbf{H}(\mathbf{x})$ an einem Konsensusvektor \mathbf{x}^* auf $\text{col}((\text{proj}_{\mathcal{X}^h}[x^{h,*} - F^h(x^*)])_{h \in \mathcal{H}}) - x^{h,*} = 0$ reduziert, was äquivalent zu $\text{proj}_{\mathcal{X}}[x^* - M(x^*)] - x^*$ ist. Daher ist $\mathbf{x}^* = \mathbf{1}_N \otimes x^*$ ein Fixpunkt der Abbildung $\mathbf{H}(\mathbf{x})$, wenn x^* ein Fixpunkt der Abbildung $\text{proj}_{\mathcal{X}}[x^* - M(x^*)]$ ist. Letztes gilt, wenn x^* eine Lösung der Variationsungleichung VI(Ω, M) ist.

Teil 2) \iff 4): Befinden sich die Schätzungen der Agenten, gespeichert im Vektor \mathbf{x} , in einem Konsensuszustand, d. h. es gilt, dass

$(\mathbf{I}_{Nn} - \mathbf{W})\mathbf{x} = 0$. Dann sind die Formulierungen der Probleme 5 und 6 zueinander identisch. Da $(\mathbf{I}_{Nn} - \mathbf{W})\mathbf{x}^* = 0$ und $x_i = x^*$ für $i = 1, \dots, N$ gilt, ist \mathbf{x}^* eine Lösung des erweiterten Problems 6 dann und nur dann, wenn x^* eine Lösung des zentralen Problems 5 ist.

Zusammengefasst folgt: 1) \iff 2) \iff 3) \iff 4).

□

B.2 Beweis Lemma 5.4.2

Beweis. Teil 1:

Zunächst wird die Herleitung der Ungleichung betrachtet. Diese basiert auf der Beweisstruktur von Theorem 1 aus [4] und wird auf die gegebene Problemstellung an den relevanten Stellen angepasst. In den nachfolgenden Überlegung wird auf eine explizite Darstellung des Zeitindex k verzichtet, sodass beispielsweise $\mathbf{x}(k) = \mathbf{x}$. Zusätzlich wird die Notation $\mathbf{M}\mathbf{x} = \mathbf{M}(\mathbf{x})$ verwendet, um die Übersichtlichkeit zu wahren.

Zunächst wird der Konsensusunterraum $\mathcal{E}_{||} := \{\mathbf{z} \in \mathbb{R}^{Nd} | \mathbf{z} = \mathbf{1}_N \otimes z, z \in \mathbb{R}^d\}$ und sein orthogonales Komplement $\mathcal{E}_{\perp} = \{\mathbf{z} \in \mathbb{R}^{Nd} | (\mathbf{1}_N \otimes \mathbf{I}_d)^T \mathbf{z} = 0\}$ definiert. Der Konsensusvektor $\mathbf{x}_{||}$ kann von dem Vektor \mathbf{x} durch dessen Projektion $\mathbf{x}_{||} = \text{proj}_{\mathcal{E}_{||}}[\mathbf{x}] = \frac{1}{N}(\mathbf{1}_N \mathbf{1}_N^T \otimes \mathbf{I}_n)\mathbf{x}$ abgeleitet werden und \mathbf{x}_{\perp} durch $\mathbf{x}_{\perp} = \text{proj}_{\mathcal{E}_{\perp}}[\mathbf{x}]$. Es gilt dabei $\mathbf{x}_{||}^T \mathbf{x}_{\perp} = 0$ und damit $\mathbf{x} = \mathbf{x}_{||} + \mathbf{x}_{\perp}$. Diese Beschreibung wird beispielsweise auch in den Veröffentlichungen [4][126] verwendet.

Unter Verwendung von $\hat{\mathbf{x}} = \mathbf{W}(\mathbf{x}_{||} + \mathbf{x}_{\perp}) = \hat{\mathbf{x}}_{||} + \hat{\mathbf{x}}_{\perp}$ können die folgen-

den Umformulierungen getätigt werden:

$$\begin{aligned} & \|\hat{\mathbf{x}} - \mathbf{x}^* - \alpha \mathbf{Q}^T \mathbf{M} \mathbf{x} + \alpha \mathbf{Q}^T \mathbf{M} \mathbf{x}^*\|^2 \\ &= \|\hat{\mathbf{x}}_{\parallel} + \hat{\mathbf{x}}_{\perp} - \mathbf{x}^* + \alpha \mathbf{Q}^T (-\mathbf{M} \mathbf{x} + \mathbf{M} \mathbf{x}^* - \mathbf{M} \mathbf{x}_{\parallel} + \mathbf{M} \mathbf{x}_{\parallel})\|^2 \\ &\leq \|\hat{\mathbf{x}}_{\parallel} - \mathbf{x}^*\|^2 + \|\hat{\mathbf{x}}_{\perp}\|^2 \\ &+ \alpha^2 \|\mathbf{Q}^T (\mathbf{M} \mathbf{x} - \mathbf{M} \mathbf{x}_{\parallel} + \mathbf{M} \mathbf{x}_{\parallel} - \mathbf{M} \mathbf{x}^*)\|^2 \end{aligned} \quad (\text{B.1})$$

$$+ 2(\hat{\mathbf{x}}_{\parallel} - \mathbf{x}^*)^T (\hat{\mathbf{x}}_{\perp}) \quad (\text{B.2})$$

$$- 2\alpha (\hat{\mathbf{x}}_{\parallel} - \mathbf{x}^*)^T \mathbf{Q}^T (\mathbf{M} \mathbf{x} - \mathbf{M} \mathbf{x}_{\parallel}) \quad (\text{B.3})$$

$$- 2\alpha (\hat{\mathbf{x}}_{\parallel} - \mathbf{x}^*)^T \mathbf{Q}^T (\mathbf{M} \mathbf{x}_{\parallel} - \mathbf{M} \mathbf{x}^*) \quad (\text{B.4})$$

$$- 2\alpha (\hat{\mathbf{x}}_{\perp})^T \mathbf{Q}^T (\mathbf{M} \mathbf{x} - \mathbf{M} \mathbf{x}_{\parallel}) \quad (\text{B.5})$$

$$- 2\alpha (\hat{\mathbf{x}}_{\perp})^T \mathbf{Q}^T (\mathbf{M} \mathbf{x}_{\parallel} - \mathbf{M} \mathbf{x}^*). \quad (\text{B.6})$$

Nachfolgend wird nun jede Zeile der obigen Ungleichung separat betrachtet.

Zeile (B.1):

Aufgrund der Lipschitzstetigkeit des erweiterten Mappings M nach Lemma 5.4.1 gilt

$$\|\mathbf{M} \mathbf{x} - \mathbf{M} \mathbf{x}_{\parallel}\|^2 \leq L^2 \|\mathbf{x} - \mathbf{x}_{\parallel}\|^2 = L^2 \|\mathbf{x}_{\perp}\|^2$$

und aufgrund der Lipschitzstetigkeit des ursprünglichen Spielmappings nach Pauschalannahme 5.2 gilt

$$\begin{aligned} & \|\mathbf{M} \mathbf{x}_{\parallel} - \mathbf{M} \mathbf{x}^*\|^2 \\ &= \sum_{h=1}^H N^h \frac{1}{N^h} \left\| \sum_{i=1}^{N^h} (\nabla_h f_i^h(x_{\parallel}) - \nabla_h f_i^h(x^*)) \right\|^2 \\ &= \|\mathbf{M} \mathbf{x}_{\parallel} - \mathbf{M} \mathbf{x}^*\|^2 \leq L_0^2 \|\mathbf{x}_{\parallel} - \mathbf{x}^*\|^2 \\ &= \frac{L_0^2}{N} \|\mathbf{x}_{\parallel} - \mathbf{x}^*\|^2. \end{aligned}$$

Unter Verwendung dieser Ergebnisse und durch $\|\mathbf{Q}^T\| = 1$ resultiert

$$\begin{aligned} & \alpha^2 \|\mathbf{Q}^T\|^2 \|\mathbf{M} \mathbf{x} - \mathbf{M} \mathbf{x}^* - \mathbf{M} \mathbf{x}_{\parallel} + \mathbf{M} \mathbf{x}_{\parallel}\|^2 \\ &\leq \alpha^2 \|\mathbf{Q}^T\|^2 (\|\mathbf{M} \mathbf{x} - \mathbf{M} \mathbf{x}_{\parallel}\| + \|\mathbf{M} \mathbf{x}_{\parallel} - \mathbf{M} \mathbf{x}^*\|)^2 \\ &\leq \alpha^2 (L^2 \|\mathbf{x}_{\perp}\|^2 + \frac{2LL_0}{\sqrt{N}} \|\mathbf{x}_{\perp}\| \|\mathbf{x}_{\parallel} - \mathbf{x}^*\| + \frac{L_0^2}{N} \|\mathbf{x}_{\parallel} - \mathbf{x}^*\|^2) \end{aligned}$$

Zeile (B.2):

Es gilt

$$2(\hat{\mathbf{x}}_{\parallel} - \mathbf{x}^*)^T(\hat{\mathbf{x}}_{\perp}) = 2\hat{\mathbf{x}}_{\parallel}^T \hat{\mathbf{x}}_{\perp} - 2(\mathbf{x}^*)^T \hat{\mathbf{x}}_{\perp} = 0$$

per Definition und da $\mathbf{x}^* \in \mathcal{E}_{\parallel}$.

Zeile (B.3):

Unter Verwendung von

$$\|\mathbf{Q}(\mathbf{x}_{\parallel} - \mathbf{x}^*)\|^2 = \sum_{h=1}^H N^h \left\| x_{\parallel}^h - (x^*)^h \right\|^2 \quad (\text{B.7})$$

$$\leq N \sum_{h=1}^H \left\| x_{\parallel}^h - (x^*)^h \right\|^2 \quad (\text{B.8})$$

$$= N \|\mathbf{x}_{\parallel} - \mathbf{x}^*\|^2 = \|\mathbf{x}_{\parallel} - \mathbf{x}^*\|^2, \quad (\text{B.9})$$

der Lipschitzstetigkeit des erweiterten Mappings und $\mathbf{W}\mathbf{x}_{\parallel} = \mathbf{x}_{\parallel}$ gilt

$$\begin{aligned} -2\alpha(\hat{\mathbf{x}}_{\parallel} - \mathbf{x}^*)^T \mathbf{Q}^T (\mathbf{M}\mathbf{x} - \mathbf{M}\mathbf{x}_{\parallel}) &= 2\alpha(\mathbf{Q}(\mathbf{x}_{\parallel} - \mathbf{x}^*))^T (\mathbf{M}\mathbf{x}_{\parallel} - \mathbf{M}\mathbf{x}) \\ &\leq 2\alpha \|\mathbf{Q}(\mathbf{x}_{\parallel} - \mathbf{x}^*)\| \|\mathbf{M}\mathbf{x}_{\parallel} - \mathbf{M}\mathbf{x}\| \\ &\leq 2\alpha L \|\mathbf{x}_{\parallel} - \mathbf{x}^*\| \|\mathbf{x}_{\perp}\|. \end{aligned}$$

Zeile (B.4):

Es gilt

$$\begin{aligned} &(\mathbf{Q}(\hat{\mathbf{x}}_{\parallel} - \mathbf{x}^*))^T (\mathbf{M}\mathbf{x}_{\parallel} - \mathbf{M}\mathbf{x}^*) \\ &= \sum_{h=1}^H \left[N^h (x_{\parallel}^h - (x^*)^h)^T \frac{1}{N^h} \sum_{i=1}^{N^h} (\nabla_h f_i^h(x_{\parallel}) - \nabla_h f_i^h(x^*)) \right] \\ &= \sum_{h=1}^H \left[(x_{\parallel}^h - (x^*)^h)^T \sum_{i=1}^{N^h} (\nabla_h f_i^h(x_{\parallel}) - \nabla_h f_i^h(x^*)) \right] \\ &= (x_{\parallel} - \mathbf{x}^*)^T (\mathbf{M}\mathbf{x}_{\parallel} - \mathbf{M}\mathbf{x}^*) \\ &\geq \mu \|\mathbf{x}_{\parallel} - \mathbf{x}^*\|^2 = \frac{\mu}{N} \|\mathbf{x}_{\parallel} - \mathbf{x}^*\|^2, \end{aligned}$$

unter Annahme von gleichmäßiger Monotonie des Mappings nach Pau-

schalannahme 5.2. Es folgt

$$\begin{aligned} -2\alpha(\hat{\mathbf{x}}_{\parallel} - \mathbf{x}^*)^T \mathbf{Q}^T (\mathbf{M}\mathbf{x}_{\parallel} - \mathbf{M}\mathbf{x}^*) &\leq -2\alpha\mu \|\mathbf{x}_{\parallel} - \mathbf{x}^*\|^2 \\ &= -2\alpha \frac{\mu}{N} \|\mathbf{x}_{\parallel} - \mathbf{x}^*\|^2. \end{aligned}$$

Zeile (B.5):

$$\begin{aligned} &-2\alpha(\hat{\mathbf{x}}_{\perp})^T \mathbf{Q}^T (\mathbf{M}\mathbf{x} - \mathbf{M}\mathbf{x}_{\parallel}) \\ &= 2\alpha(\mathbf{Q}\hat{\mathbf{x}}_{\perp})^T (F\mathbf{x}_{\parallel} - F\mathbf{x}) \leq 2\alpha \|\hat{\mathbf{x}}_{\perp}\| \|\mathbf{M}\mathbf{x}_{\parallel} - \mathbf{M}\mathbf{x}\| \\ &\leq 2\alpha \|\hat{\mathbf{x}}_{\perp}\| L \|\mathbf{x}_{\parallel} - \mathbf{x}\| = 2\alpha L \|\hat{\mathbf{x}}_{\perp}\|^2. \end{aligned}$$

Zeile (B.6):

$$\begin{aligned} &-2\alpha(\hat{\mathbf{x}}_{\perp})^T \mathbf{Q}^T (\mathbf{M}\mathbf{x}_{\parallel} - \mathbf{M}\mathbf{x}^*) \\ &= 2\alpha(\mathbf{Q}\hat{\mathbf{x}}_{\perp})^T (\mathbf{M}\mathbf{x}^* - \mathbf{M}\mathbf{x}_{\parallel}) \leq 2\alpha \frac{L_0}{\sqrt{N}} \|\hat{\mathbf{x}}_{\perp}\| \|\mathbf{x}_{\parallel} - \mathbf{x}^*\|. \end{aligned}$$

Zusammenfassung:

Mit $\|\hat{\mathbf{x}}_{\perp}\| = \|\mathbf{W}\mathbf{x}_{\perp}\| = \|\mathbf{W}\mathbf{x} - \mathbf{x}_{\parallel}\| \leq \sigma \|\mathbf{x}_{\perp}\|$ durch Annahme 5.4.2 und Gleichung (5.18) können die obigen Überlegungen zu einer einzigen Ungleichung zusammengefasst werden:

$$\begin{aligned} \|\hat{\mathbf{x}} - \mathbf{x}^* - \alpha \mathbf{Q}^T \mathbf{M}\mathbf{x} + \alpha \mathbf{Q}^T \mathbf{M}\mathbf{x}^*\|^2 &\leq \begin{bmatrix} \|\mathbf{x}_{\parallel} - \mathbf{x}^*\| \\ \|\mathbf{x}_{\perp}\| \end{bmatrix}^T A(\alpha) \begin{bmatrix} \|\mathbf{x}_{\parallel} - \mathbf{x}^*\| \\ \|\mathbf{x}_{\perp}\| \end{bmatrix} \\ &\leq \rho(A(\alpha)) \|\mathbf{x}(k) - \mathbf{x}^*\|^2, \end{aligned}$$

mit $A(\alpha)$ wie in Gleichung (5.21).

Teil 2

Um das Intervall für α zu finden, indem $\sqrt{\rho(A(\alpha))} < 1$ gilt, muss zunächst gezeigt werden, dass alle Elemente von $A(\alpha)$ nicht-negativ für $\alpha > 0$ sind. Anschließend kann die positive Definitheit der Matrix $A(\alpha)$ nachgewiesen werden, indem gezeigt wird, dass die Matrix diagonal dominant ist für $0 < \alpha < \alpha_{A,1}$. Hiernach wird unter Verwendung des Sylvesterkriteriums gezeigt, dass die Matrix $I_2 - A(\alpha)$ positiv definit für $0 < \alpha < \alpha_{A,2}$ ist. In Kombination ergibt sich die angegebene Abschätzung des spektralen Radius von $A(\alpha)$.

Aus Lemma 5.4.1 ist bekannt, dass $0 < \mu \leq L \leq L_0$ gilt. Zusätzlich

gilt $0 < \sigma < 1$, wie in Gleichung (5.18) verwendet. Es folgt, dass $A_{12}, A_{21}, A_{22} > 0$ für $\alpha > 0$. Die Diskriminante des quadratischen Polynoms bezüglich α in Element A_{11} ist $\frac{4\mu^2}{N^2} - \frac{4L_0^2}{N} < 0$ da $\mu \leq L_0$. Folglich sind die einzigen Wurzeln des Polynoms komplex und damit ist das Polynom positiv für alle $\alpha > 0$. Damit ist $A(\alpha)$ nicht-negativ für $\alpha > 0$. Um eine obere Schranke für α zu finden, sodass die Matrix diagonal dominant ist, d. h. $A_{11}, A_{22} > A_{12} = A_{21}$, werden zunächst die Faktoren

$$\begin{aligned} k_{11} &= L_0^2 \sqrt{N} - LL_0 N, \\ k_{12} &= -2\mu \sqrt{N} - \sigma L_0 N - LN \sqrt{N} < 0, \\ k_{13} &= N \sqrt{N} > 0, \\ k_{21} &= \sqrt{N} L^2 - LL_0, \\ k_{22} &= 2\sigma^2 L \sqrt{N} - L \sqrt{N} - \sigma L_0, \\ k_{23} &= \sigma^2 \sqrt{N} > 0. \end{aligned}$$

definiert. Durch die Untersuchung der Polynome $A_{11} - A_{12}$ und $A_{22} - A_{21}$ können die folgenden oberen Grenzen identifiziert werden:

$$\begin{aligned} \alpha_{M,1}^1 &= \left| \frac{-k_{12} - \sqrt{k_{12}^2 - 4k_{11}k_{13}}}{2k_{11}} \right|, \\ \alpha_{M,1}^2 &= \left| \frac{-k_{22} - \sqrt{k_{22}^2 - 4k_{21}k_{23}}}{2k_{21}} \right|, \\ \alpha_{M,1}^3 &= \frac{k_{13}}{|k_{12}|}, \alpha_{M,1}^4 = \frac{k_{23}}{|k_{22}|}, \alpha_{M,1}^5 = \sqrt{\frac{k_{23}}{|k_{21}|}}. \end{aligned}$$

Hierbei gilt, dass wenn $k_{21} = 0$, $k_{22} \neq 0$, da $\sigma \in (0, 1)$. Damit gilt, dass $\alpha_{A,1}^i > 0$, für $i = 1, \dots, 5$. Wenn $k_{11} = 0$, $k_{21} = 0$ oder $k_{22} = 0$, steigt die entsprechende Schranke ins Unendliche. Daher gilt mit einem $\alpha_{A,1} = \min\{\alpha_{A,1}^1, \alpha_{A,1}^2, \alpha_{A,1}^3, \alpha_{A,1}^4, \alpha_{A,1}^5\}$, dass durch die Wahl $0 < \alpha < \alpha_{A,1}$ die positive Definitheit der Matrix $A(\alpha)$ gewährleistet ist.

Als nächstes wird das Sylvester Kriterium angewendet um eine obere Schranke für α zu finden, die garantiert, dass die symmetrische Matrix $I_2 - A(\alpha)$ positiv definit ist: $1 - a_{11} > 0$ gilt, wenn $\alpha < \alpha_{A,2}^1 = \frac{2\mu}{L_0^2}$. Seien

die Faktoren

$$\begin{aligned} q_1 &= 2\mu L^2 - 2\sigma^2 LL_0^2 + 2L^2 L_0 \sqrt{N} + 2\sigma LL_0^2, \\ q_2 &= 4\mu\sigma^2 L + L_0^2 + NL^2 + 2\sigma LL_0 \sqrt{N}, \\ q_3 &= 2\mu(1 - \sigma^2), \end{aligned}$$

definiert, für die $q_1, q_2, q_3 > 0$ gilt, da $\mu, \sigma, N, L, L_0 > 0$ und $\sigma^2 < \sigma$ mit $\sigma \in (0, 1)$. Damit gilt $\det(I_2 - A(\alpha)) > 0$, wenn $q_1\alpha^2 + q_2\alpha - q_3 < 0$, was wiederum für $0 < \alpha < \alpha_{A,2}^2 = \frac{-q_2 + \sqrt{q_2^2 + 4q_1q_3}}{2q_1} > 0$ gilt.

In Kombination der obigen Resultate ergibt sich, dass $\sqrt{\rho(A(\alpha))} < 1$, wenn $0 < \alpha < \alpha_A = \min\{\alpha_{A,1}, \alpha_{A,2}\}$ mit $\alpha_{A,2} = \min\{\alpha_{A,2}^1, \alpha_{A,2}^2\}$, was die Beweisführung schließt. \square

B.3 Beweis von Satz 5.4.1

Beweis. Der Beweis wird in zwei Teile aufgeteilt. Im ersten wird gezeigt, dass die Aktualisierungsgleichungen des Algorithmus 4 bzw. in vektorisierter Form (5.16) sich als lineares Ungleichungssystem darstellen lassen und es wird auf die Struktur der Matrix $A_\tau(\alpha)$ eingegangen. Im zweiten Teil wird der spektrale Radius der Matrix in Abhängigkeit von α untersucht und gezeigt, dass eine obere Schranke $\bar{\alpha}$ existiert, unter welcher der Algorithmus konvergiert.

Teil 1 - Lineares Ungleichungssystem Unter Verwendung der Fixpunktgleichung $\mathbf{x}^* = \text{proj}_{\mathcal{X}} [\mathbf{x}^* - \alpha \mathbf{Q}^T \mathbf{M}(\mathbf{x}^*)]$ und der Gleichung (5.16b) ergibt sich:

$$\begin{aligned} & \|\mathbf{x}(k+1) - \mathbf{x}^*\| \\ &= \|\text{proj}_{\mathcal{X}} [\hat{\mathbf{x}}(k) - \alpha \mathbf{Q}^T \mathbf{y}(k)] - \text{proj}_{\mathcal{X}} [\mathbf{x}^* - \alpha \mathbf{Q}^T \mathbf{M}(\mathbf{x}^*)]\| \\ &\leq \|\hat{\mathbf{x}}(k) - \alpha \mathbf{Q}^T \mathbf{y}(k) - \mathbf{x}^* + \alpha \mathbf{Q}^T \mathbf{M}(\mathbf{x}^*)\| \\ &\leq \|\hat{\mathbf{x}}(k) - \mathbf{x}^* - \alpha \mathbf{Q}^T \mathbf{M}(\mathbf{x}(k)) + \alpha \mathbf{Q}^T \mathbf{M}(\mathbf{x}^*)\| \\ &\quad + \|\alpha \mathbf{Q}^T \mathbf{M}(\mathbf{x}(k)) - \alpha \mathbf{Q}^T \mathbf{y}(k)\| \end{aligned}$$

Für die erste Ungleichung wird die Nicht-Expansivitätseigenschaft des Projektionsoperators verwendet, siehe Gleichung (2.15). Unter Verwendung

der Resultate des Lemmas 5.4.2 und von

$$\begin{aligned}
 & \left\| \alpha \mathbf{Q}^T \mathbf{M}(\mathbf{x}(k)) - \alpha \mathbf{Q}^T \mathbf{y}(k) \right\|^2 \\
 & \leq \alpha^2 \left\| \mathbf{Q}^T \right\|^2 \left\| \mathbf{M}(\mathbf{x}(k)) - \mathbf{y}(k) \right\|^2 \\
 & \leq \alpha^2 \left\| \mathbf{Q}^T \right\|^2 \sum_{h=1}^H \left\| \mathbf{1}_{N^h} \otimes \frac{1}{N^h} \sum_{i=1}^{N^h} \nabla_h f_i^h(x_i(k)) - \mathbf{y}^h(k) \right\|^2 \\
 & = \alpha^2 \sum_{h=1}^H \left\| \mathbf{y}^h(k) - \bar{\mathbf{y}}^h(k) \right\|^2 = \alpha^2 \left\| \mathbf{y}(k) - \bar{\mathbf{y}}(k) \right\|^2,
 \end{aligned}$$

wobei die Definition von $\bar{\mathbf{y}}^h(k)$ verwendet wird, ergibt sich

$$\begin{aligned}
 \left\| \mathbf{x}(k+1) - \mathbf{x}^* \right\|^2 & \leq \sqrt{\rho(A(\alpha))} \left\| \mathbf{x}(k) - \mathbf{x}^* \right\| \\
 & \quad + \alpha \left\| \mathbf{y}(k) - \bar{\mathbf{y}}(k) \right\|.
 \end{aligned}$$

Als nächstes wird die Norm $\left\| \mathbf{y}(k+1) - \bar{\mathbf{y}}(k+1) \right\|$ und die Aktualisierungsgleichung (5.16c) betrachtet:

$$\begin{aligned}
 \left\| \mathbf{y}(k+1) - \bar{\mathbf{y}}(k+1) \right\| & = \left\| \mathbf{V} \mathbf{y}(k) + \mathbf{G}(k+1) - \mathbf{G}(k) \right. \\
 & \quad \left. - \mathbf{R}(\mathbf{y}(k) + \mathbf{G}(k+1) - \mathbf{G}(k)) \right\| \\
 & \leq \left\| \mathbf{V} \mathbf{y}(k) - \bar{\mathbf{y}}(k) \right\| + \left\| \mathbf{I} - \mathbf{R} \right\| \left\| \mathbf{G}(k+1) - \mathbf{G}(k) \right\|
 \end{aligned}$$

Hierbei wird die doppelt-stochastische Gewichtung von V^h , $\forall h \in \mathcal{H}$, aus Annahme 5.4.1 ausgenutzt bzw. deren Folge, dass $\mathbf{R} \mathbf{V} = \mathbf{R}$, die sich unter Verwendung der Eigenschaft $AC \otimes BD = (A \otimes B)(C \otimes D)$ des Kroneckerprodukts ergibt, sodass

$$\begin{aligned}
 \mathbf{R}^h \mathbf{V}^h & = \left(\frac{1}{N^h} \mathbf{1}_{N^h} \mathbf{1}_{N^h}^T \otimes \mathbf{I}_{d^h} \right) (\mathbf{V}^h \otimes \mathbf{I}_{d^h}) \\
 & = \left(\frac{1}{N^h} \mathbf{1}_{N^h} \mathbf{1}_{N^h}^T \mathbf{V}^h \right) \otimes (\mathbf{I}_{d^h} \mathbf{I}_{d^h}) \\
 & = \left(\frac{1}{N^h} \mathbf{1}_{N^h} \mathbf{1}_{N^h}^T \right) \otimes \mathbf{I}_{d^h} = \mathbf{R}^h,
 \end{aligned}$$

wobei die vorletzte Gleichung aufgrund der Annahme 5.4.1 gilt. Unter Verwendung dieses Resultats ergibt sich $\mathbf{R} \mathbf{V} = \text{diag}((\mathbf{R}^h \mathbf{V}^h)_{h \in \mathcal{H}}) = \mathbf{R}$. Weiterhin gelten unter Verwendung der Lipschitz Stetigkeit der lokalen

Gradienten aus Annahme 5.4.3 die Ungleichungen

$$\begin{aligned}
\|\mathbf{G}(k+1) - \mathbf{G}(k)\|^2 &= \sum_{h=1}^H \|\mathbf{G}^h(k+1) - \mathbf{G}^h(k)\|^2 \\
&= \sum_{h=1}^H \sum_{i=1}^{N^h} \|\nabla_h f_i^h(x_i(k+1)) - \nabla_h f_i^h(x_i(k))\|^2 \\
&\leq L_1^2 \sum_{h=1}^H \sum_{i=1}^{N^h} \|x_i(k+1) - x_i(k)\|^2 = L_1^2 \|\mathbf{x}(k+1) - \mathbf{x}(k)\|^2
\end{aligned}$$

und

$$\begin{aligned}
\|\mathbf{x}(k+1) - \mathbf{x}(k)\| &= \|\text{proj}_{\mathcal{X}} [\hat{\mathbf{x}}(k) - \alpha \mathbf{Q}^T \mathbf{y}(k)] - \mathbf{x}(k)\| \\
&= \|\text{proj}_{\mathcal{X}} [\hat{\mathbf{x}}(k) - \alpha \mathbf{Q}^T \mathbf{y}(k)] - \text{proj}_{\mathcal{X}} [\mathbf{x}^* - \alpha \mathbf{Q}^T \mathbf{M}(\mathbf{x}^*)]\| \\
&\quad + \|\mathbf{x}(k) - \mathbf{x}^*\| \\
&\leq \|\hat{\mathbf{x}}(k) - \alpha \mathbf{Q}^T \mathbf{y}(k) - \mathbf{x}^* + \alpha \mathbf{Q}^T \mathbf{M}(\mathbf{x}^*)\| + \|\mathbf{x}(k) - \mathbf{x}^*\| \\
&\leq \|\hat{\mathbf{x}}(k) - \alpha \mathbf{Q}^T \mathbf{M}(\mathbf{x}(k)) - \mathbf{x}^* + \alpha \mathbf{Q}^T \mathbf{M}(\mathbf{x}^*)\| + \\
&\quad \alpha \|\mathbf{Q}^T \mathbf{M}(\mathbf{x}(k)) - \mathbf{Q}^T \mathbf{y}(k)\| + \|\mathbf{x}(k) - \mathbf{x}^*\| \\
&\leq (\sqrt{\rho(A(\alpha))} + 1) \|\mathbf{x}(k) - \mathbf{x}^*\| + \alpha \|\mathbf{y}(k) - \bar{\mathbf{y}}(k)\|,
\end{aligned}$$

wobei die Resultate des Lemmas 5.4.2 für die letzte Ungleichung verwendet werden. Durch die Kombination der obigen Ergebnisse und unter Verwendung der Ungleichung (5.19), ergibt sich die folgende Schranke für die zweite Norm

$$\begin{aligned}
\|\mathbf{y}(k+1) - \bar{\mathbf{y}}(k+1)\| &\leq (\sigma_V + \alpha L_1 \|\mathbf{I} - \mathbf{R}\|) \|\mathbf{y}(k) - \bar{\mathbf{y}}(k)\| \\
&\quad + L_1 (\sqrt{\rho(A(\alpha))} + 1) \|\mathbf{I} - \mathbf{R}\| \|\mathbf{x}(k) - \mathbf{x}^*\|.
\end{aligned}$$

Der erste Teil des Beweises schließt sich durch die Sammlung der Elemente der obigen Ungleichungen in der Matrix

$$A_\tau(\alpha) = \begin{bmatrix} \sqrt{\rho(A(\alpha))} & \alpha \\ a_{21}(\alpha) & \sigma_V + \alpha a_{22} \end{bmatrix}, \quad (\text{B.10})$$

mit $\sigma_V \in (0, 1)$ wie in (5.19), $a_{21}(\alpha) = L_1 (\sqrt{\rho(A(\alpha))} + 1) \|\mathbf{I} - \mathbf{R}\|$ und $a_{22} = L_1 \|\mathbf{I} - \mathbf{R}\| > 0$.

Part 2 - Spektraler Radius: Nach Korollar 8.1.29 von [140] gilt für den Spektralen Radius von $A_T(\alpha)$, dass $\rho(A_T(\alpha)) \leq (1 - \epsilon) < 1$ mit einem kleinen $\epsilon > 0$, wenn

$$\begin{pmatrix} \sqrt{\rho(A(\alpha))} & \alpha \\ a_{21}(\alpha) & \sigma_V + \alpha a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq (1 - \epsilon) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad (\text{B.11})$$

für positive Vektoren $x_1, x_2 > 0$.

Als nächstes müssen Schranken für α gefunden werden, welche die obige Bedingung erfüllt. Von Lemma 5.4.2 ist bekannt, dass $0 < \sqrt{\rho(A(\alpha))} < 1$, wenn $0 < \alpha < \alpha_A$, wobei α_A eine von Parametern des Problems abhängige Konstante ist. Zusätzlich sei die Schranke $0 < \alpha < (1 - \sigma_V)/a_{22} = \alpha_\sigma > 0$ gegeben. Gilt diese Bedingung, dann kann der Term $\sigma_V + \alpha a_{22}$ durch $\sigma_{v,\alpha} \in (0, 1)$ ersetzt werden.

Aus dem Ungleichungssystem (B.11) können die Ungleichungen

$$\alpha \leq \frac{x_1}{x_2} \left(1 - \epsilon - \sqrt{\rho(A(\alpha))}\right), \quad \frac{x_1}{x_2} \leq \frac{(1 - \epsilon - \sigma_{v,\alpha})}{a_{21}(\alpha)},$$

gewonnen werden. Die Kombination dieser Ungleichungen resultiert wiederum in

$$\alpha \leq \frac{(1 - \epsilon - \sqrt{\rho(A(\alpha))})(1 - \epsilon - \sigma_{v,\alpha})}{a_{21}(\alpha)} = \alpha_\epsilon.$$

Es sei angemerkt, dass $a_{21}(\alpha) > 0$. Mit

$$\alpha^*(\alpha) = \lim_{\epsilon \rightarrow 0} \alpha_\epsilon = \frac{(1 - \sqrt{\rho(A(\alpha))})(1 - \sigma_{v,\alpha})}{a_{21}(\alpha)}$$

folgt unter der Wahl $\alpha \in (0, \min\{\alpha^*(\alpha), \alpha_A, \alpha_\sigma\})$, dass $\rho(A_T(\alpha)) < 1$.

Als nächstes wird demonstriert, wie α gewählt werden kann, damit $0 < \alpha < \min\{\alpha^*(\alpha), \alpha_A, \alpha_\sigma\}$ erfüllt ist.

Für jedes $\alpha > 0$, sodass $\alpha < \min\{\alpha_A/2, \alpha_\sigma\} = \alpha_{A,\sigma}$, ergibt sich $\alpha^*(\alpha) > \frac{\delta_1(\alpha)\delta_2(\alpha)}{2L_1\|I-\mathbf{R}\|}$ mit $\delta_1(\alpha) = 1 - \sqrt{\rho(A(\alpha))} > 0$ and $\delta_2(\alpha) = 1 - \sigma_{v,\alpha} > 0$. Unter Verwendung von Korollar 8.1.19 aus [140] und durch Untersuchen der Matrix $A(\alpha)$ aus Lemma 5.4.2, ergibt sich $\sqrt{\rho(A(\alpha))} \leq \sqrt{\rho(A(\alpha_{A,\sigma}))}$, wodurch $\delta_1(\alpha) \geq \delta_1(\alpha_{A,\sigma})$ und $\delta_2(\alpha) > \delta_2(\alpha_{A,\sigma})$ gilt. Wenn also $\alpha < \alpha_{A,\sigma}$ gilt, folgt $\alpha^*(\alpha) > \alpha^*(\alpha_{A,\sigma}) > 0$. Daher stellt die Wahl $0 < \alpha < \bar{\alpha} = \min\{\alpha^*(\alpha_{A,\sigma}), \alpha_{A,\sigma}\}$ mit $\alpha_{A,\sigma} = \min\{\alpha_A, \alpha_\sigma\}$ sicher, dass $\rho(A_T(\alpha)) < 1$. \square

B.4 Parameterkonfiguration für Simulationen des Einsatzplanproblems zwischen Microgrids

In den Tabellen B.1, ?? und B.2 ist die Konfiguration und Parametrisierung des Multiclusterspiels zwischen Microgrids bezüglich der Simulationen aus Abschnitt 5.5 aufgeführt.

Tabelle B.1: Parametrierung der Kostenfunktionen sowie der Kapazität \bar{e} der Leitungen zwischen den Bussen der Microgrids und der Leitung zum Hauptnetzwerk (HN). Alle drei 5-Busmicrogrids besitzen die selbe Konfiguration der Leitungen. Alle Werte sind auf die Einheit MW bezogen.

	$a_{L,i}$	\bar{e}_i
1 - HN	0	800
1 - 2	0.01	400
1 - 3	0.01	100
1 - 5	0.01	100
2 - 3	0.01	100
2 - 4	0.01	100
4 - 5	0.01	240

Tabelle B.2: Übersicht über die Parametrierung der Simulation des Einsatzplanproblems zwischen Microgrids. Alle Werte bezogen auf die Einheit MW. Zuordnung der Speicher und Generatoren aus Tabellen (b) und (c) zu den Bussen der Microgrids in Tabelle (a).

(a) Bus Konfiguration für Microgrids (MG) unter Angabe der angeschlossenen Generatoren (G_i) und Speicher (S_i).

	MG 1	MG 2	MG 3
Bus 1	G_1^1, G_2^1		S_1^3
Bus 2	-	S_1^2	S_2^3
Bus 3	G_3^1	G_1^2, S_2^2	S_3^3
Bus 4	G_4^1	G_2^2	S_4^3
Bus 5	G_5^1	G_3^2	G_5^3

(b) Parametrierung der Kostenfunktion, Startwerte und Ladungskapazitäten ($q(0), \bar{q}$), Leakagerate (γ), maximale Abweichung der Ladung zwischen Startwert und Endwert (ϵ) sowie untere und obere Schranke der Strategie (\underline{s}, \bar{s}) aller Speicher der Microgrids.

	$a_{S,i}^h$	$q_i^h(0)$	\bar{q}_i^h	γ_i^h	ϵ_i^h	\underline{s}_i^h	\bar{s}_i^h
S_1^2	0,05	50	600	0,99	0,01	-300	300
S_2^2	0,09	100	400	0,99	0,01	-200	200
S_1^3	0,05	20	600	0,99	0,01	-600	600
S_2^3	0,09	20	600	0,99	0,01	-300	300
S_3^3	0,05	100	400	0,99	0,01	-200	200
S_4^3	0,09	100	600	0,99	0,01	-300	300

(c) Parametrierung der Kostenfunktionen sowie untere und obere Schranke (\underline{g}, \bar{g}) aller Generatoren der Microgrids.

	$a_{G,i}^h$	$b_{G,i}^h$	$c_{G,i}^h$	\underline{g}_i^h	\bar{g}_i^h
G_1^1	0,2	6	0	0	40
G_2^1	0,047	7	0	0	170
G_3^1	0,038	10	0	0	520
G_4^1	0,145	11	0	0	200
G_5^1	0,008	5	0	0	600
G_1^2	0,038	10	0	0	520
G_2^2	0,145	11	0	0	200
G_3^3	0,008	5	0	0	600
G_1^3	0,008	5	0	0	600

Tabelle B.2: Verbrauch innerhalb der Microgrids über die Planungsperiode $T = 24$. Alle drei 5-Busmicrogrids besitzen den selben Verbrauchsverlauf. Alle Angaben in MW.

	Bus 2	Bus 3	Bus 4				
t = 1	30	30	30	t = 13	120	100	120
t = 2	30	30	30	t = 14	70	80	70
t = 3	30	30	30	t = 15	70	100	70
t = 4	30	30	30	t = 16	120	100	120
t = 5	70	40	70	t = 17	170	110	170
t = 6	80	50	80	t = 18	220	170	220
t = 7	120	50	120	t = 19	320	220	320
t = 8	120	50	120	t = 20	270	170	270
t = 9	170	70	170	t = 21	220	120	220
t = 10	120	50	120	t = 22	220	120	220
t = 11	100	70	100	t = 23	120	70	120
t = 12	140	100	140	t = 24	30	40	30

B.5 Kommunikationsgraphen für Simulationen aus Abschnitt 5.5

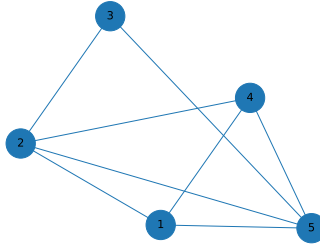


Abbildung B.1: Lokaler Kommunikationsgraph des 5-Bussystems. In Simulationen für alle 5-Bussysteme gleich.

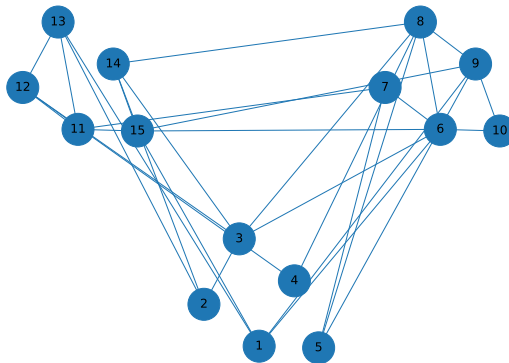


Abbildung B.2: Globaler Kommunikationsgraph, der alle Agenten unabhängig der Clusterzugehörigkeit miteinander verbindet.

Veröffentlichungen

- [32] T. Tatarenko, J. Zimmermann, V. Willert und J. Adamy, Penalized Push-Sum Algorithm for Constrained Distributed Optimization with Application to Energy Management in Smart Grid, in *2019 IEEE 58th Conference on Decision and Control (CDC)*, ISSN: 2576-2370, 2019, S. 6234–6241.
- [33] J. Zimmermann, T. Tatarenko, V. Willert und J. Adamy, Projected Push-Sum Gradient Descent-Ascent for Convex Optimization with Application to Economic Dispatch Problems, in *2020 59th IEEE Conference on Decision and Control (CDC)*, ISSN: 2576-2370, 2020, S. 542–547.
- [34] J. Zimmermann, T. Tatarenko, V. Willert und J. Adamy, Projected gradient-tracking in multi-cluster games and its application to power management, *arXiv:2202.12124 [cs, eess]*, 2022.
- [35] J. Zimmermann, T. Tatarenko, V. Willert und J. Adamy, Optimales Energie-Management über verteilte, beschränkte Gradientenverfahren, *at - Automatisierungstechnik*, Jg. 67, Nr. 11, S. 922–935, 2019.
- [36] T. Tatarenko, J. Zimmermann und J. Adamy, Gradient Play in n-Cluster Games with Zero-Order Information, in *2021 60th IEEE Conference on Decision and Control (CDC)*, ISSN: 2576-2370, 2021, S. 3104–3109.
- [37] T. Tatarenko und J. Zimmermann, Distributed optimization methods for N -cluster games, *at - Automatisierungstechnik*, Jg. 70, Nr. 3, S. 237–247, 2022.
- [38] J. Zimmermann, T. Tatarenko, V. Willert und J. Adamy, Solving leaderless multi-cluster games over directed graphs, *European Journal of Control*, 2021 European Control Conference Special Issue, Jg. 62, S. 14–21, 2021.
- [39] J. Zimmermann, T. Tatarenko, V. Willert und J. Adamy, Gradient-Tracking-basierte Lösung von Multi-Cluster-Spielen, *at - Automatisierungstechnik*, Jg. 70, Nr. 4, S. 355–366, 2022.

Wissenschaftliche Vorträge

- [40] J. Zimmermann, Penalized Push-Sum Algorithm for Constrained Distributed Optimization with Application to Energy Management in Smart Grid, 58th Conference on Decision and Control, Nizza, Frankreich, 2019.
- [41] J. Zimmermann, Projected Push-Sum Gradient Descent-Ascent for Convex Optimization with Application to Economic Dispatch Problems, 59th Conference on Decision and Control, online, 2020.
- [42] J. Zimmermann, Gradient-Play in n -Cluster Games with Zero-Order Information, 60th Conference on Decision and Control, online, 2021.
- [43] J. Zimmermann, Solving Leaderless Multi-Cluster Games over Directed Graphs, 2021 European Control Conference, online, 2021.
- [44] J. Zimmermann, Verteilte Clusteranalyse von vernetzten Datensätzen, GMA 1.50 Grundlagen vernetzter Systeme, Günzburg, Deutschland, 2019.
- [45] J. Zimmermann, Gradient-Tracking Algorithmus zur Lösung von Multi-Cluster-Spielen, GMA 1.50 Grundlagen vernetzter Systeme, online, 2021.

Literatur

- [1] I. Llatser, T. Michalke, M. Dolgov, F. Wildschütte und H. Fuchs, Cooperative Automated Driving Use Cases for 5G V2X Communication, in *2019 IEEE 2nd 5G World Forum (5GWF)*, 2019, S. 120–125.
- [2] Z. Lin und H. H. Liu, Topology-Based Distributed Optimization for Multi-UAV Cooperative Wildfire Monitoring, *Optimal Control Applications and Methods*, Jg. 39, Nr. 4, S. 1530–1548, 2018.
- [3] M. Cesana, I. Malanchini und A. Capone, Modelling Network Selection and Resource Allocation in Wireless Access Networks with Non-Cooperative Games, in *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, 2008, S. 404–409.
- [4] M. Bianchi und S. Grammatico, Fully Distributed Nash Equilibrium Seeking Over Time-Varying Communication Networks with Linear Convergence Rate, *IEEE Control Systems Letters*, Jg. 5, Nr. 2, S. 499–504, 2021.
- [5] M. Bianchi, G. Belgioioso und S. Grammatico, Fast Generalized Nash Equilibrium Seeking under Partial-Decision Information, *Automatica*, Jg. 136, 2022.
- [6] J. Gochermann, Expedition Energiewende. Spektrum Akademischer Verlag GmbH, 2016.
- [7] A. F. Zobaa, S. H. E. A. Aleem und A. Y. Abdelaziz, Classical and Recent Aspects of Power System Optimization. Academic Press, 2018.
- [8] D. K. Molzahn, F. Dörfler, H. Sandberg u. a., A Survey of Distributed Optimization and Control Algorithms for Electric Power Systems, *IEEE Transactions on Smart Grid*, Jg. 8, Nr. 6, S. 2941–2962, 2017.
- [9] H. Rahimi-Eichi, U. Ojha, F. Baronti und M.-Y. Chow, Battery Management System: An Overview of Its Application in the Smart Grid and Electric Vehicles, *IEEE Industrial Electronics Magazine*, Jg. 7, Nr. 2, S. 4–16, 2013.

- [10] Fraunhofer ISE. Quartierspeicher. Letzter Zugriff am 22.12.2022. (2022), Adresse: <https://www.ise.fraunhofer.de/de/leitthemen/stationaere-batteriespeicher/quartierspeicher.html>.
- [11] D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi u. a., Trends in Microgrid Control, *IEEE Transactions on Smart Grid*, Jg. 5, Nr. 4, S. 1905–1919, 2014.
- [12] M. F. Zia, M. Benbouzid, E. Elbouchikhi, S. M. Muyeen, K. Techato und J. M. Guerrero, Microgrid Transactive Energy: Review, Architectures, Distributed Ledger Technologies, and Market Analysis, *IEEE Access*, Jg. 8, S. 19410–19432, 2020.
- [13] A. Nedić und J. Liu, Distributed Optimization for Control, *Annual Review of Control, Robotics, and Autonomous Systems*, Jg. 1, Nr. 1, S. 77–103, 2018.
- [14] T. Yang, X. Yi, J. Wu u. a., A Survey of Distributed Optimization, *Annual Reviews in Control*, Jg. 47, S. 278–305, 2019.
- [15] A. C. Tellidou und A. G. Bakirtzis, Agent-Based Analysis of Capacity Withholding and Tacit Collusion in Electricity Markets, *IEEE Transactions on Power Systems*, Jg. 22, Nr. 4, S. 1735–1742, 2007.
- [16] I. Atzeni, L. G. Ordóñez, G. Scutari, D. P. Palomar und J. R. Fonollosa, Noncooperative and Cooperative Optimization of Distributed Energy Generation and Storage in the Demand-Side of the Smart Grid, *IEEE Transactions on Signal Processing*, Jg. 61, Nr. 10, S. 2454–2472, 2013.
- [17] G. Belgioioso, W. Ananduta, S. Grammatico und C. Ocampo-Martinez, Energy Management and Peer-To-Peer Trading in Future Smart Grids: A Distributed Game-Theoretic Approach, in *2020 European Control Conference (ECC)*, 2020, S. 1324–1329.
- [18] A. Nedić und A. Olshevsky, Distributed Optimization Over Time-Varying Directed Graphs, *IEEE Transactions on Automatic Control*, Jg. 60, Nr. 3, S. 601–615, 2015.
- [19] W. Shi, Q. Ling, G. Wu und W. Yin, Extra: An Exact First-Order Algorithm for Decentralized Consensus Optimization, *SIAM Journal on Optimization*, Jg. 25, Nr. 2, S. 944–966, 2015.
- [20] D. Jakovetic, J. Xavier und J. M. Moura, Fast Distributed Gradient Methods, *IEEE Transactions on Automatic Control*, Jg. 59, Nr. 5, S. 1131–1146, 2014.

- [21] G. Qu und N. Li, Accelerated Distributed Nesterov Gradient Descent, in *IEEE Transactions on Automatic Control*, Bd. 65, Institute of Electrical und Electronics Engineers Inc., 2020, S. 2566–2581.
- [22] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee und L. Massoulié, Optimal Algorithms for Smooth and Strongly Convex Distributed Optimization in Networks, in *International Conference on Machine Learning*, PMLR, 2017, S. 3027–3036.
- [23] S. A. Gabriel, S. A. Siddiqui, A. J. Conejo und C. Ruiz, Solving Discretely-Constrained Nash–Cournot Games with an Application to Power Markets, *Networks and Spatial Economics*, Jg. 13, Nr. 3, S. 307–326, 2013.
- [24] M. Meng und X. Li, On the Linear Convergence of Distributed Nash Equilibrium Seeking for Multi-Cluster Games under Partial-Decision Information, Nr. arXiv:2005.06923, 2022.
- [25] J. Zhou, Y. Lv, G. Wen, J. Lü und D. Zheng, Distributed Nash Equilibrium Seeking in Consistency-Constrained Multicoalition Games, *IEEE Transactions on Cybernetics*, S. 1–13, 2022.
- [26] Y. Pang und G. Hu, Fully Distributed Nash Equilibrium Seeking in N-Cluster Games, arXiv, arXiv:2012.11347, 2021.
- [27] Y. Pang und G. Hu, Nash Equilibrium Seeking in N-Coalition Games Via a Gradient-Free Method, *Automatica*, Jg. 136, 2022.
- [28] S. Pu, W. Shi, J. Xu und A. Nedić, Push-Pull Gradient Methods for Distributed Optimization in Networks, *IEEE Transactions on Automatic Control*, S. 1–1, 2020.
- [29] Z. Dong, S. Mao, W. Du und Y. Tang, Distributed Constrained Optimization with Linear Convergence Rate, in *IEEE International Conference on Control and Automation, ICCA*, Bd. 2020-Octob, IEEE Computer Society, 2020, S. 937–942.
- [30] K. I. Tsianos, S. Lawlor und M. G. Rabbat, Consensus-Based Distributed Optimization: Practical Issues and Applications in Large-Scale Machine Learning, in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, 2012, S. 1543–1550.
- [31] A. Nedić, A. Olshevsky und W. Shi, Achieving Geometric Convergence for Distributed Optimization Over Time-Varying Graphs, *SIAM J. Control and Optimization*, Jg. 18, Nr. 3, S. 1046–1060, 2017.

- [46] P. Baran, On Distributed Communications Networks, *IEEE Transactions on Communications Systems*, Jg. 12, Nr. 1, S. 1–9, 1964.
- [47] C. Kanzow und A. Schwartz, Spieltheorie. Springer International Publishing, 2018.
- [48] R. A. Polyak, Introduction to Continuous Optimization. Springer Nature, 2021, 552 S.
- [49] R. G. Bartle und D. R. Sherbert, Introduction to Real Analysis, 4. Aufl. Wiley, 2011, 416 S.
- [50] D. P. Bertsekas, Nonlinear Programming, *Journal of the Operational Research Society*, Jg. 48, Nr. 3, S. 334–334, 1997.
- [51] S. Boyd, L. Vandenberghe und C. Optimization, Convex Optimization, *Communication Networkingo*, S. 803–818, 2004.
- [52] A. Beck, First-Order Methods in Optimization. Society for Industrial und Applied Mathematics, 2017.
- [53] A. Nedić und A. Ozdaglar, Subgradient methods for saddle-point problems, *Journal of Optimization Theory and Applications*, Jg. 142, Nr. 1, S. 205–228, 2009.
- [54] Y. Nesterov, Introductory Lectures on Convex Programming Volume I: Basic course, *Lecture Notes*, Jg. I, S. 1–211, 1998.
- [55] T. Tatarenko und M. Kamgarpour, Learning Generalized Nash Equilibria in a Class of Convex Games, *IEEE Transactions on Automatic Control*, Jg. 64, Nr. 4, S. 1426–1439, 2019.
- [56] G. Scutari, D. Palomar, F. Facchinei und J.-S. Pang, Convex Optimization, Game Theory, and Variational Inequality Theory, *IEEE Signal Processing Magazine*, Jg. 27, Nr. 3, S. 35–49, 2010.
- [57] F. Facchinei und J.-S. Pang, Finite-Dimensional Variational Inequalities and Complementarity Problems: Volume I. Springer Science & Business Media, 2007.
- [58] M. Mesbahi und M. Egerstedt, Graph Theoretic Methods in Multiagent Networks. Princeton University Press, 2010.
- [59] T. C. Aysal, M. E. Yildiz, A. D. Sarwate und A. Scaglione, Broadcast Gossip Algorithms for Consensus, *IEEE Transactions on Signal Processing*, Jg. 57, Nr. 7, S. 2748–2761, 2009.

- [60] T. Charalambous, M. G. Rabbat, M. Johansson und C. N. Hadjicostis, Distributed Finite-Time Computation of Digraph Parameters: Left-Eigenvector, Out-Degree and Spectrum, *IEEE Transactions on Control of Network Systems*, Jg. 3, Nr. 2, S. 137–148, 2016.
- [61] B. Gharesifard und J. Cortés, Distributed strategies for generating weight-balanced and doubly stochastic digraphs, *European Journal of Control*, Jg. 18, Nr. 6, S. 539–557, 2012.
- [62] R. Olfati-Saber, J. A. Fax und R. M. Murray, Consensus and Cooperation in Networked Multi-Agent Systems, *Proceedings of the IEEE*, Jg. 95, Nr. 1, S. 215–233, 2007.
- [63] E. Seneta, *Non-Negative Matrices and Markov Chains*. Springer Science & Business Media, 2006.
- [64] A. D. Dominguez-Garcia und C. N. Hadjicostis, Coordination and Control of Distributed Energy Resources for Provision of Ancillary Services, in *2010 First IEEE International Conference on Smart Grid Communications*, 2010, S. 537–542.
- [65] T. Charalambous und C. N. Hadjicostis, Average Consensus in the Presence of Dynamically Changing Directed Topologies and Time Delays, in *53rd IEEE Conference on Decision and Control*, 2014, S. 709–714.
- [66] L. Ageeva, M. Majidi und D. Pozo, *Analysis of Feasibility Region of Active Distribution Networks*. 2019.
- [67] Y. Wang, L. Wu und S. Wang, A Fully-Decentralized Consensus-Based ADMM Approach for DC-OPF with Demand Response, *IEEE Transactions on Smart Grid*, Jg. 8, Nr. 6, S. 2637–2647, 2017.
- [68] C. Zhao, J. He, P. Cheng und J. Chen, Consensus-Based Energy Management in Smart Grid with Transmission Losses and Directed Communication, *IEEE Transactions on Smart Grid*, Jg. 8, Nr. 5, S. 2049–2061, 2017.
- [69] Z.-L. Gaing, Particle Swarm Optimization to Solving the Economic Dispatch Considering the Generator Constraints, *IEEE Transactions on Power Systems*, Jg. 18, Nr. 3, S. 1187–1195, 2003.
- [70] J. Marden, S. Ruben und L. Pao, Surveying Game Theoretic Approaches for Wind Farm Optimization, in *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics und Astronautics, 2012.

- [71] A. Nemirovski und A. Shapiro, Convex Approximations of Chance Constrained Programs, *SIAM Journal on Optimization*, Jg. 17, Nr. 4, S. 969–996, 2006.
- [72] A. Kargarian, J. Mohammadi, J. Guo u. a., Toward Distributed/decentralized DC Optimal Power Flow Implementation in Future Electric Power Systems, *IEEE Transactions on Smart Grid*, Jg. 9, Nr. 4, S. 2574–2594, 2018.
- [73] A. Wood und B. Wollenberg, Power Generation, Operation, and Control. John Wiley & Sons Inc, 2012, S. 660.
- [74] Z. Zhang und M.-Y. Chow, Convergence Analysis of the Incremental Cost Consensus Algorithm under Different Communication Network Topologies in a Smart Grid, *IEEE Transactions on Power Systems*, Jg. 27, Nr. 4, S. 1761–1768, 2012.
- [75] P.-H. Chen und H.-C. Chang, Large-Scale Economic Dispatch by Genetic Algorithm, *IEEE Transactions on Power Systems*, Jg. 10, Nr. 4, S. 1919–1926, 1995.
- [76] S. Frank und S. Rebennack, An Introduction to Optimal Power Flow: Theory, Formulation, and Examples, *IIE Transactions*, Jg. 48, Nr. 12, S. 1172–1197, 2016.
- [77] A. Conejo und L. Baringo, Power System Operations. Springer International Publishing AG, 2018.
- [78] F. Guo, C. Wen, J. Mao und Y.-D. Song, Distributed Economic Dispatch for Smart Grids with Random Wind Power, *IEEE Transactions on Smart Grid*, Jg. 7, Nr. 3, S. 1572–1583, 2016.
- [79] W. Shi, X. Xie, C.-C. Chu und R. Gadh, Distributed Optimal Energy Management in Microgrids, *IEEE Transactions on Smart Grid*, Jg. 6, Nr. 3, S. 1137–1146, 2015.
- [80] S. Mhanna, G. Verbič und A. C. Chapman, Adaptive ADMM for Distributed AC Optimal Power Flow, *IEEE Transactions on Power Systems*, Jg. 34, Nr. 3, S. 2025–2035, 2019.
- [81] A. Falsone, K. Margellos, S. Garatti und M. Prandini, Dual Decomposition for Multi-Agent Distributed Optimization with Coupling Constraints, *Automatica*, Jg. 84, S. 149–158, 2017.
- [82] S. Van Mai und E. H. Abed, Distributed Optimization Over Weighted Directed Graphs Using Row Stochastic Matrix, *Proceedings of the American Control Conference*, Jg. 2016-July, S. 7165–7170, 2016.

- [83] H. Li, Q. Lu und T. Huang, Distributed Projection Subgradient Algorithm over Time-Varying General Unbalanced Directed Graphs, *IEEE Transactions on Automatic Control*, Jg. 64, Nr. 3, S. 1309–1316, 2019.
- [84] K. Tsianos, S. Lawlor und M. Rabbat, Push-Sum Distributed Dual Averaging for Convex Optimization, in *Proceedings of the 51st IEEE Annual Conference on Decision and Control*, 2012, S. 5453–5458.
- [85] Y. Zhang und M. M. Zavlanos, A Consensus-Based Distributed Augmented Lagrangian Method, in *Proceedings of the IEEE Conference on Decision and Control*, Institute of Electrical und Electronics Engineers Inc., 2019, S. 1763–1768.
- [86] J. Giesen und S. Laue, Distributed Convex Optimization with Many Convex Constraints, *arXiv:1610.02967*, S. 1–18, 2016.
- [87] S. Boyd, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, *Foundations and Trends in Machine Learning*, Jg. 3, Nr. 1, S. 1–122, 2010.
- [88] D. Jakovetic, J. Xavier und J. M. Moura, Cooperative Convex Optimization in Networked Systems: Augmented Lagrangian Algorithms with Directed Gossip Communication, *IEEE Transactions on Signal Processing*, Jg. 59, Nr. 8, S. 3889–3902, 2011.
- [89] A. Falsone und M. Prandini, Distributed Decision-Coupled Constrained Optimization Via Proximal-Tracking, *Automatica*, Jg. 135, 2022.
- [90] Z. Li, W. Shi und M. Yan, A Decentralized Proximal-Gradient Method With Network Independent Step-Sizes and Separated Convergence Rates, *IEEE Transactions on Signal Processing*, Jg. 67, Nr. 17, S. 4494–4506, 2019.
- [91] X. Li, G. Feng und L. Xie, Distributed Proximal Algorithms for Multiagent Optimization with Coupled Inequality Constraints, *IEEE Transactions on Automatic Control*, Jg. 66, Nr. 3, S. 1223–1230, 2021.
- [92] K. Margellos, A. Falsone, S. Garatti und M. Prandini, Distributed Constrained Optimization and Consensus in Uncertain Networks via Proximal Minimization, *IEEE Transactions on Automatic Control*, Jg. 63, Nr. 5, S. 1372–1387, 2018.

- [93] A. Nedić und A. Ozdaglar, Distributed Subgradient Methods for Multi-Agent Optimization, *IEEE Transactions on Automatic Control*, Jg. 54, Nr. 1, S. 48–61, 2009.
- [94] S. A. Alghunaim und A. H. Sayed, Linear Convergence of Primal–Dual Gradient Methods and Their Performance in Distributed Optimization, *Automatica*, Jg. 117, S. 109 003, 2020.
- [95] G. Qu und N. Li, Harnessing Smoothness to Accelerate Distributed Optimization, *IEEE Transactions on Control of Network Systems*, Jg. 5, Nr. 3, S. 1245–1260, 2018.
- [96] R. Xin, C. Xi und U. A. Khan, Frost—Fast Row-Stochastic Optimization with Uncoordinated Step-Sizes, *Eurasip Journal on Advances in Signal Processing*, Jg. 2019, Nr. 1, S. 1, 2019.
- [97] R. Xin, A. K. Sahu, U. A. Khan und S. Kar, Distributed Stochastic Optimization with Gradient Tracking Over Strongly-Connected Networks, in *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, 2019, S. 8353–8358.
- [98] R. Xin und U. A. Khan, A Linear Algorithm for Optimization over Directed Graphs with Geometric Convergence, *IEEE Control Systems Letters*, Jg. 2, Nr. 3, S. 313–318, 2018.
- [99] F. Saadatniaki, R. Xin und U. A. Khan, Decentralized Optimization over Time-Varying Directed Graphs with Row and Column-Stochastic Matrices, *IEEE Transactions on Automatic Control*, Jg. 65, Nr. 11, S. 4769–4780, 2020.
- [100] A. Nedić, A. Ozdaglar und P. A. Parrilo, Constrained Consensus and Optimization in Multi-Agent Networks, *IEEE Transactions on Automatic Control*, Jg. 55, Nr. 4, S. 922–938, 2010.
- [101] P. Lin, W. Ren und Y. Song, Distributed Multi-Agent Optimization Subject to Nonidentical Constraints and Communication Delays, *Automatica*, Jg. 65, S. 120–131, 2016.
- [102] D. Jakovetić, D. Bajović, N. Krejić und N. Krklec Jerinkić, Distributed Gradient Methods with Variable Number of Working Nodes, *IEEE Transactions on Signal Processing*, Jg. 64, Nr. 15, S. 4080–4095, 2016.
- [103] S. Lee und A. Nedić, Distributed Random Projection Algorithm for Convex Optimization, *IEEE Journal on Selected Topics in Signal Processing*, Jg. 7, Nr. 2, S. 221–229, 2013.

- [104] C. Xi und U. A. Khan, Distributed Subgradient Projection Algorithm over Directed Graphs, *IEEE Transactions on Automatic Control*, Jg. 62, Nr. 8, S. 3986–3992, 2017.
- [105] T. H. Chang, A. Nedić und A. Scaglione, Distributed Constrained Optimization by Consensus-Based Primal-Dual Perturbation Method, *IEEE Transactions on Automatic Control*, Jg. 59, Nr. 6, S. 1524–1538, 2014.
- [106] M. Zhu und S. Martinez, On Distributed Convex Optimization under Inequality and Equality Constraints, *IEEE Transactions on Automatic Control*, Jg. 57, Nr. 1, S. 151–164, 2012.
- [107] J. Lei, H. F. Chen und H. T. Fang, Primal–Dual Algorithm for Distributed Constrained Optimization, *Systems and Control Letters*, Jg. 96, S. 110–117, 2016.
- [108] J. Zhang, K. You und K. Cai, Distributed Dual Gradient Tracking for Resource Allocation in Unbalanced Networks, *IEEE Transactions on Signal Processing*, Jg. 68, S. 2186–2198, 2020.
- [109] A. Nedić, A. Olshevsky und M. G. Rabbat, Network Topology and Communication-Computation Tradeoffs in Decentralized Optimization, *Proceedings of the IEEE*, Jg. 106, Nr. 5, S. 953–976, 2018.
- [110] A. Nedić, Distributed Gradient Methods for Convex Machine Learning Problems in Networks: Distributed Optimization, *IEEE Signal Processing Magazine*, Jg. 37, Nr. 3, S. 92–101, 2020.
- [111] K. Yuan, Q. Ling und W. Yin, On the Convergence of Decentralized Gradient Descent, *SIAM Journal on Optimization*, Jg. 26, Nr. 3, S. 1835–1854, 2016.
- [112] M. Zhu und S. Martínez, Discrete-Time Dynamic Average Consensus, *Automatica*, Jg. 46, Nr. 2, S. 322–329, 2010.
- [113] C. Xi, V. S. Mai, R. Xin, E. H. Abed und U. A. Khan, Linear Convergence in Optimization Over Directed Graphs With Row-Stochastic Matrices, *IEEE Transactions on Automatic Control*, Jg. 63, Nr. 10, S. 3558–3565, 2018.
- [114] K. Cai und H. Ishii, Average Consensus on Arbitrary Strongly Connected Digraphs with Time-Varying Topologies, *IEEE Transactions on Automatic Control*, Jg. 59, Nr. 4, S. 1066–1071, 2014.

- [115] J.-B. Hiriart-Urruty und C. Lemarechal, *Convex Analysis and Minimization Algorithms I: Fundamentals*. Springer Science & Business Media, 1996.
- [116] H. Robbins und D. Siegmund, A Convergence Theorem for Non Negative Almost Supermartingales and Some Applications, in *Herbert Robbins Selected Papers*, Springer, 1985, S. 111–135.
- [117] L. Thurner, A. Scheidler, F. Schäfer u. a., Pandapower—An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems, *IEEE Transactions on Power Systems*, Jg. 33, Nr. 6, S. 6510–6521, 2018.
- [118] Fraunhofer IEE und Universität Kassel. Pandapower Documentation - Power System Test Cases. Letzter Zugriff am 22.12.2022. (2022), Adresse: https://pandapower.readthedocs.io/en/v2.11.0/networks/power_system_test_cases.html.
- [119] F. Li und R. Bo, Small Test Systems for Power System Economic Studies, in *IEEE PES General Meeting*, 2010, S. 1–4.
- [120] P. M. Anderson und A. A. Fouad, *Power System Control and Stability*. Wiley, 2003.
- [121] pyOpt Developers. SLSQP - Sequential Least Squares Programming. Letzter Zugriff am 22.12.2022. (2022), Adresse: <http://www.pyopt.org/reference/optimizers.slsqp.html>.
- [122] J. Lei, U. Shanbhag und J. Chen, Distributed Computation of Nash Equilibria for Monotone Aggregative Games Via Iterative Regularization, in *2020 59th IEEE Conference on Decision and Control (CDC) Jeju Island, Republic of Korea, December 14-18, 2020*, IEEE, 2020.
- [123] M. Bianchi, G. Belgioioso und S. Grammatico, A Fully-Distributed Proximal-Point Algorithm for Nash Equilibrium Seeking with Linear Convergence Rate, in *2020 59th IEEE Conference on Decision and Control (CDC) Jeju Island, Republic of Korea, December 14-18, 2020*, IEEE, 2020.
- [124] T. Tatarenko und M. Kamgarpour, Payoff-Based Approach to Learning Nash Equilibria in Convex Games, *IFAC-PapersOnLine*, Jg. 50, Nr. 1, S. 1508–1513, 2017.

- [125] T. Tatarenko, W. Shi und A. Nedić, Geometric Convergence of Gradient Play Algorithms for Distributed Nash Equilibrium Seeking, *IEEE Transactions on Automatic Control*, Jg. 66, Nr. 11, S. 5342–5353, 2021.
- [126] L. Pavel, Distributed GNE Seeking under Partial-Decision Information Over Networks Via a Doubly-Augmented Operator Splitting Approach, *IEEE Transactions on Automatic Control*, Jg. 65, Nr. 4, S. 1584–1597, 2020.
- [127] T. Tatarenko und A. Nedić, Geometric Convergence of Distributed Gradient Play in Games with Unconstrained Action Sets, *IFAC-PapersOnLine*, 21st IFAC World Congress, Jg. 53, Nr. 2, S. 3367–3372, 2020.
- [128] M. Ye, G. Hu und F. L. Lewis, Nash equilibrium seeking for n-coalition noncooperative games, *Automatica*, Jg. 95, S. 266–272, 2018.
- [129] M. Ye, G. Hu, F. L. Lewis und L. Xie, A Unified Strategy for Solution Seeking in Graphical n-Coalition Noncooperative Games, *IEEE Transactions on Automatic Control*, Jg. 64, Nr. 11, S. 4645–4652, 2019.
- [130] M. Ye, G. Hu und S. Xu, An Extremum Seeking-Based Approach for Nash Equilibrium Seeking in N-Cluster Noncooperative Games, *Automatica*, Jg. 114, S. 108 815, 2020.
- [131] F. Liu, X. Dong, J. Yu, Y. Hua, Q. Li und Z. Ren, Distributed Nash Equilibrium Seeking of n-Coalition Noncooperative Games with Application to UAV Swarms, *IEEE Transactions on Network Science and Engineering*, Jg. 9, Nr. 4, S. 2392–2405, 2022.
- [132] D. Yue und Z. Meng, Distributed Nash Equilibrium Seeking for Constrained Games, in *2018 37th Chinese Control Conference (CCC)*, 2018, S. 9660–9665.
- [133] Y. Zou, B. Huang, Z. Meng und W. Ren, Distributed Nash Equilibrium Seeking Algorithms for Two-Layer Constrained Non-Cooperative Games, in *2019 Chinese Control Conference (CCC)*, 2019, S. 5776–5781.
- [134] X. Zeng, J. Chen, S. Liang und Y. Hong, Generalized nash equilibrium seeking strategy for distributed nonsmooth multi-cluster game, *Automatica*, Jg. 103, S. 20–26, 2019.

-
- [135] C. Sun und G. Hu, Distributed Generalized Nash Equilibrium Seeking of N-Coalition Games with Inequality Constraints, in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, S. 215–220.
- [136] G. Hu, Robust consensus tracking of a class of second-order multi-agent dynamic systems, *Systems & Control Letters*, Jg. 61, Nr. 1, S. 134–142, 2012.
- [137] A. S. Berahas, R. Bollapragada, N. S. Keskar und E. Wei, Balancing Communication and Computation in Distributed Optimization, *IEEE Transactions on Automatic Control*, Jg. 64, Nr. 8, S. 3141–3155, 2019.
- [138] V. Blondel, J. Hendrickx, A. Olshevsky und J. Tsitsiklis, Convergence in Multiagent Coordination, Consensus, and Flocking, in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, S. 2996–3000.
- [139] J. M. Anthonisse und H. Tijms, Exponential convergence of products of stochastic matrices, *Journal of Mathematical Analysis and Applications*, Jg. 59, Nr. 2, S. 360–364, 1977.
- [140] R. A. Horn und C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1990.