

Universität Stuttgart
Institut für Energiespeicherung
Professor Dr. André Thess



**Deutsches Zentrum
für Luft- und Raumfahrt**
Institut für Technische Thermodynamik

Forschungsarbeit

Evaluierung konzeptioneller Beschleunigungsstrategien für optimierende Energiesystemmodelle

eingereicht von Simon Brodbeck

Matr.-Nr.: 2712804

Beginn: 01.12.2016

Abgabe: 31.03.2017

Betreuer:

Professor Dr. André Thess

Karl-Kiên Cao, M. Sc.

Dipl.-Phys. Kai Von Krbek

Erklärung

Ich versichere hiermit, dass ich diese Forschungsarbeit selbstständig durchgeführt und verfasst habe und dass ich keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Arbeit ist weder veröffentlicht worden noch hat sie in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegen. Weiterhin versichere ich, dass das elektronische Exemplar mit den anderen Exemplaren übereinstimmt.

Ort

Datum

Unterschrift

Kurzfassung/Abstract

Kurzfassung

Die steigende Komplexität von Energiesystemmodellen resultiert in steigenden Laufzeiten der Optimierungsprobleme, was eine Betrachtung von Beschleunigungsmethoden unerlässlich macht. Diese Forschungsarbeit untersucht mehrere Beschleunigungsmethoden in REMix (Renewable Energy Mix), darunter räumliche Aggregation, rollierende Einsatzplanung (*Rolling Horizon*) und Veränderungen in der Modellierungssprache GAMS in Bezug auf eine Reduktion der Rechenzeiten unter Berücksichtigung der Veränderungen der Modellierungsgenauigkeit. Zur Anwendung kommen zwei Modelle, die sich in räumlicher Konfiguration unterscheiden und variablen zeitlichen Parametrierungen unterzogen werden. Die räumliche Aggregation zeigt bei 10 % Abweichung des Zielwerts eine Reduktion der Rechenzeit um bis zu 50 %, die *Rolling Horizon*-Methode zeigt bei ähnlicher Reduktion der Rechenzeit eine Abweichung um knapp 30 %. Die Umformulierungen des GAMS-Code zeigen keine allgemeinen Verbesserungen.

Abstract

The rising complexity of energy system models leads to rising computing times of optimisation problems, which makes the evaluation of possible acceleration strategies necessary. This work deals with the evaluation of spatial aggregation techniques, a *Rolling Horizon*-method and changes to the code of the modelling language (GAMS) regarding a reduction in computational time while considering the changes in the optimal value. Two models with different spatial characteristics are utilised and the temporal configuration is varied. The spatial aggregation shows a reduction in computing time of up to 50 % with changes to the optimal value below 10 %. The *Rolling Horizon*-method shows similar reductions of computing time, but greater changes to the optimal value (almost 30 %) are observed. Changes to the modelling software generally don't show improvements in computing time.

Inhaltsverzeichnis

Erklärung	i
Kurzfassung/Abstract	ii
Abbildungsverzeichnis	iv
Tabellenverzeichnis	vi
Abkürzungen und Begriffe	vii
1 Einleitung	1
2 Methodik	2
2.1 Spezifizierung der verschiedenen Phasen	3
2.2 Variation der CPLEX-Zeit	4
3 Ergebnisse	8
3.1 Reduktion des Problems	8
3.1.1 Räumliche Skalierung	8
3.1.2 Verringerung der Modulauswahl	23
3.2 Rolling Horizon	26
3.3 Set-Order	33
4 Zusammenfassung und Ausblick	42
Literatur	44
Anhang	45

Abbildungsverzeichnis

2.1	Streuung der <i>CPLEX</i> -Zeit und beiden <i>GAMS</i> -Zeiten für Erstellung und Ausführung bei 10 Durchläufen des gleichen Optimierungsproblems	5
2.2	Streuung der <i>CPLEX</i> -Zeit und beiden <i>GAMS</i> -Zeiten für Erstellung und Ausführung bei 421 Durchläufen des gleichen Optimierungsproblems und Variation des <i>Seed</i> -Parameters von <i>CPLEX</i>	6
3.1	Verlauf von Zielwert und <i>CPLEX</i> -Zeit im <i>Transgrid2</i> -Projekt über der Anzahl an Clustern. Der absolute Wert der jeweiligen Referenzgröße ist in der Legende eingeklammert.	10
3.2	Verlauf weiterer Kennwerte für Komplexität und Leitungsausbau; die Bezugswerte des Referenzfalls sind in der Legende eingeklammert.	11
3.3	Verlauf von Zielwert und <i>CPLEX</i> -Zeit im <i>Transgrid2</i> -Projekt über der Anzahl an Clustern. Absolute Größen sind der Legende entnehmbar.	13
3.4	Verlauf weiterer Kennwerte für Komplexität im <i>Transgrid2</i> -Projekt (90 d)	14
3.5	Verlauf von Zielwert und <i>CPLEX</i> -Zeit über der Anzahl an Clustern. Absolute Größen sind der Legende entnehmbar. (365 d/12 h)	16
3.6	Verlauf weiterer Kennwerte für Komplexität und Leitungsausbau	17
3.7	Ausschnitt aus Arbeit [7]: <i>CPLEX</i> -Zeit über Anzahl an Clustern	18
3.8	Vergleich der <i>CPLEX</i> -Zeiten im <i>INTEEVER</i> - und <i>Transgrid2</i> -Projekt mit jeweiligen Näherungspolynomen. Beide Achsen sind auf jeweilige Referenzfälle normiert.	19
3.9	Aufsummierte Jahresenergiemenge aller Knoten nach Kraftwerkstyp bei 90 Tagen als Zeithorizont	22
3.10	<i>CPLEX</i> -Zeit für minimale und maximale Modulauswahl bei Variation der räumlichen und zeitlichen Auflösung	24
3.11	Erstellungszeit für minimale und maximale Modulauswahl bei Variation der räumlichen und zeitlichen Auflösung	25
3.12	Verlauf der Gesamtzeit im H ₂ -Szenario bezogen auf den Referenzfall ohne RH bei Variation der RH-Intervalle für zeitliche Auflösungen von 6- und 12-Stunden. Zusätzlich wird der Zielwert gezeigt, der ebenso auf den Referenzfall normiert ist. Die Zeitdauer beträgt ein Jahr, die Überlappung zwischen den RH-Intervallen 10 %.	28
3.13	Anteil der <i>GAMS</i> -Zeit an der Gesamtzeit über der Anzahl der RH-Intervalle.	29
3.14	<i>GAMS</i> -Zeit über <i>CPLEX</i> -Zeit bei Variation der <i>Rolling Horizon</i> -Intervalle.	30
3.15	Auf Referenzfall (ohne <i>Rolling Horizon</i>) bezogene Kennwerte bei 365 d/2 h	31

3.16	Veränderungen durch neue <i>Set-Order</i> bei den Szenarien <i>EHighwaySM</i> und <i>DG0-FC0</i> mit 12- und 6-stündlicher Auflösung (365 d) bezogen auf Revision r20778. Dabei wurden die g00-Dateien mit <i>GAMS 24.3</i> erstellt und unter <i>GAMS 24.7</i> ausgeführt.	35
3.17	Veränderungen durch Revision r20911 bei den Szenarien <i>EHighwaySM</i> und <i>DG0-FC0</i> mit 12, 6- und 2-stündlicher Auflösung (365 d). Die Nichtnulleinträge weisen keine Unterschiede auf und werden in der unteren Teilgrafik nicht dargestellt. (<i>GAMS-Version 23.8</i>)	37
3.18	Veränderungen durch neue <i>Set-Order</i> bei den Szenarien <i>EHighwaySM</i> und <i>DG0-FC0</i> gemittelt über verschiedene zeitliche Konfigurationen bezogen auf Revision r20778. (<i>GAMS-Version 24.8</i>)	38

Tabellenverzeichnis

2.1	Relative Standardabweichung von CPLEX-Zeit, Erzeugungszeit und Ausführungszeit	4
2.2	Relative Standardabweichung von Rechenzeit, Erzeugungszeit und Ausführungszeit bei Variation des <i>Seed</i> -Parameters	7
3.1	Gewählte Parameter für Clustering im <i>Transgrid2</i> -Projekt	9
3.2	Ausgabe der verschiedenen <i>GAMS</i> -Zeitdauern bei 4 und 64 Intervallen	27
3.3	Übersicht über Wahl von zeitlicher Auflösung, erstem und letztem Tag des Optimierungsproblem. Die Kombination von diesen Parametern ergibt (bei Auslassung von Fällen, in denen der Startzeitpunkt vor dem Endzeitpunkt liegt) insgesamt 18 Kombinationen.	36
3.4	Übersicht über die 10 längsten Teilschritte der <i>General Algebraic Modeling System, siehe [2]</i> (<i>GAMS</i>)-Zeit im <i>EHighwaySM</i> -Szenario (2h/r20911)	39
3.5	Übersicht über die 10 längsten Teilschritte der <i>GAMS</i> -Zeit im <i>EHighwaySM</i> -Szenario (2h/r20778)	40
3.6	Übersicht über die 10 längsten Teilschritte der <i>GAMS</i> -Zeit im <i>EHighwaySM</i> -Szenario (12h/r20911)	40
3.7	Übersicht über die 10 längsten Teilschritte der <i>GAMS</i> -Zeit im <i>EHighwaySM</i> -Szenario (12h/r20778)	41
4.1	Modulauswahl für minimale Module	45
4.2	Modulauswahl für maximale Module	46

Abkürzungen und Begriffe

Agglomerative Clustering Clusteringalgorithmus, der zusammenfassend hierarchische Cluster bildet, siehe <http://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html#sklearn.cluster.AgglomerativeClustering>, abgerufen am 28. April 2017 9

Barrier Barriere-Verfahren, auch Innere-Punkte-Verfahren genannt: Klasse von Algorithmen zur Lösung von linearen und quadratischer Optimierungsproblemen 3

CPLEX IBM ILOG CPLEX Optimization Studio, siehe [3] iv, 2–7, 9, 10, 12, 13, 15, 16, 18–21, 23, 24, 26, 27, 30, 32–34, 36, 39, 42, 43

DG0-FC0 Szenario mit geringer installierter Leistung und dementsprechend viel Ausbauroptimierung v, 33–39

DG80-FC80 Szenario mit geringer installierter Leistung, dementsprechend viel Ausbauroptimierung und Beschränkungen des Energieimports über Ländergrenzen 4

EHighwaySM Szenario mit hoher installierter Leistung und dementsprechend geringer Ausbauroptimierung v, vi, 26, 33–41

GAMS General Algebraic Modeling System, siehe [2] ii, iv–vi, viii, 1–7, 26, 27, 29, 30, 32–43

INTEEVER “Analyse von Strukturoptionen zur Integration erneuerbarer Energien in Deutschland und Europa unter Berücksichtigung der Versorgungssicherheit”, hier speziell ein Energiesystemmodell mit mehreren parametrisierten Szenarien, umfangreicher Modulauswahl und geographischer Abdeckung von Europa und Nordafrika iv, 1, 2, 15, 19–21, 26, 33, 42

K-Means Bewährter Clusteringalgorithmus; für eine Beschreibung der prinzipiellen Arbeitsweise siehe <http://scikit-learn.org/stable/modules/clustering.html#k-means>, abgerufen am 28. April 2017 8, 9

OMaT OptiMo Management Tool 2, 3

OptiMo Optimization Module 2, 3

REMix Renewable Energy Mix for sustainable energy supply 1–3, 8

Rolling Horizon Rollierende Einsatzplanung ii, iv, 1, 26, 27, 30–32, 42

Set-Order Reihenfolge der Mengen im *GAMS*-Code v, 1, 33–36, 38, 39, 41–43

Ticks CPLEX-internes Maß für den gesamten Rechenaufwand 4, 6, 12, 15, 21, 32, 34, 36

Transgrid2 Modell des deutschen Übertragungsnetzes auf Basis von [5] iv, vi, 2, 8–10, 13–15,
19–21, 42

1 Einleitung

Das Energiesystem in Deutschland unterliegt durch Maßnahmen wie Energiewende, Ausbau und Integration von Erneuerbaren Energien und potentiellen Flexibilisierungsoptionen wie Speicher und Lastmanagement großen Herausforderungen. Ähnliche Vorgänge sind europaweit zu beobachten.

Um diese Entwicklungen zu modellieren und untersuchen sind optimierende Energiesystemmodelle wie *Renewable Energy Mix for sustainable energy supply (REMIX)* ([8]) hilfreich. *REMIX* setzt auf eine lineare Optimierung des Energiesystems und kann durch eine modulare Bauweise verschiedene Technologien wie Mobilität, Wärmebedarf- und Erzeugung, verschiedene Speichertechnologien und Lastmanagement abbilden. Ein Überblick über *REMIX* und dessen Bestandteile wird in Abschnitt 2.1 auf Seite 3 gegeben. Die zeitliche und räumliche Auflösung ist dabei variabel, wobei bei der räumlichen Auflösung eine Aggregation erwünscht ist, um die steigende Knotenanzahl zu begrenzen, was unter anderem durch die Betrachtung von europaweiten Energiesystemen hervorgerufen wird. Herausforderungen bestehen dabei in der resultierenden Komplexität des Optimierungsproblems. Der steigende Umfang der Modellierung kombiniert mit einer räumlich und zeitlich feinen Auflösung bedingt Optimierungsprobleme, die einen Rechen- und Speicherbedarf aufweisen können, der ohne Hochleistungsrechner nicht lösbar ist.

Um die Rechenzeit zu begrenzen gibt es eine Vielzahl an Möglichkeiten. Darunter fallen unter anderem die Reduktion zeitlicher und räumlicher Auflösung, die Aufteilung in sequentiell zu lösende Zeitabschnitte (*Rollierende Einsatzplanung (Rolling Horizon)*) und Umformulierungen des Optimierungsproblems in der Modellierungssoftware (hier *GAMS*).

Bei der räumlichen Skalierung in Abschnitt 3.1.1 auf Seite 8 liegt der Fokus auf der Anwendung einer Aggregierungsmethode einer früheren Arbeit auf ein anderes Modell (*INTEEVER*), die Vervollständigung und mögliche Validierung der Ergebnisse.

In Abschnitt 3.1.2 auf Seite 23 wird die Auswirkung einer Verringerung der Modulauswahl untersucht.

Bei *Rolling Horizon* in Abschnitt 3.2 auf Seite 26 sollen ebenso die Ergebnisse einer früheren Arbeit bei Anwendung im *INTEEVER*-Modell untersucht werden.

Die *Set-Order*-Änderungen in Abschnitt 3.3 auf Seite 33 werden untersucht, um mögliche Beschleunigungen in der Modellierung zu evaluieren.

2 Methodik

In diesem Abschnitt wird die Vorgehensweise der weiteren Arbeit erläutert und einige Hintergründe beschrieben. Für umfangreiche Erläuterungen zu *REMix* sei unter anderem auf [8] verwiesen. *REMix* ist ein Energiesystemmodell und besteht unter anderem aus *Optimization Module (OptiMo)*. Dieses wiederum besteht aus *OptiMo Management Tool (OMaT)*, einer Software zur Modellkonfiguration und *GAMS*-Code-Erzeugung, der Modellierungssprache *GAMS* und dem Löser *CPLEX*. In *OMaT* besteht die Möglichkeit das Modell zu konfigurieren, also zum Beispiel zeitliche und räumliche Parameter festzulegen, die Modulauswahl anzupassen oder Optionen in den Modulen zu verändern, also z.B. den Leitungsausbau zuzulassen. Ausgabe von *OMaT* ist *GAMS*-Code, der dann die gewählten Parameter und impliziten (Un)gleichungen in die mathematische Formulierung umsetzt und an den Löser übergibt. Der Löser soll das Optimierungsproblem lösen und gibt die gefundene Lösung in gewünschter Form an *GAMS* zurück.

In dieser Arbeit wird als Löser nur *CPLEX* betrachtet, da dieser sich in *REMix* als vorteilhaftesten herausstellte [1]. Die Beschleunigung wird hier an Reduktionen der *CPLEX*-Zeit und gegebenenfalls an Reduktionen der *GAMS*-Zeit betrachtet. Dabei wird die Modellierungsgenauigkeit an Veränderungen des Zielwerts, hier die gesamten Systemkosten, berücksichtigt. Es werden zwei Modelle verwendet: *Transgrid2* und *INTEEVER*, wobei das *Transgrid2*-Modell zur Untersuchung der räumlichen Aggregation über einen weiten Rahmen und Vervollständigung vorhandener Ergebnisse dient und das *INTEEVER*-Modell in der restlichen Arbeit verwendet wird.

Nach Möglichkeit wird dabei die zeitliche Konfiguration variiert, um Abhängigkeiten der Parametrierung abzumindern und allgemeingültigere Ergebnisse zu erhalten. Die Erzeugung der Modelle und das Lösen dieser erfolgte mit einem sich in der Entwicklung befindenen Tools, das das Lösen vieler Modelle erleichtern soll, indem es das manuelle Ausführen durch *OMaT* ersetzt. In einem ersten Schritt werden alle Daten von *GAMS* in eine binäre Datei¹ zusammengefasst. Diese kann dann mit minimalem *GAMS*-Code auf anderen System ausgeführt werden. Ein erstelltes Skript liest die Kennwerte des Optimierungslaufes wie unter anderem Zielwert und Anzahl der Leitungen aus den Ergebnisdateien von *GAMS*² und Kennwerte über das Verhalten des Löser wie benötigte Zeit aus dessen Protokolldaten. Vorteile dieser Vorgehensweise bestehen in der Entkopplung von Erstellung und Ausführung der Modelle: *OMaT* bietet die Möglichkeit Modelle zu konfigurieren, die Softwarerevision zu verwalten und die Ausführung der Modelle zu starten, bedingt jedoch in der Praxis eine Beschränkung auf eine sequentielle Arbeitsweise, das heißt jede Rekonfiguration sollte erst nach dem Ende des aktuellen Optimierungsproblems stattfinden. Das Ausführen vieler Modelle ist damit eingeschränkt. Durch die entkoppelte Arbeitsweise wird

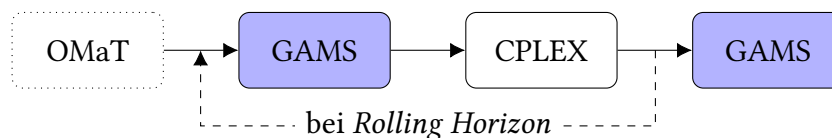
¹mit Dateiondung .g00

²mit Dateiondung .gdx

nur vergleichsweise wenig Zeit in *OMaT* benötigt und dadurch die Flexibilität bezüglich Rekonfiguration und Verändern der Softwareversion erhöht. Auch beim Ausführen auf verschiedenen Rechnern erleichtert die entkoppelte Arbeitsweise das Vorgehen.

2.1 Spezifizierung der verschiedenen Phasen

Hier soll ein Überblick über die verschiedenen Abschnitte eines Ablaufs in *REMix*, bzw. *OptiMo* gegeben werden. Prinzipiell besteht dieser aus mehreren Phasen:



Aufgaben dieser Phasen und Grenzen für mögliche Beschleunigungen dieser sind wie folgt:

1. (*OMaT*): Software zur Konfiguration des Modells
 - ändert die *GAMS*-Formulierung, falls notwendig (z. B. bei räumlicher Aggregation notwendig)
 - limitiert durch: Einlesen von Daten, nicht parallelisiert
2. *GAMS*:
 - limitiert durch: Einlesen von Daten, nicht parallelisiert
 - beeinflusst durch: Größe und Komplexität des Problems, Formulierung (siehe Abschnitt 3.3 auf Seite 33)
3. *CPLEX*: Die *CPLEX*-Phase besteht aus Presolve und Solve:
 - Presolve: Umformungen und Konfiguration des Lösers um die Komplexität zu verringern
 - beeinflusst durch: Größe des Problems, Struktur des Problems, gewählter Lösungsalgorithmus
 - nicht parallelisiert
 - Solve: Lösen des Optimierungsproblems
 - beeinflusst durch: Parametrierung, Grad der Vereinfachung durch Presolve, Lösungsalgorithmus
 - bei verwendetem *Barrier*-Algorithmus: weitgehende Parallelisierung gegeben
4. *GAMS*: Einlesen der von *CPLEX* gefundenen Lösung

- *PostCalc*: Ausgabe der gefundenen Lösung als „Punktlösung“, die in weiteren Läufen *CPLEX* als Warmstart übergeben werden kann
 - limitiert durch Einlesen der *CPLEX*-Lösung und Zusammenstellen der Ausgabeform

Bei Nutzung der *Rolling Horizon*-Methode wird der erste Aufruf von *GAMS* und folgendem *CPLEX*-Lösen wie angedeutet mehrmals ausgeführt. Nur der Teilabschnitt des eigentlichen Lösungsalgorithmus kann derzeit durch Parallelisierung beschleunigt werden, dies zeigt die Notwendigkeit der Betrachtung von Beschleunigungsmethoden auf. Veränderungen bei *CPLEX*-Zeit und *GAMS*-Zeit sind auch unter Berücksichtigung dieser Tatsache zu betrachten: Veränderungen bei nicht-parallelisierten Prozessen sind kritischer als bei parallelisierten, da diese direkt von einer ansteigenden Anzahl an Prozessorkernen profitieren können.

2.2 Variation der CPLEX-Zeit

In diesem Abschnitt soll die Deterministik des Löfers *CPLEX* und die Schwankungen der Modellierungssoftware *GAMS* untersucht werden. Für eine genauere Betrachtung sollte dies zusätzlich zu den Untersuchungen der Beschleunigungsmethoden durchgeführt werden, aus Zeitgründen ist diese Betrachtung in dieser Arbeit vorangestellt und nur exemplarisch durchgeführt worden.

Abb. 2.1 auf der nächsten Seite zeigt die Schwankungsbreite der *CPLEX*-Zeit des *DG80-FC80*-Szenarios bei der Dauer eines Jahres und einer zeitlichen Auflösung von 12 h. Tabelle 2.1 zeigt dazu die relativen Standardabweichungen. Alle Einstellungen sind dabei über die Läufe unverändert. Laut IBM ist der verwendete Barrier Algorithmus seit Version 12.3³ deterministisch⁴, was an der über alle Läufe gleichen Anzahl an *Ticks* ($5,571 \cdot 10^6$) ersichtlich ist. Die relative Standardabweichung von 0,43 % kann für die weitere Arbeit als Minimum für Abweichungen der dargestellten *CPLEX*-Rechenzeiten angenommen werden, da hier die sonstige Systemaktivität gering war und nicht alle Kerne der CPU benutzt wurden. Im Gegensatz hierzu ist die Abweichung bei der Erzeugung des Modells (*Erstellungszeit*) mit fast 10 % deutlich höher, insbesondere kann es hier deutliche Ausreißer nach oben geben. Diese Zeit macht jedoch nur einen Teil der gesamten *GAMS*-Zeit aus, die gesamte *GAMS*-Zeit (hier: *Ausführungszeit*) zeigt nur geringe Streuungen.

Tabelle 2.1: Relative Standardabweichung von *CPLEX*-Zeit, Erzeugungszeit und Ausführungszeit

Dauer	relative Standardabweichung/%
<i>CPLEX</i> -Zeit/s	$430 \cdot 10^{-3}$
Erstellungszeit/s	9,360
Ausführungszeit/s	1,030

³Die älteste verwendete *GAMS*-Version ist 24.3, die *CPLEX* in Version 12.6 und aktueller beinhaltet.

⁴https://www.ibm.com/support/knowledgecenter/en/SSSA5P_12.3.0/ilog.odms.studio.help/Content/Optimization/Documentation/Optimization_Studio/_pubskel/ps_COS_Eclipse201.html, abgerufen am 28. April 2017

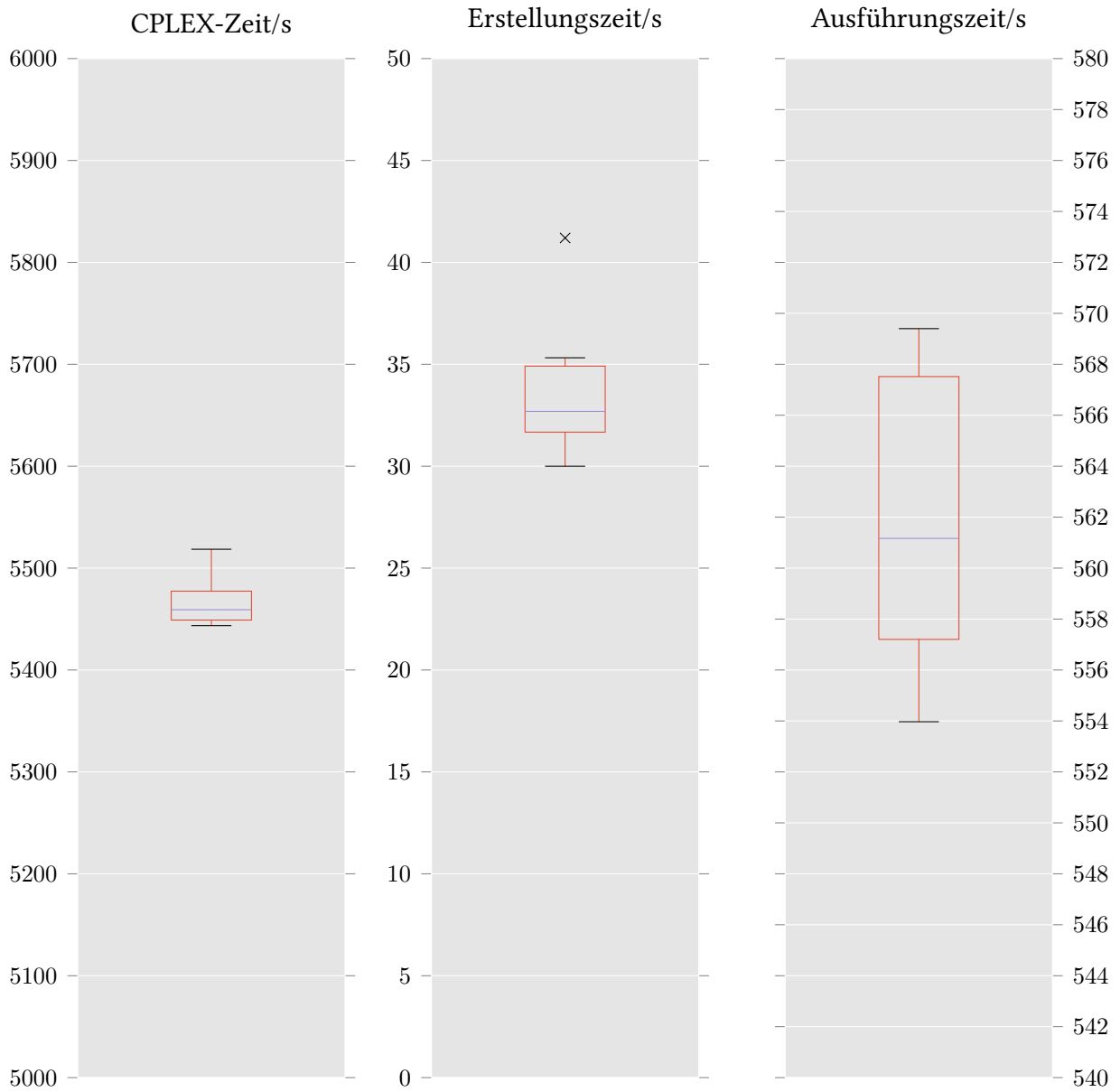


Abbildung 2.1: Streuung der *CPLEX*-Zeit und beiden *GAMS*-Zeiten für Erstellung und Ausführung bei 10 Durchläufen des gleichen Optimierungsproblems

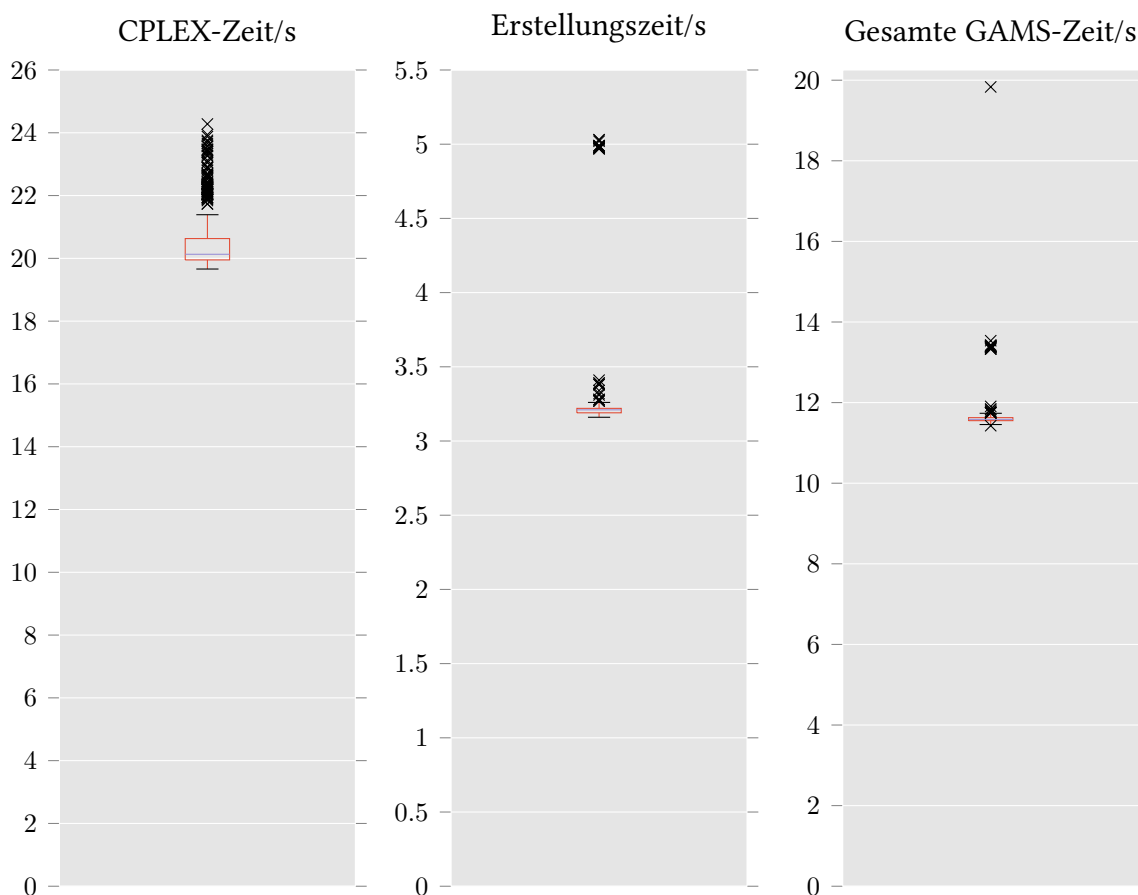


Abbildung 2.2: Streuung der *CPLEX*-Zeit und beiden *GAMS*-Zeiten für Erstellung und Ausführung bei 421 Durchläufen des gleichen Optimierungsproblems und Variation des *Seed*-Parameters von *CPLEX*

Abb. 2.2 und Tabelle 2.2 auf der nächsten Seite zeigen das Verhalten bei einem kleineren Optimierungsproblem. Außerdem wurde *CPLEX* eine zufällig gewählte ganzzahlige positive Zahl als *Seed*-Parameter übergeben. Dies soll laut Hersteller die “Vielfältigkeit” der Lösungen erhöhen⁵.

In der Tat zeigt sich hier eine höhere Streuungsbreite der Rechenzeit. Bei der Betrachtung der Rechenzeiten über verschiedene *CPLEX*-Versionen und Plattformen (z.B. unterschiedliche Betriebssysteme) muss also eine höhere Streuung der Rechenzeit berücksichtigt werden. Das *CPLEX*-interne Maß für die Komplexität des Optimierungsproblems (*Ticks*) ist dabei identisch ($16,503 \cdot 10^3$). Die Streuung der Erstellungszeit verhält sich ähnlich wie im zuerst gezeigten Beispiel, die gesamte *GAMS*-Zeit zeigt einen deutlichen Ausreißer, der zu einer höheren relativen Standardabweichung von 4,43 % führt.

Soweit hilfreich wird in der restlichen Arbeit somit bei Betrachtungen der Rechenzeit die Entwicklung der *Ticks* betrachtet, da diese nicht die hier betrachteten Schwankungen aufweisen.

⁵https://www.ibm.com/support/knowledgecenter/en/SS9UKU_12.5.0/com.ibm.cplex.zos.help/Parameters/topics/RandomSeed.html, abgerufen am 28. April 2017

Tabelle 2.2: Relative Standardabweichung von Rechenzeit, Erzeugungszeit und Ausführungszeit bei Variation des *Seed*-Parameters

Dauer/s	relative Standardabweichung/%
<i>CPLEX</i> -Zeit	5,030
Erstellungszeit	9,820
Gesamte <i>GAMS</i> -Zeit	4,430

3 Ergebnisse

3.1 Reduktion des Problems

3.1.1 Räumliche Skalierung

In diesem Abschnitt soll die räumliche Skalierung in zwei Modellen untersucht werden. Räumliche Skalierung bzw. Aggregation meint das Zusammenfassen von Knoten im Modell, was die Komplexität durch Wegfall von Netzbeschränkungen stark reduziert. Extrempunkte sind einerseits die Bedingungen in der Realität, wobei fast jeder Erzeuger, Verbraucher, Speicher etc. eine unterschiedliche geographische Lage aufweist und sich nur eingeschränkt gegenseitig austauschen kann und die so genannte “Kupferplatte”, bei der alle Elemente an einem gedachten Ort angesiedelt werden, wobei keine Netzbeschränkung mehr berücksichtigt wird. In Optimierungsmodellen beträgt die Anzahl der Knoten typischerweise einige Dutzend bis einige Hundert, die Anwendung bei mehr Knoten ist derzeit zu komplex. Ausgangspunkt ist hierbei die Arbeit von Johannes Metzdorf [7], der eine Methode zum Aggregieren von Netzknoten unter Berücksichtigung der Netztopologie (*spectral clustering*) für die Anwendung mit *REMix* entwickelte. *Spectral* bezieht sich dabei auf die Verwendung von Laplace-Matrizen im Algorithmus, die Information aus der Netztopologie und den Netzkenndaten enthalten¹.

Transgrid2

Transgrid2 bezeichnet in dieser Arbeit ein Modell des deutschen Übertragungsnetzes bestehend aus 505 Knoten, das auf Daten aus [5] basiert.

Für die Parametrierung des Clusterings wurden die von Johannes Metzdorf [7] als optimal² identifizierten Werte ausgewählt, siehe dazu Tabelle 3.1 auf der nächsten Seite.

Beim Ausführen des Clustering-Skripts zeigt sich, dass ab einer Anzahl von 100 Clustern, *K-Means* als Clustering-Algorithmus nicht mehr praxistauglich ist. Durch die Konsistenzüberprüfung, die hauptsächlich aus dem Berechnen von Eigenwerten besteht, und den in *K-Means* intern benötigten Iterationen steigt die Rechenzeit stark an, so dass ein Durchlauf eines Clusterings länger als 24 h benötigt. Außerdem gelingt es *K-Means* immer seltener konsistente Cluster zu finden. Konsistente

¹Im Fall der ungewichteten Laplacematrix entspricht diese weitestgehend der Knotenpunktadmittanzmatrix.

²Optimal meint hierbei das Verhalten der von Johannes Metzdorf [7] gewählten Bewertungskennzahl $r(k)$, die die Zahl der durch Aggregation entfallenen überlasteten Leitungen charakterisiert als auch die Konsistenz der entstehenden Cluster.

Tabelle 3.1: Gewählte Parameter für Clustering im *Transgrid2*-Projekt

Parameter	Wert	Beschreibung
use_case	lm_pp_wp	Zeitpunkt mit geringer Last und hoher PV- und Windeinspeisung
t	6158h	Zeitpunkt, an dem die Grenzkosten der Knoten und die Netzauslastung bestimmt wird
edge_weight	reciprocal	Funktion zur Bestimmung der Gewichte zwischen den Knoten
Laplacian matrix	L	Art der Laplace-Matrix, die die Graphenstruktur enthält (hier: ungewichtete Laplacematrix)

Cluster bestehen aus Knoten, die alle miteinander verbunden sind, bzw. inkonsistenten Cluster enthalten Knoten, die nur über Knoten eines anderen Clusters verbunden sind ([7, siehe Abschnitt 2.3.2]). Damit ist die praktische Anwendbarkeit für den Bereich der Aggregationen mit mehr als 100 Clustern stark eingeschränkt. Als Alternative zeigt sich, dass durch einen Austausch von *K-Means* durch eine hierarchische Clustering-Methode (*Agglomerative Clustering*) die Rechenzeit auch bei steigender Anzahl an Clustern nicht stark ansteigt und im einstelligen Minutenbereich verbleibt. Auch erzielt das agglomerative Clusterverfahren größtenteils konsistente Cluster, was *K-Means* auch bei einem Vielfachen der Rechenzeit nicht gelingt. Agglomerative Clusterverfahren können dabei als Baumstruktur gesehen werden, bei denen ausgehend von den Blättern mit einem Element pro Cluster immer mehr Cluster zusammengefasst werden, bis sich alle Elemente in einem Cluster befinden. Die gewünschte Anzahl an Cluster kann dann durch Schnitt des Baumes erfolgen.

Bei diesem Verfahren wird für die Berechnung von k Clustern die Berechnung für l Cluster (mit $l > k$) benötigt, was sowohl die begrenzt wachsende Rechenzeit erklärt, als auch die Eigenschaft der Stabilität von einmal gebildeten Clustern mitbringen sollte.

30 Tage mit zeitlicher Auflösung von 1h In diesem Abschnitt soll die räumliche Aggregation an einem Optimierungsproblem mit Zeitfenster von 30 Tagen um den Zeitpunkt aus Tabelle 3.1 untersucht werden.

Abb. 3.1 auf der nächsten Seite zeigt den Verlauf des Zielwerts und der *CPLEX*-Zeit von 100 bis 499 Clustern. Dabei sind beide Werte auf den jeweiligen Wert des Referenzfalls bei 499 Clustern bezogen. Es zeigt sich ein annähernd linearer Verlauf der *CPLEX*-Zeit mit starken Schwankungen ab 350 Clustern. Innerhalb kleiner Abschnitte zeigen sich rein lineare Verläufe (z. B. von 370 bis 390), die von großen Sprüngen und Bereichen negativer Steigung unterbrochen werden.

Der Zielwert zeigt einen Anstieg bis 290 Clustern, dann einen Sprung zu 300 Clustern und anschließend einen nahezu konstanten Verlauf bis bei 480 und 490 Clustern eine erneuter Sprung auftritt. Dieser korreliert mit dem Sprung der *CPLEX*-Zeit, die hier 20 % über dem Referenzfall liegt. In dieser Konfiguration ist ein Clustering hauptsächlich im Bereich unterhalb 300 Clustern

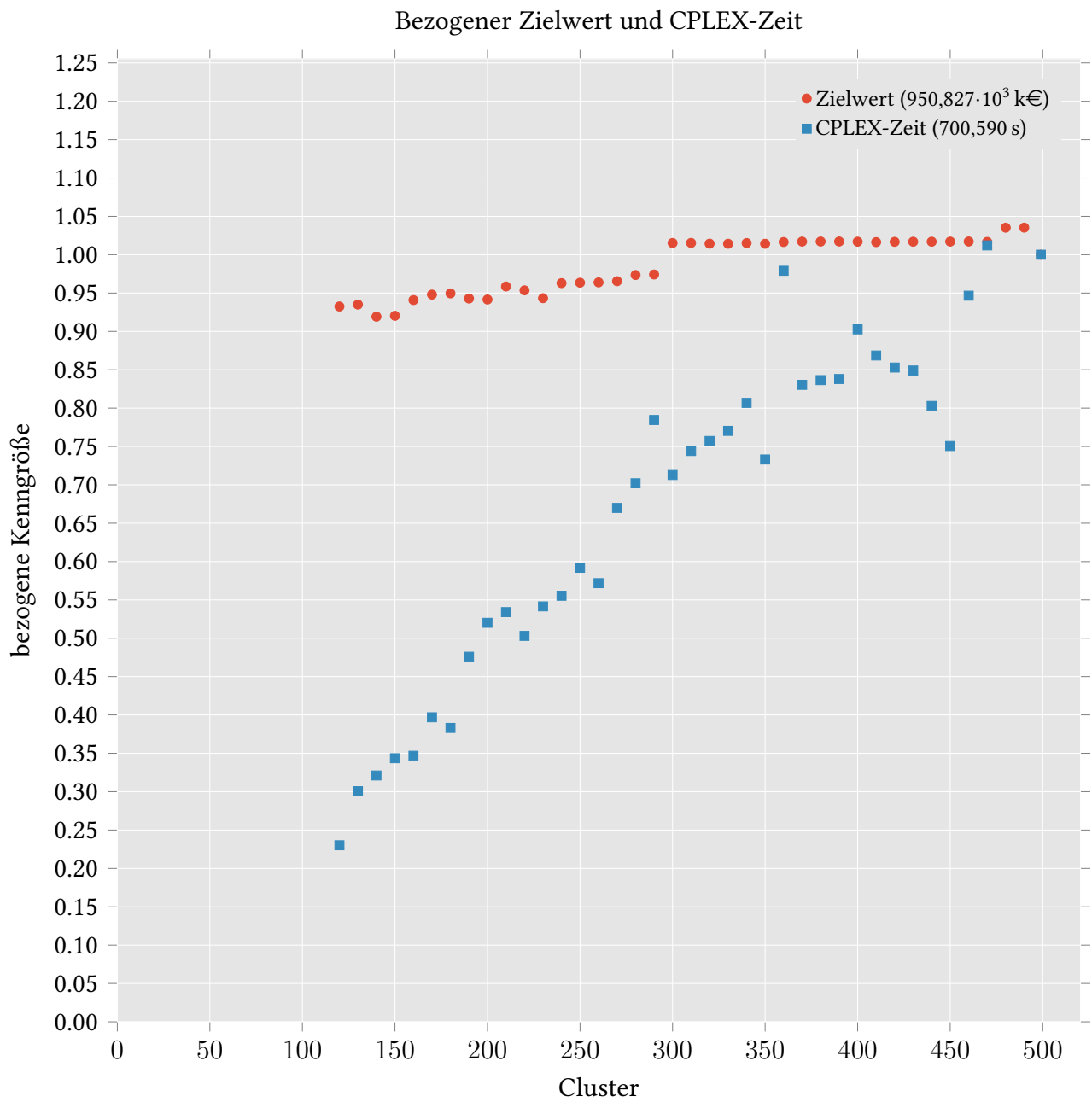


Abbildung 3.1: Verlauf von Zielwert und CPLEX-Zeit im *Transgrid2*-Projekt über der Anzahl an Clustern. Der absolute Wert der jeweiligen Referenzgröße ist in der Legende eingeklammert.

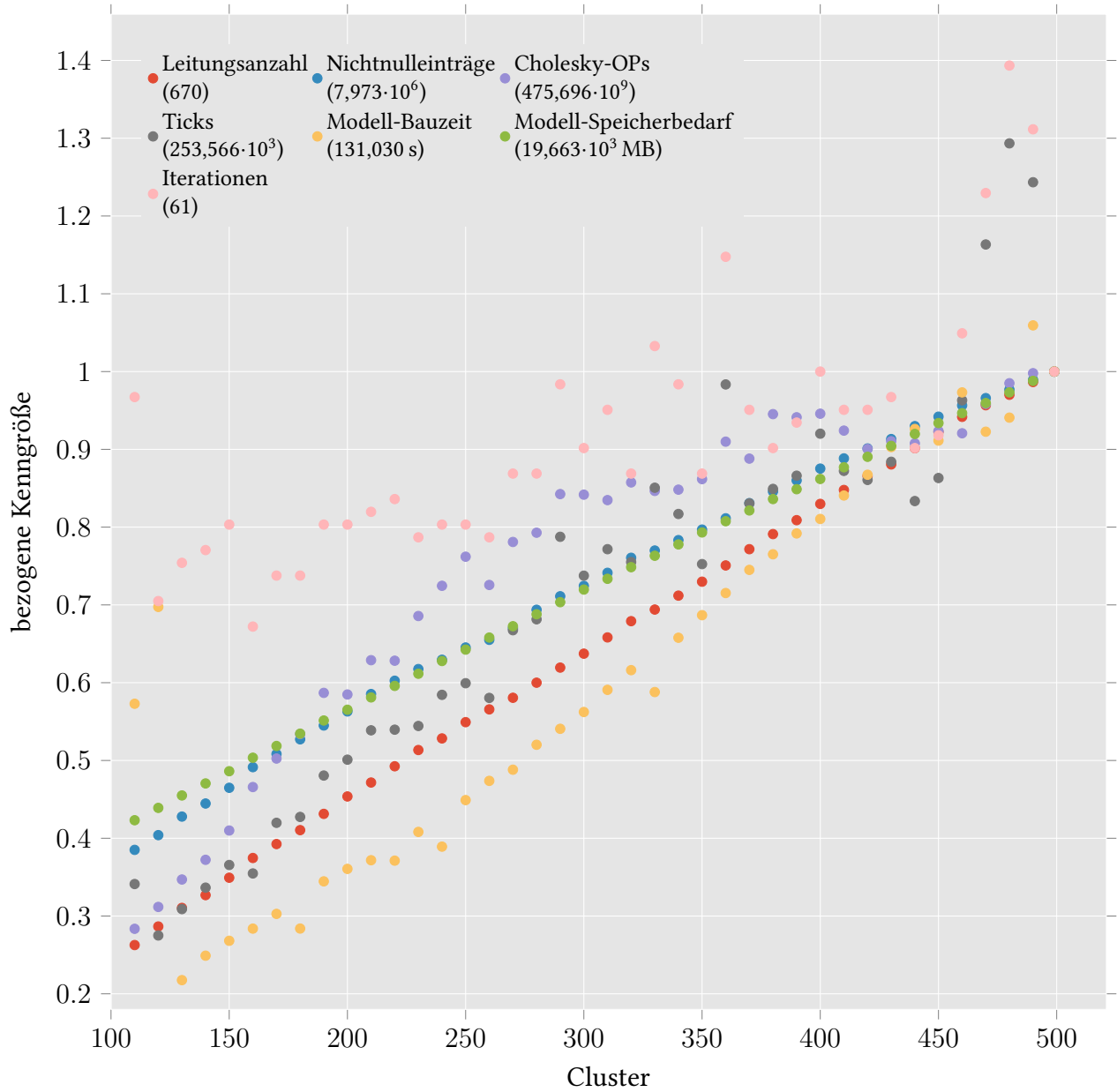


Abbildung 3.2: Verlauf weiterer Kennwerte für Komplexität und Leitungsausbau; die Bezugswerte des Referenzfalls sind in der Legende eingeklammert.

interessant, da hier der Verlauf des Zielwerts auf ein sinnvolles Clustering und eine deutliche Reduktion der *CPLEX*-Zeit hinweist. Bei 5% Abweichung des Zielwerts ist so zum Beispiel eine Halbierung der Rechenzeit möglich (im Bereich von 180 bis 200 Clustern).

Abb. 3.2 auf Seite 11 zeigt weitere Kenngrößen zu den Läufen aus Abb. 3.1 auf Seite 10. Aus dem gleichmäßigen Verlauf der Anzahl der Leitungen lassen sich keine Rückschlüsse auf vermeintlich schlecht gewählte Cluster bilden. Ebenso verlaufen die Nichtnulleinträge und der Modellspeicherbedarf über den kompletten Bereich des Clusterings sehr gleichmäßig und zeigen keine Korrelation mit den Ausschlägen der *CPLEX*-Zeit in Abb. 3.1 auf Seite 10. Die *Ticks*, die ein *CPLEX*-spezifisches Maß für den benötigten Rechenaufwand sind, zeigen einen ähnlichen Verlauf zur *CPLEX*-Zeit in Abb. 3.1 auf Seite 10. Insbesondere zeichnet sich hier ab 450 Clustern ein starker Anstieg der Rechenzeit ab, der allerdings schon mit dem letzten Clustering (490) wieder abnimmt. Das Maximum liegt hier mit 130 % im Vergleich zur Referenz noch höher als das der *CPLEX*-Zeit. Im mittleren Bereich (200–400 Cluster) zeigt die Entwicklung der Cholesky-OPs³ einen Bauch, der teilweise mit den benötigten *Ticks* übereinstimmt. Hier steigt bei linear wachsender Matrix der Rechenaufwand pro Iterationsschritt stärker an, was auf Unterschiede der Parametrierung zurückzuführen ist.

Die Entwicklung der benötigten Iterationsanzahl zeigt einen Anstieg bis zum Referenzpunkt auf, der allerdings von starken Sprüngen durchsetzt ist, welche ähnlichen Verlauf zu den *Ticks* aufweisen, aber noch deutlicher ausfällt. Verglichen zum Modellspeicherbedarf steigt die Modell-Bauzeit steiler und ungleichmäßiger an, was auf die höhere Abhängigkeit zur restlichen Systemaktivität hinweist. Interessant ist hier, dass bei z. B. 200 Clustern nur noch 35 % der Zeit des Referenzfalls benötigt werden, während der Modell-Speicherbedarf äquivalent wie die *CPLEX*-Zeit halbiert ist.

90 Tage mit zeitlicher Auflösung von 1h Im Folgenden soll die räumliche Aggregation mit mehr Zeitschritten gezeigt werden, um eventuelle Schwankungen bei geringen Zeitdauern zu vermeiden. Der Unterschied zu Abschnitt 3.1.1 auf Seite 9 besteht im Zeitfenster, das nun 90 Tage um den Zeitpunkt aus Abschnitt 3.1.1 auf Seite 8 aufgespannt ist. Die zeitliche Auflösung von 1 h bleibt identisch.

Abb. 3.3 auf der nächsten Seite zeigt den bezogenen Zielwert und die *CPLEX*-Zeit. Der Verlauf des Zielwerts entspricht qualitativ dem Verlauf bei 30 Tagen. Die *CPLEX*-Zeit verhält sich unterschiedlich: von 100–300 zeigt sie einen steileren und gleichmäßigeren Verlauf. Bei 200 Clustern wird nur noch 35 % der Rechenzeit des Referenzfalls benötigt. Ab 300 Clustern flacht sich die Kurve ab, bis sie ab 450 Clustern wieder stark zu steigen beginnt. Hier liegt das Maximum sogar bei ca. 175 % der Rechenzeit des Referenzfalls. Der zweite Punkt bei 499 Clustern, der oberhalb der Referenz liegt, zeigt den Referenzfall mit einem Transportmodul, das die schon bei voller Knotenanzahl vorliegenden parallelen Leitungsabschnitte nicht aggregiert.

Abschnitt 3.1.1 auf Seite 14 zeigt wie schon Abb. 3.2 auf Seite 11 den Verlauf weiterer Kenngrößen.

³Es ist die in *CPLEX* `cholesky_fp_ops` genannte Zahl gemeint, die die Anzahl der Gleitkommaoperationen der in jedem Iterationsschritt nötigen Cholesky-Zerlegung darstellt.

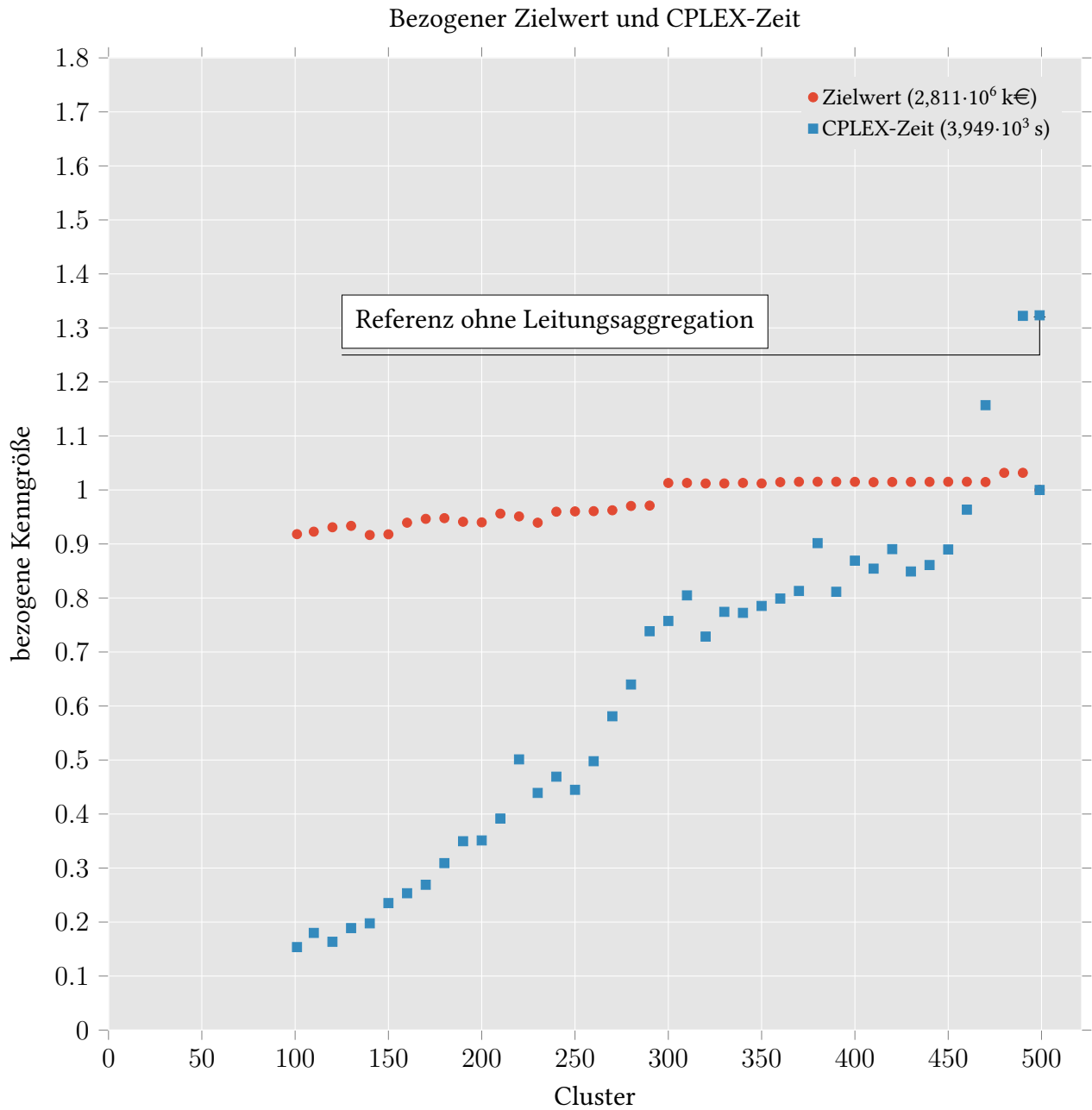


Abbildung 3.3: Verlauf von Zielwert und *CPLEX*-Zeit im *Transgrid2*-Projekt über der Anzahl an Clustern. Absolute Größen sind der Legende entnehmbar.

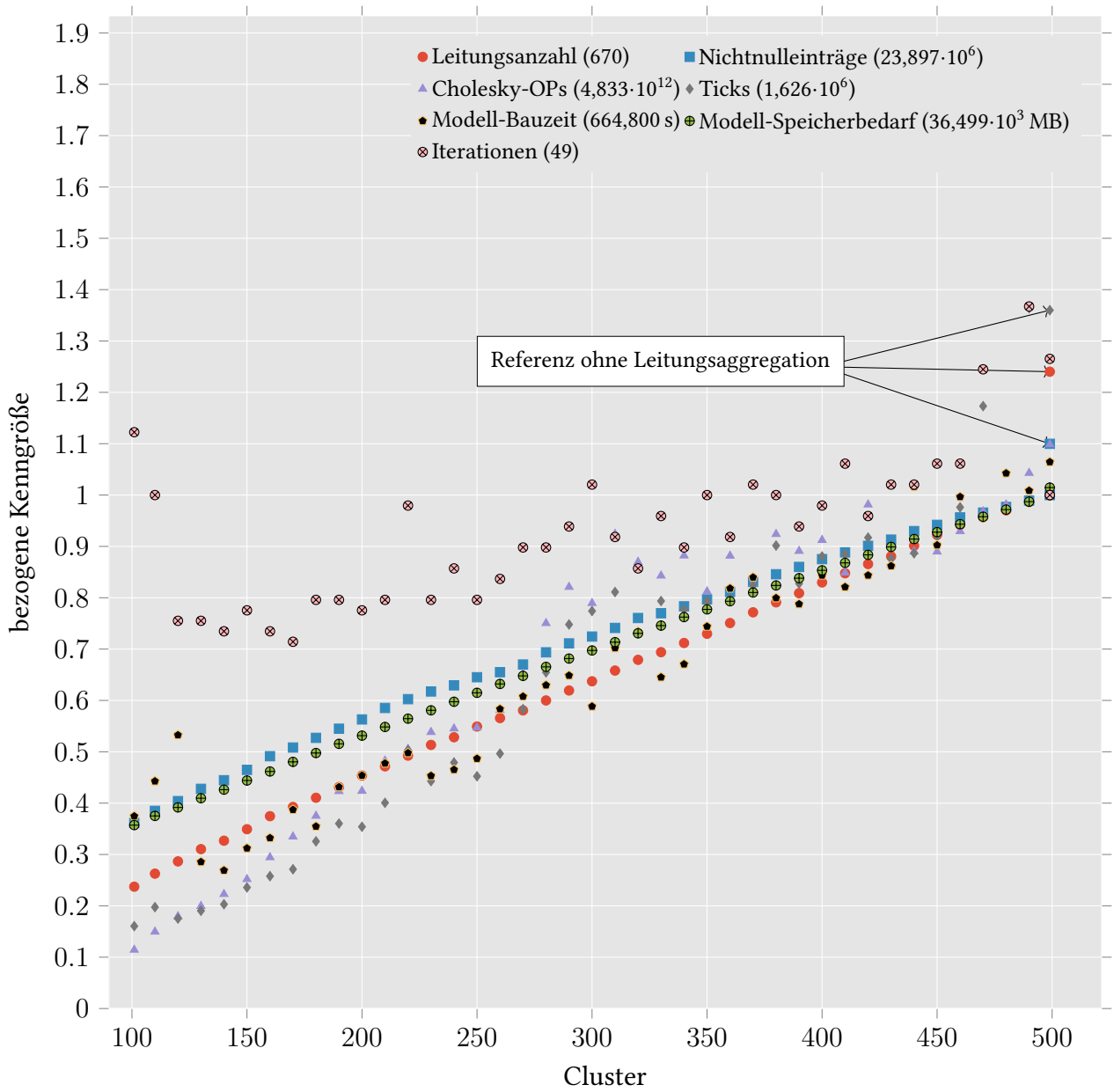


Abbildung 3.4: Verlauf weiterer Kennwerte für Komplexität im *Transgrid2*-Projekt (90 d)

Insbesondere finden sich hier bei 499 Clustern oberhalb des Referenzpunkts die Kenngrößen für die Benutzung des Transportmoduls ohne Leitungsaggregation. Hier lässt sich isoliert die Wirkung der Leitungsaggregation beobachten: ca. 25 % mehr Leitungen vergrößern die komplette Matrix um 10 % und erhöhen den Rechenaufwand um 35 %. Bei 110 Clustern zeigt sich, dass die Minima von *Ticks* und Cholesky-OPs unterhalb denen aus Abschnitt 3.1.1 auf Seite 9 liegen, bei größeren Problemen sorgt eine räumliche Aggregation also für relativ mehr Einsparungen. Bei den absoluten Werten bei beiden Referenzläufen zeigt sich, dass eine Verdreifachung der zeitlichen Komplexität die Nichtnulleinträge verdreifacht, die Modell-Bauzeit verfünffacht und den Rechenaufwand gemessen an *Ticks* um ca. 641 % erhöht. Auffällig ist, dass die Anzahl der benötigten Iteration im Referenzfall von 61 auf 49 sinkt und die Schwankungsbreite ebenfalls leicht sinkt.

INTEEVER

In diesem Abschnitt soll das Verfahren aus Abschnitt 3.1.1 auf Seite 8 für das *INTEEVER*-Projekt untersucht werden. *INTEEVER* besteht dabei im Vergleich zu *Transgrid2* aus mehr Modulen, Modulen mit Flexibilisierungsoptionen und der Möglichkeit für Leitungsausbau.

365 Tage mit zeitlicher Auflösung von 12h Abb. 3.5 auf der nächsten Seite zeigt den Verlauf von Zielwert und *CPLEX*-Zeit im *INTEEVER*-Projekt. Der Zielwert zeigt mehrere lineare Bereiche auf, die von deutlichen Sprüngen voneinander getrennt sind. Auch fällt der Zielwert bei wenigen Clustern auf fast 10 % des Referenzwertes. Die *CPLEX*-Zeit zeigt ab 15 Clustern einen steilen und sehr ungleichmäßigen Anstieg auf. Die Sprünge in der Kostenfunktion scheinen dabei nicht mit der *CPLEX*-Zeit in Verbindung zu stehen. Eine Halbierung der Rechenzeit ist hier schon unterhalb ungefähr 45 Clustern möglich.

Abb. 3.6 auf Seite 17 zeigt den Verlauf von Komplexitätskenngrößen sowie die Anzahl der Leitungen. Der Leitungsausbau zeigt einen weitgehend gleichmäßigen linearen Verlauf, ebenso verhalten sich der Modell-Speicherbedarf und die Modell-Bauzeit. Die Anzahl der Nichtnulleinträge zeigt einige Male einen “Knick” (z.B. bei 10–11 Cluster), der sich auch in den *Ticks* niederschlägt. Die *Ticks* zeigen bis 50 Cluster einen meist stetigen, nicht-linearen Anstieg, bis es von 50 auf 51 einen Sprung gibt und schließlich von 52 auf 53 Cluster eine Verdopplung des Rechenaufwands stattfindet. Ähnlich verhält sich der Verlauf der Cholesky-OPs, der bis 52 Cluster unterhalb der 30 %-Marke verharrt, dann verdoppelt und in den letzten Clusterings hin zur Referenz ansteigt. Bis zum Referenzpunkt gibt es schließlich einen geringen Überschwinger. Auffällig ist, dass bei ähnlichem Aggregationsgrad⁴ die Einsparung an Rechenaufwand (über 95 %) signifikant höher ist als im *Transgrid2*-Projekt. Dies weist auf die erhöhte Komplexität des *INTEEVER*-Projekts hin. Die Verdopplung des Rechenaufwand schlägt sich, wie in Abb. 3.1 auf Seite 10 gesehen, nur in gerinerem Ausmaße in der *CPLEX*-Zeit nieder, sie springt um ca. 20 Prozentpunkte. Die Kenngröße, die hier dem ungleichmäßigen Verlauf der *CPLEX*-Zeit ähnelt, ist die Iterationsanzahl. Aureißer nach oben und unten scheinen hier mit entsprechenden Änderungen der *CPLEX*-Zeit

⁴ $\frac{\text{Anzahl der Cluster}}{\text{Anzahl der zu aggregierenden Elemente}} \simeq 20 \%$

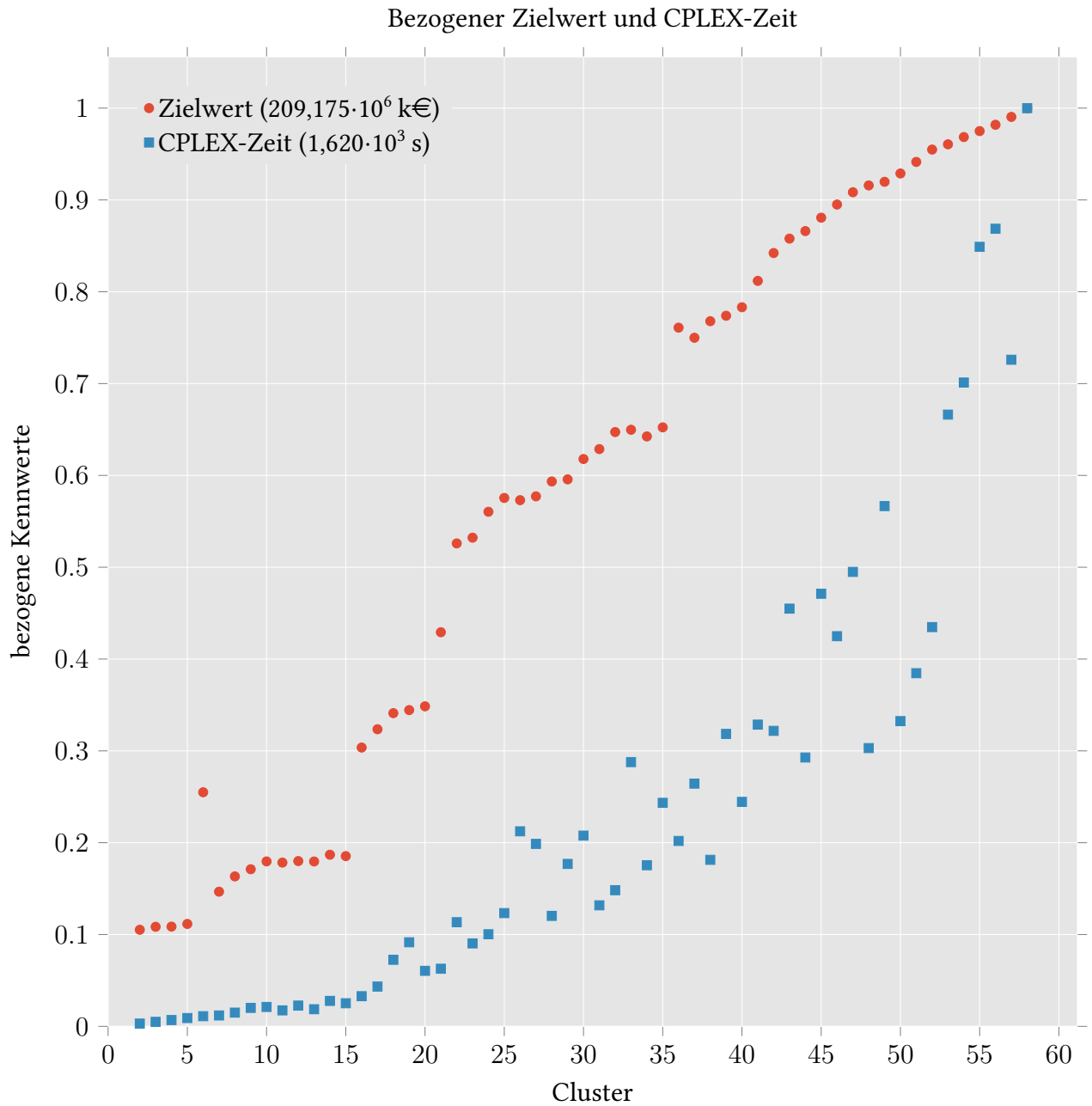


Abbildung 3.5: Verlauf von Zielwert und CPLEX-Zeit über der Anzahl an Clustern. Absolute Größen sind der Legende entnehmbar. (365 d/12 h)

Bezogene Kennwerte im Inteever-Projekt (365 d/12 h)

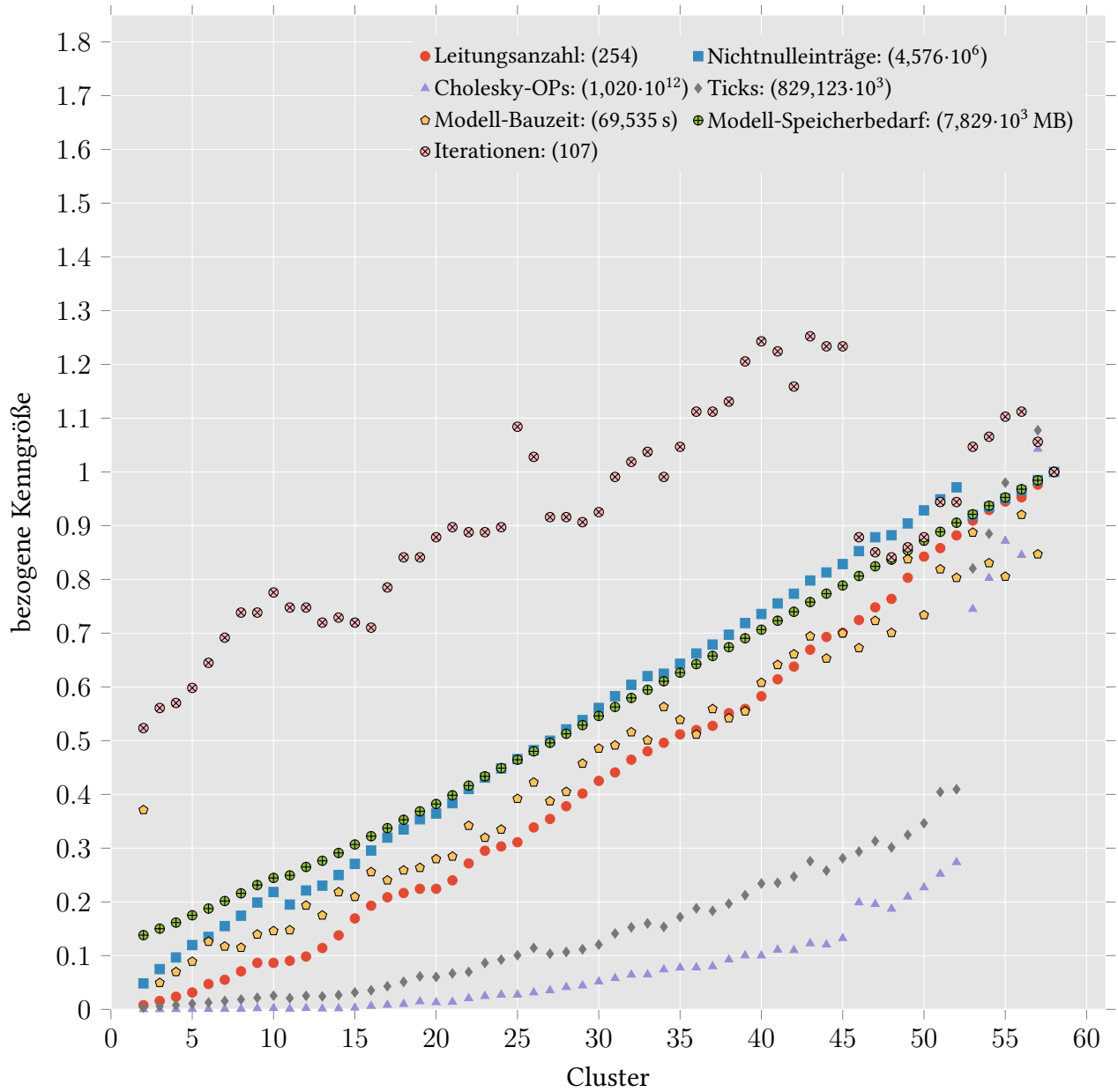


Abbildung 3.6: Verlauf weiterer Kennwerte für Komplexität und Leitungsausbau

einherzugehen. Damit ist ein Grund für den Verlauf der *CPLEX*-Zeit im Verhalten des Löser selbst zu sehen, der in manchen räumlichen Konfigurationen deutliche Unterschiede der benötigten Iterationsschritte zu den vorherigen bzw. folgenden Aggregation aufzeigt. Dies kann auch ein Hinweis auf eine große Parametersensitivität sein, die durch weniger strenge Toleranzgrenzwerte des Lösungsalgorithmus gelockert werden könnte.

Vergleich mit früheren Arbeiten

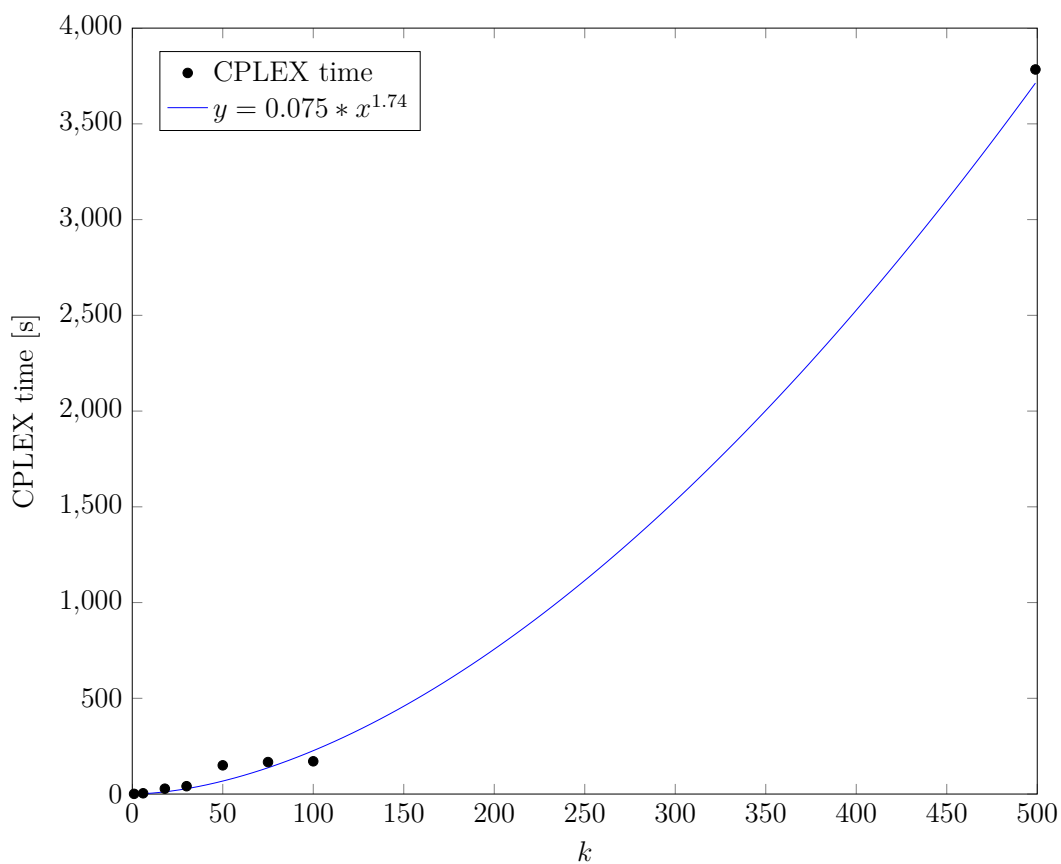


Figure 3.7.: CPLEX time [s] for different levels of aggregation for the investigated time period in logarithmic representation

Abbildung 3.7: Ausschnitt aus Arbeit [7]: *CPLEX*-Zeit über Anzahl an Clustern

Abb. 3.7 zeigt einen Ausschnitt aus der Arbeit von Johannes Metzdorf [7].

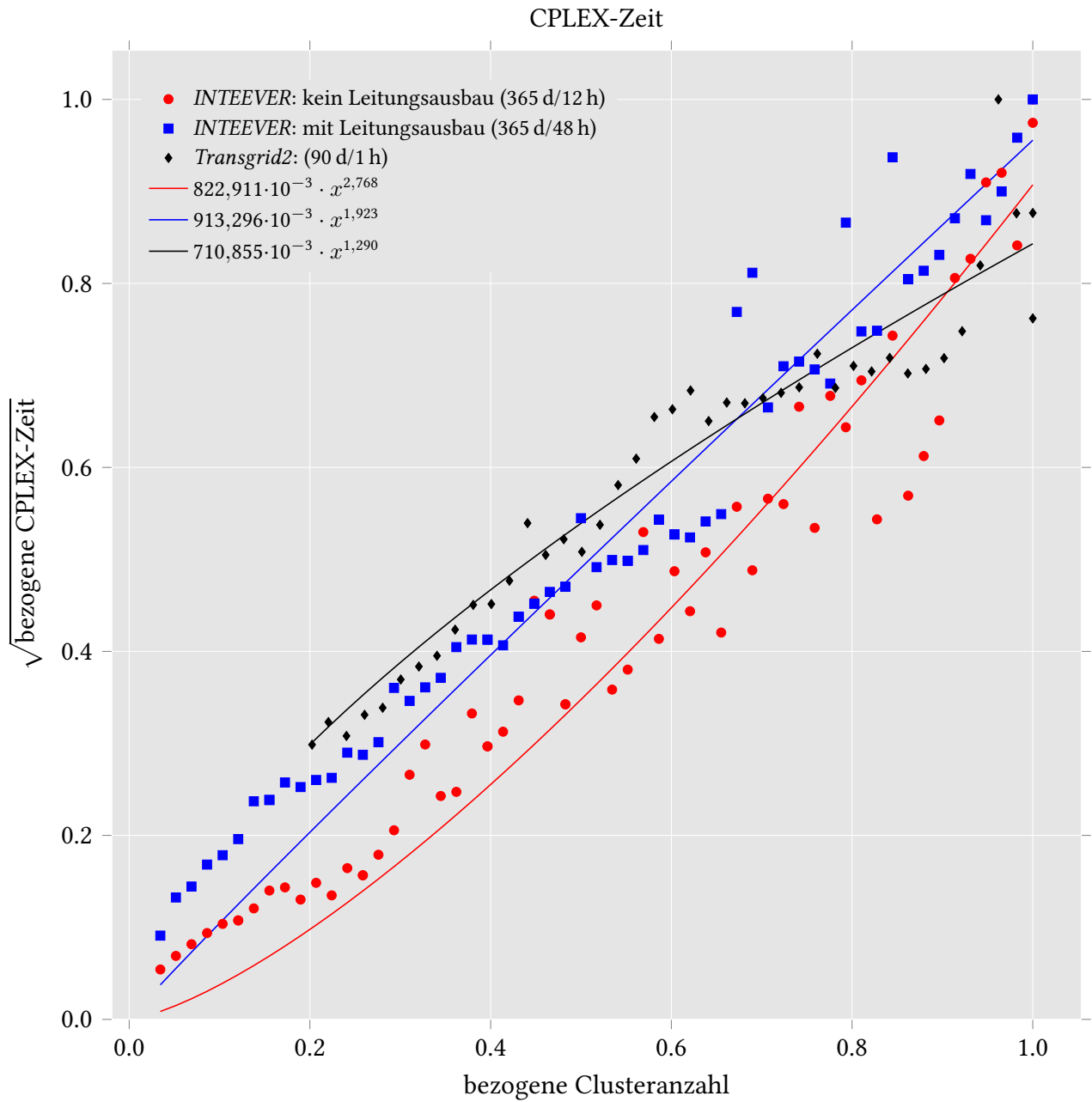


Abbildung 3.8: Vergleich der *CPLEX*-Zeiten im *INTEEVER*- und *Transgrid2*-Projekt mit jeweiligen Näherungspolynomen. Beide Achsen sind auf jeweilige Referenzfälle normiert.

Abb. 3.8 auf Seite 19 zeigt den Verlauf der *CPLEX*-Zeit im *INTEEVER*-Projekt sowie im *Transgrid2*-Projekt. Die Quadratwurzeldarstellung der y-Achse und die abgebildeten Näherungsterme sollen einen Vergleich mit einem vermuteten quadratischen Verlauf der *CPLEX*-Zeit erleichtern. Die beiden *INTEEVER*-Kurven zeigen einen Exponenten von knapp 2 bis 3 und könnten ein mindestens quadratisch ansteigendes Wachstum der *CPLEX*-Zeit bei räumlicher Aggregation nahelegen. Sie unterscheiden sich durch die Parameterwahl des Leitungsausbau, der erlaubt bzw. verboten war. Der Leitungsausbau besteht hier in der Erhöhung der Übertragungskapazität zwischen zwei Knoten, jedoch nicht in einer Topologieänderung. Dass ein annähernd quadratischer Verlauf aber nicht allgemein gültig ist, zeigen einerseits die großen Unterschiede der beiden *INTEEVER*-Kurven in Abb. 3.8 auf Seite 19 als auch das Verhalten im *Transgrid2*-Projekt, das fast linear verläuft. Zu beachten ist hier, dass die zahlreichen Ausreißer eine Herausforderung für den Fitting-Algorithmus darstellen und die hier dargestellten Terme nicht die beste Wahl eines Fittings und Näherungspolynom bilden sollen, sondern lediglich eine Tendenz des Verlaufs der *CPLEX*-Zeiten bei räumlicher Aggregation vermitteln sollen. Auch die Schwankungen, wie sie in Abschnitt 2.2 auf Seite 4 beschrieben wurden, fanden hier keine Beachtung. Unterschiede zwischen *Transgrid2* und *INTEEVER* bestehen unter anderem aus:

- Menge und Auswahl der Module: *INTEEVER* enthält mehr Module mit mehr Technologien wie Elektro-Mobilität, Speicher, etc
- leistungsfähigere Module: *INTEEVER* enthält aktuellere Software mit mehr Fähigkeiten, wie z. B. Leitungsausbau. Die Aggregation im *INTEEVER*-Projekt wurde allerdings mit einem Modul durchgeführt, das weniger Lastflusseinschränkungen implementiert als das im *Transgrid2*-Modul.
- geographische Verteilung der Knoten: *INTEEVER* besteht aus 58 Knoten, wobei viele Knoten dabei ganze Länder abbilden. Für Clusteringalgorithmen stellt diese Situation eine Herausforderung dar, besonders da auch die ans restliche kontinentaleuropäische Netz schwach angebundenen nordafrikanischen Länder (Algerien, Tunesien, Marokko) betrachtet werden.

Die Unterschiede zwischen Abb. 3.7 auf Seite 18 und dem Verhalten aus Abb. 3.8 auf Seite 19 können auch durch Wahl des Transportmoduls verursacht werden. Wie in Abschnitt 3.1.1 auf Seite 14 hervorgehoben, kann beim Referenzfall das nicht-aggregierende Transportmodul einen Sprung in Komplexität und damit *CPLEX*-Zeit verursachen, da es parallel verlaufende Leitungen nicht zusammenfasst und somit einen erhöhten Rechenaufwand besitzt. Dieses könnte beim Vergleich mit dem für das Clustering notwendige aggregierende Transportmodul ein nicht-lineares Verhalten nahelegen. Für eine abschließende Betrachtung wäre es notwendig, die Qualität des Clusterings selbst zu bewerten⁵ und die Unterschiede in den Modulen bezüglich Softwareversion und Parametrierung zu analysieren und zu bewerten.

⁵zumindest bei Ausreißern

Ergebnisdiskussion

INTEEVER und *Transgrid2* ist gemein, dass Kennwerte wie die Anzahl der Nichtnulleinträge, die Modell-Bauzeit und der Modell-Speicherbedarf linear verlaufen. Auch der Leitungsausbau scheint bei beiden Modellen, trotz den sehr unterschiedlichen geographischen Gegebenheiten, ähnlich linear zu verlaufen. Aus diesen Kennwerten ist also kein vollständiger Rückschluss auf die Qualität der Aggregation möglich, die bei einer Cluster-Anzahl nahe des Referenzpunktes zu ungewünschten Ergebnissen führt: im *Transgrid2*-Modell gibt es in diesem Bereich große Überschwinger der *Ticks*, der benötigten Iteration und auch die *CPLEX*-Zeit liegt 30 % über dem Referenzfall. Im *INTEEVER*-Projekt fällt der starke Anstieg der *CPLEX*-Zeit auf, der schon bei wenigen aggregierten Elementen zu einer Halbierung der Rechenzeit führt. Überschwinger nahe des Referenzpunkts sind in diesem Modell auf *Ticks* beschränkt, die *CPLEX*-Zeit ist bei jeder Aggregation unterhalb des Referenzpunkts.

Im *Transgrid2*-Modell weicht der Zielwert weniger von der Referenz ab, die Rechenzeiteinsparungen fallen allerdings auch geringer aus.

Wie schon erwähnt bedarf es also weiteren Untersuchungen zur Wahl des Clusteringalgorithmus, Einsatz von räumlicher Aggregation über einen weiten Bereich, Anwendung und Vergleich bei verschiedenen Modellen und Auswahl und Bewertung der gewählten räumlichen Aggregation.

Sprünge in der Kostenfunktion Große Unterschiede im *Transgrid2*-Projekt deuten auf einen großen Unterschied der Einsatzplanung konventioneller Kraftwerke hin, da die Systemkosten hauptsächlich durch Kosten der Kraftwerke bestimmt werden. Im Folgenden soll diese Vermutung ansatzweise gezeigt werden.

Abb. 3.9 auf der nächsten Seite zeigt die erzeugte Jahresenergiemenge verschiedener konventioneller Kraftwerkstypen. Es zeigt sich, dass der Sprung im Zielwert zwischen 290 und 300 Clustern mit einem Sprung der erzeugten Energiemenge verschiedener Kraftwerke einhergeht. Vor allem der Rückgang der Braunkohle und das Wachstum der Gaskraftwerke sind ein Indiz für den Bedarf an erhöhter Flexibilität, die bei einer räumlich feineren Auflösung gefordert ist. Grund hierfür sind die zunehmenden Netzbeschränkungen wie begrenzte Kapazität und entstehenden Verluste beim Transport über Leitungen. Damit können hohe Kostensprünge plausibel erscheinen. Äquivalent dazu zeigen sich beim Übergang von 280 zu 290 Clustern wie im Zielwert nur minimale Unterschiede. Der Übergang von 300 zu 310 Clustern verhält sich ebenso.

Im *INTEEVER*-Modell ist die Kostenstruktur komplexer, was auch am stärkeren Absinken des Zielwerts bei räumlicher Aggregation in Abb. 3.5 auf Seite 16 ersichtlich ist. Die Sprünge dabei sind noch größer und legen die Vermutung nahe, dass neben ähnlichen Effekten wie in Abb. 3.9 auf der nächsten Seite gezeigt auch andere Module vergleichbares Verhalten aufweisen, das heißt sprunghaft ihr Verhalten ändern. Nähere Betrachtungen der entstehenden Aggregation mit Fokus auf die Veränderungen des Leitungsausbaus sind angebracht.

Einen möglichen Umgang mit solchen Effekten lässt sich von Abb. 3.1 auf Seite 10 und Abb. 3.3 auf Seite 13 ableiten. Das qualitative Verhalten des Zielwerts und eingeschränkt der *CPLEX*-Zeit

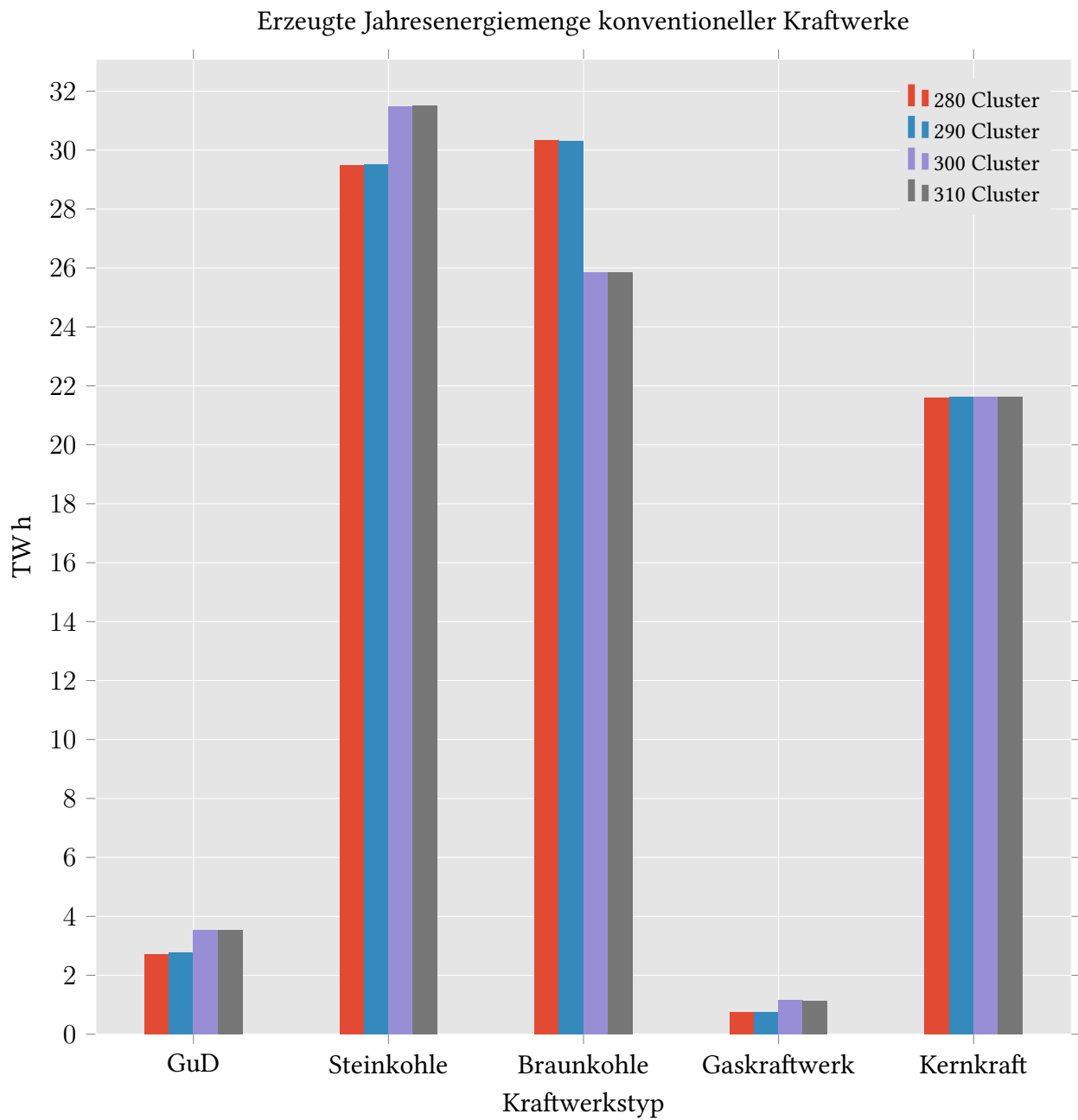


Abbildung 3.9: Aufsummierte Jahresenergiemenge aller Knoten nach Kraftwerkstyp bei 90 Tagen als Zeithorizont

bleibt bei Variation der zeitlichen Konfiguration bestehen, es können also Bereiche im Clustering identifiziert werden, die für den Anwendungsfall akzeptabel sind (zum Beispiel Reduktion des Zielwerts um 5 % bei Reduktion der *CPLEX*-Zeit um 70 %), sich in einem linearen Bereich befinden (um nicht in Gefahr zu laufen, bei geringen Änderungen großen Sprüngen zu unterliegen) und diese gefundenen Parameter dann auf die rechenzeit-intensive Parametrierung anwenden.

Bemerkungen zur räumlichen Aggregation Bei der Untersuchung der räumlichen Aggregationen ergeben sich folgende Fragen:

- kann ein für die Rechenzeit interessantes Clustering durchgeführt werden, wenn der Aggregationsgrad⁶ an die 100 % wächst?
 - die breite Anwendung von Clusteringverfahren scheint auf Situationen ausgelegt, in denen eine große Anzahl an Elemente auf wenige Cluster eingeteilt wird
- Kann ein “manuelles Clustering” nachgeahmt werden?
 - das übliche Verfahren, hochauflösende Daten in bekannten Konfigurationen nach geographischen und politischen Aspekten zusammenzufassen, könnte ein Ausgangspunkt für alternative Clustering-Strategien sein
 - * bei inkonsistenten Clustern könnten Cluster aus geographisch benachbarten Knoten gebildet werden, evtl. auch unter Berücksichtigung von Landesgrenzen oder Grenzen von Versorgungsgebieten der Energieversorger

3.1.2 Verringerung der Modulauswahl

In diesem Abschnitt soll die Auswirkung auf eine Verringerung der Modulauswahl auf die Rechenzeit untersucht werden. Unterschiede bestehen dabei in den Wärme, E-Mobilität- und Speichermodulen, Details dazu sind in Tabelle 4.1 auf Seite 45 und Tabelle 4.2 auf Seite 46 zu finden. Abb. 3.10 auf der nächsten Seite zeigt den Verlauf der *CPLEX*-Zeit bei Variation der räumlichen und zeitlichen Auflösung.

Bei zeitlicher Skalierung zeigt sich ein annähernd exponentieller Anstieg der Rechenzeit und eine ansteigende Spannweite zwischen den verschiedenen räumlichen Aggregationen: bei 12-stündlicher Auflösung liegt ungefähr der Faktor 100 zwischen kürzestem und längstem Lauf, in feinsten Auflösung beträgt dieser ungefähr 300. Bei maximaler Modulauswahl zeigt sich, dass 10 Cluster mit 1-stündlicher Auflösung, 29 Cluster mit 6-stündlicher Auflösung und 58 Cluster mit 12-stündlicher Auflösung ähnliche Rechenzeit benötigen ($1 \cdot 10^3$ s).

Abb. 3.11 auf Seite 25 zeigt die Erstellungszeit. Hier liegt das Szenario mit erweiterter Modulauswahl wie erwartet stets oberhalb des Szenarios mit verringerter Modulauswahl. Zwischen 6-stündlicher und 1-stündlicher Auflösung zeigt sich ein größerer Unterschied als in der *CPLEX*-Zeit, das heißt auch die räumlich größte Auflösung von 5 Clustern benötigt bei 1-stündlicher Auflösung deutlich

⁶siehe Fußnote 4 auf Seite 15

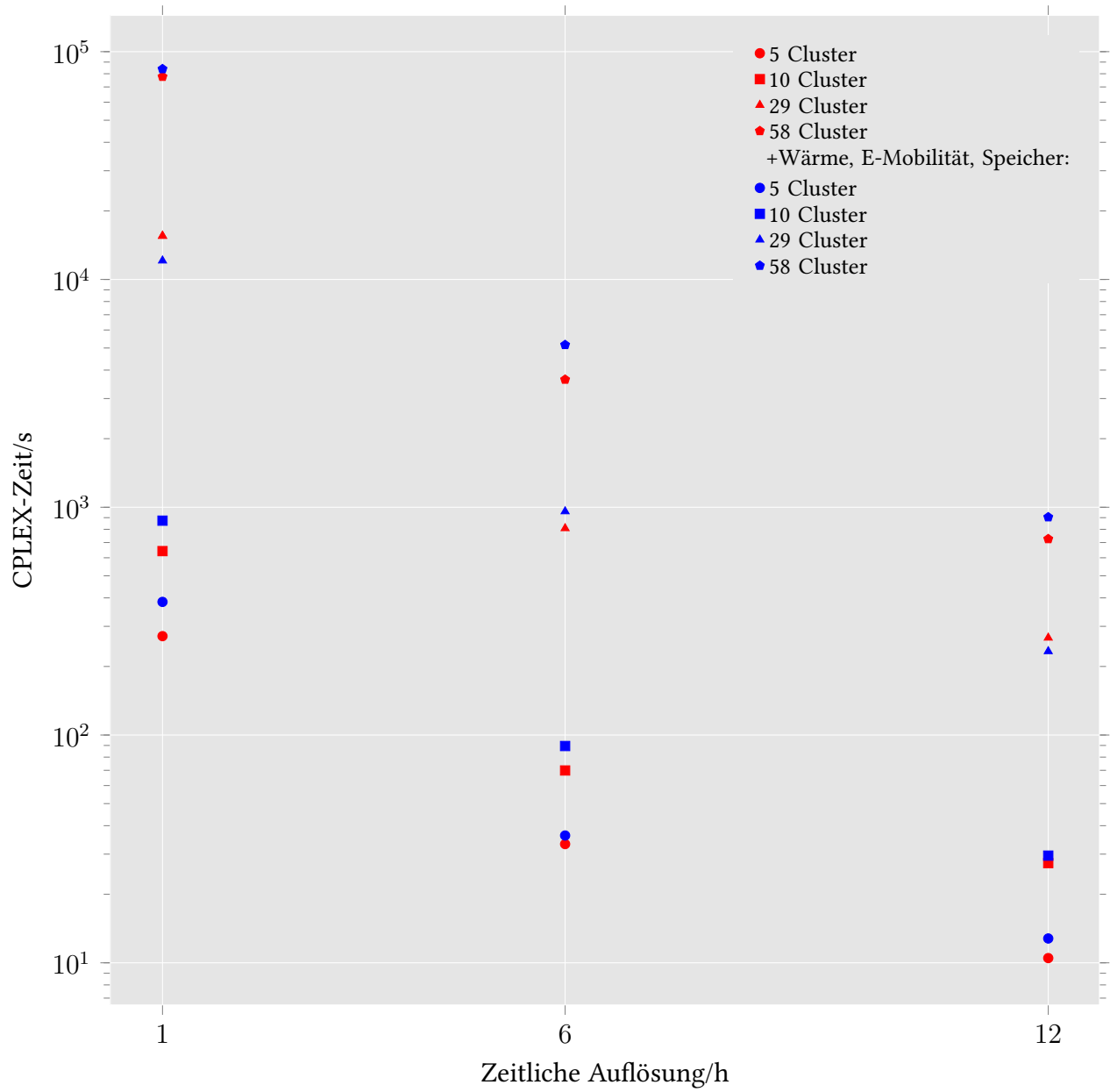


Abbildung 3.10: CPLEX-Zeit für minimale und maximale Modulauswahl bei Variation der räumlichen und zeitlichen Auflösung

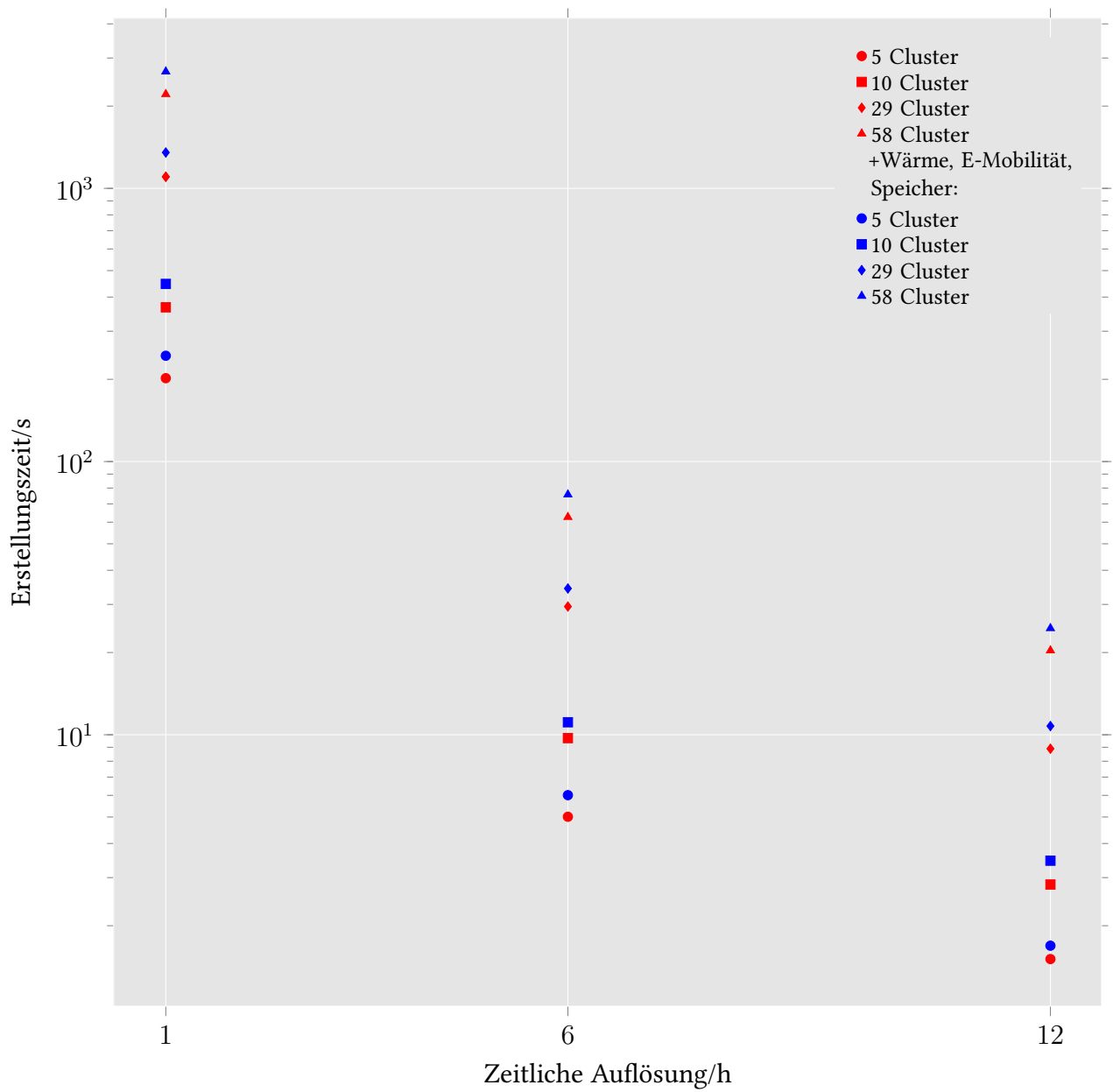


Abbildung 3.11: Erstellungszeit für minimale und maximale Modulauswahl bei Variation der räumlichen und zeitlichen Auflösung

mehr Zeit zum Erstellen als die räumlich feinste Auflösung bei 6-stündlicher Auflösung. Zwischen 6-stündlicher und 1-stündlicher Auflösung fallen die Unterschiede geringer aus. Die Spannweite zwischen den räumlichen Aggregationen scheint dabei über alle drei zeitlichen Auflösungen konstant zu bleiben.

Ergebnisdiskussion

Bei der gewählten Konfiguration und Parametrierung zeigen sich keine signifikanten Unterschiede der *CPLEX*-Zeit bei Variation der Modulauswahl. Dies kann an in folgenden Eigenschaften begründet sein:

- begrenzte Ausbauplanung im Szenario notwendig, da die installierten Leistungen meist gesetzt sind (*EHighwaySM*)
- einfaches E-Mobilität-Modul: hier war ein einfaches E-Mobilität-Modul aktiv, das nur auf nicht-rechenintensive Implementierungen von zeitlich verschobenem Laden in festen Kategorien setzt. Neben diesem ist auch ein Modul vorhanden, das aufwändige Prinzipien wie Rückspeisen implementiert. Es war hier wegen fehlenden Daten nicht aktiv.
- anscheinend geringer Einfluss der Speicher. Im Allgemeinen können Speicher großen Einfluss auf die Komplexität besitzen, da sie die Freiheitsgrade der Optimierung erhöhen, was hier allerdings nicht der Fall ist.

Bei einer gewünschten Reduktion der Rechenzeit muss also die Auswirkung von Modulen auf die Komplexität betrachtet werden, um bei einer Verringerung der Modulauswahl deutliche Rechenzeiteinsparungen zu erhalten.

3.2 Rolling Horizon

In diesem Abschnitt soll der Einfluss der Anwendung der *Rolling Horizon*-Methode auf das *IN-TEEVER*-Projekt untersucht werden. Eine Beschreibung der Implementierung findet sich in der Arbeit von Sebastian Schreck [9]. Zuerst soll kurz das Prinzip beschrieben werden: *Rolling Horizon* bedeutet das Aufteilen des betrachteten Zeithorizonts in mehrere Zeitabschnitte, die sequentiell optimiert werden. Damit ergibt sich eine Begrenzung des "Perfect Foresight", wie er sonst gegeben ist und eine Rechenzeitreduktion durch die kürzeren Zeitabschnitte. Zur Annäherung an die exakte Lösung, das heißt die Lösung ohne *Rolling Horizon*, sollten sich das Ende eines Abschnitts mit dem Start des nächsten Abschnitts überlappen, um z.B. Bedingungen bei Speichern (stetiger Verlauf der aktuellen Kapazität) gerecht zu werden. Auch sollen dabei Betriebsweisen, die in der Realität nicht praxistauglich sind, wie das vollständige Entleeren von Speichern zum Ende des jeweiligen Zeitfensters hin, verhindert werden.

Tabelle 3.2 auf der nächsten Seite zeigt die detaillierten GAMS-Zeiten bei 4 und 64 *Rolling Horizon*-Intervallen. Der iterative Ablauf von *GAMS* und *CPLEX*, wie in Abschnitt 2.1 auf Seite 3 skizziert, ist an den wachsenden Gesamtzeiten ersichtlich und schlägt sich besonders in Zuwächsen

Tabelle 3.2: Ausgabe der verschiedenen GAMS-Zeitdauern bei 4 und 64 Intervallen

	4 Intervalle		64 Intervalle	
	Einzel-schritt	Gesamtzeit	Einzel-schritt	Gesamtzeit
STARTUP/s	3,238	9,310	3,337	113,604
COMPILATION/s	0 (+0000)	$18 \cdot 10^{-3}$	0 (+0000)	$28 \cdot 10^{-3}$
EXECUTION/s	250,777	282,380	244,813	330,127
CLOSEDOWN/s	0 (+0000)	6,074	0 (+0000)	89,005
TOTAL SECONDS/s	254,015	297,782	248,150	532,764
ELAPSED SECONDS/s	258,860	$1,344 \cdot 10^3$	252,996	643,724
MAX HEAP SIZE/MB	$3,494 \cdot 10^3$	$3,494 \cdot 10^3$	$3,495 \cdot 10^3$	$3,495 \cdot 10^3$

der STARTUP- und CLOSEDOWN-Zeiten nieder, die damit Rechenzeiteinsparungen durch geringere CPLEX-Zeiten entgegenwirken. Der Teilschritt COMPILATION fällt vernachlässigbar gering aus und steigt auch bei 64 Intervallen nicht wesentlich an.

Dies liegt an der sequentiellen Arbeitsweise, wie sie in Kapitel 2 auf Seite 2 beschrieben wird: der Teilschritt der COMPILATION erfolgt vor dem Erstellen der Binärdateien und ist somit nicht mehr in der Ausgabe, die hier vorliegt, vorhanden.

Die maximal reservierte Speichermenge verändert sich nicht wesentlich zwischen 4 und 64 Intervallen, da als letzter Schritt von GAMS der komplette Zeithorizont nochmals geladen werden muss, um das gefundene Ergebnis in die genauen Verläufe aller Entscheidungsvariablen über den gesamten Zeithorizont umzusetzen. Da üblicherweise der Speicherbedarf des Lösers den der Modellierungssprache übersteigt ist dieses Verhalten vernachlässigbar. Nur in Ausnahmefällen, in denen auch der Löser für sparsamen Speicherumgang konfiguriert wurde, könnte eine mögliche Speicherreduktion durch *Rolling Horizon* interessant sein.

Abschnitt 3.2 auf der nächsten Seite zeigt die Gesamtzeit im H_2 -Szenario über die Anzahl der *Rolling Horizon*-Intervalle. Das H_2 -Szenario hat erstmals in dieser Arbeit, wie der Name nahelegt, Module aktiviert und parametrisiert, die Wasserstoffnachfrage und Wasserstoffherzeugung modellieren. Damit hat dieses Szenario die umfangreichste Modulauswahl der hier verwendeten Szenarien.

Das Minimum der Gesamtzeit liegt für beide Auflösungen bei 16 Intervallen, dabei weicht der Zielwert allerdings schon um 44 % bzw. 56 % vom Referenzfall ab. Bei einer zeitlichen Auflösung von 12 h zeigt sich, dass die Abweichung vom Referenzzielwert schneller ansteigt als bei 6 h und der Anstieg nach dem Minimum der Gesamtzeit ebenso steiler erfolgt.

Die Einsparung der Rechenzeit fällt schon bei wenigen Intervallen sehr deutlich aus: bei 4 Intervallen halbiert sich die Gesamtzeit im 6-stündlichen-Szenario während der Zielwert um 27 % ansteigt. Bei diesen Einsparungen kann eine Vergrößerung der Überlappung zwischen den *Rolling Horizon*-Intervallen sinnvoll sein, um näher am Referenzzielwert zu verbleiben. Hier betrug diese 10 %.

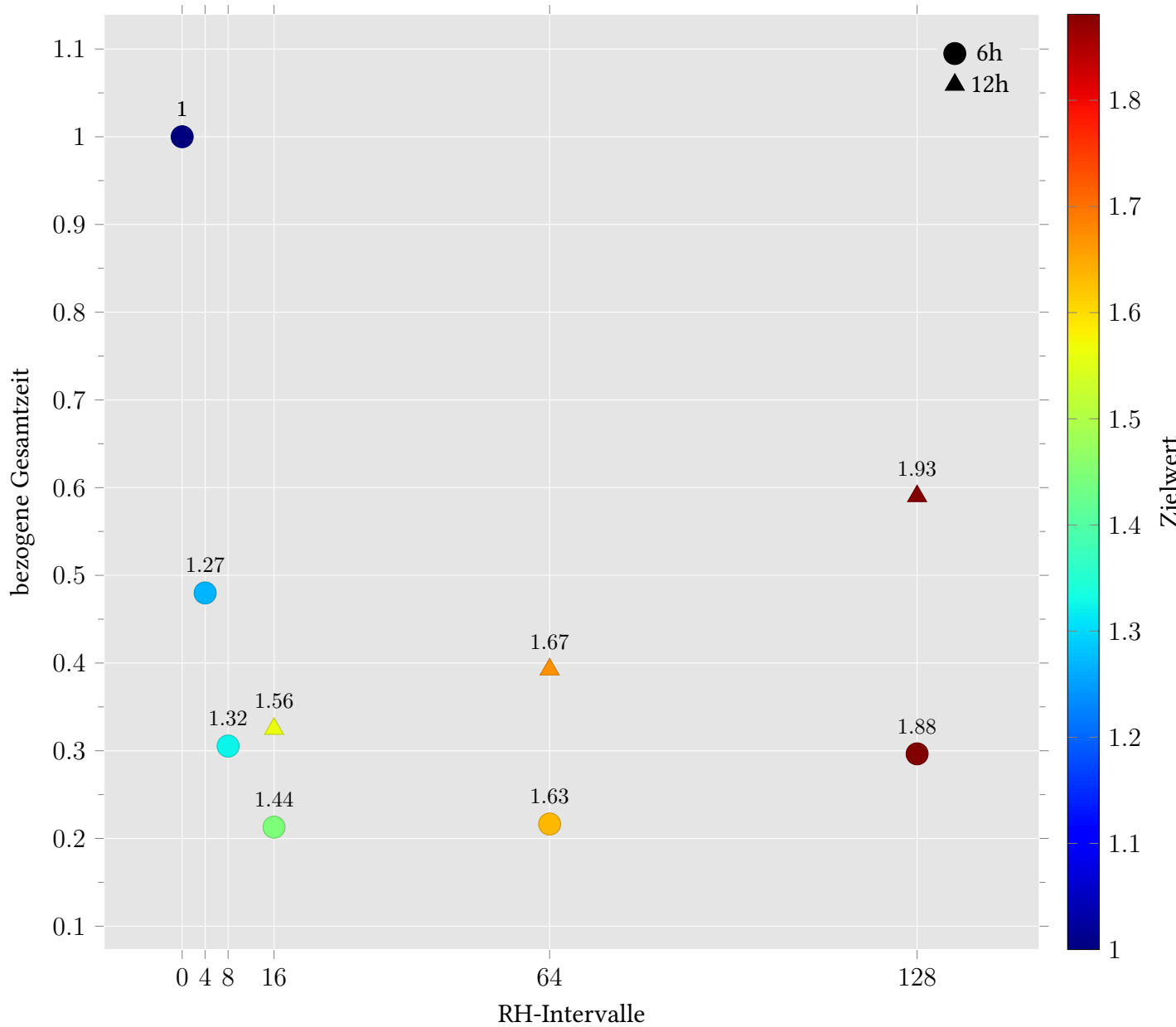


Abbildung 3.12: Verlauf der Gesamtzeit im H₂-Szenario bezogen auf den Referenzfall ohne RH bei Variation der RH-Intervalle für zeitliche Auflösungen von 6- und 12-Stunden. Zusätzlich wird der Zielwert gezeigt, der ebenso auf den Referenzfall normiert ist. Die Zeitdauer beträgt ein Jahr, die Überlappung zwischen den RH-Intervallen 10 %.

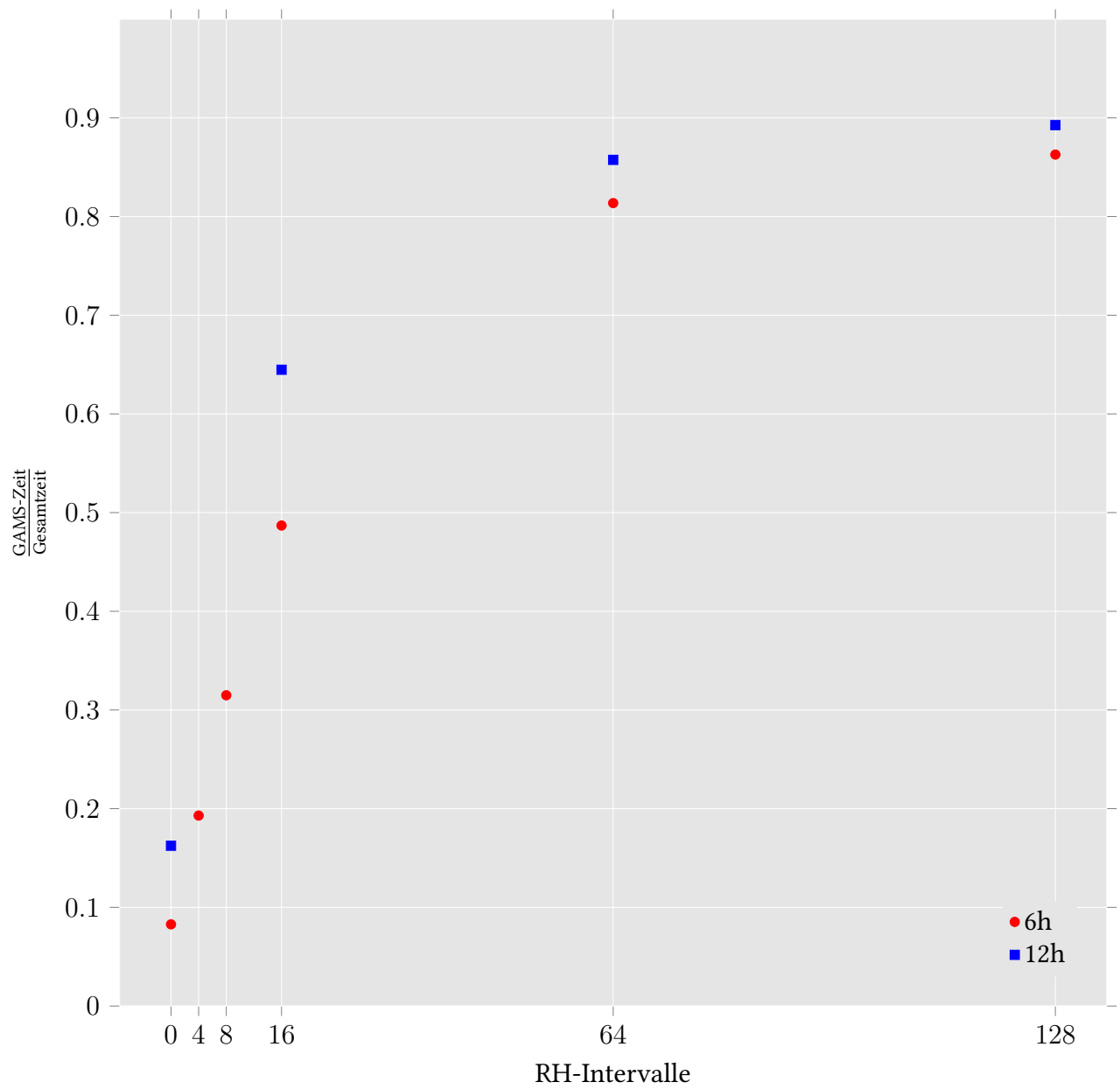


Abbildung 3.13: Anteil der GAMS-Zeit an der Gesamtzeit über der Anzahl der RH-Intervalle.

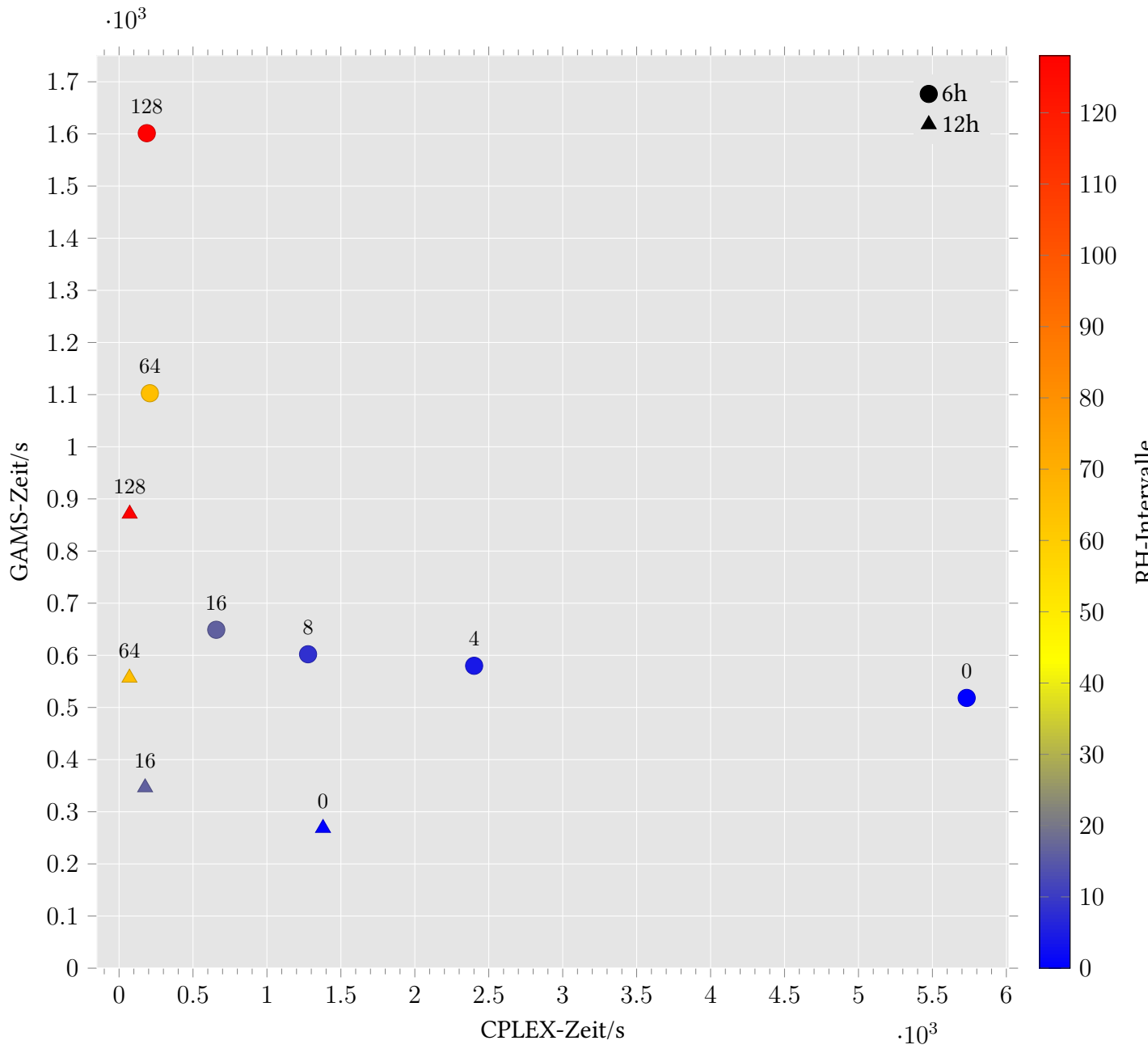


Abbildung 3.14: GAMS-Zeit über CPLEX-Zeit bei Variation der Rolling Horizon-Intervalle.

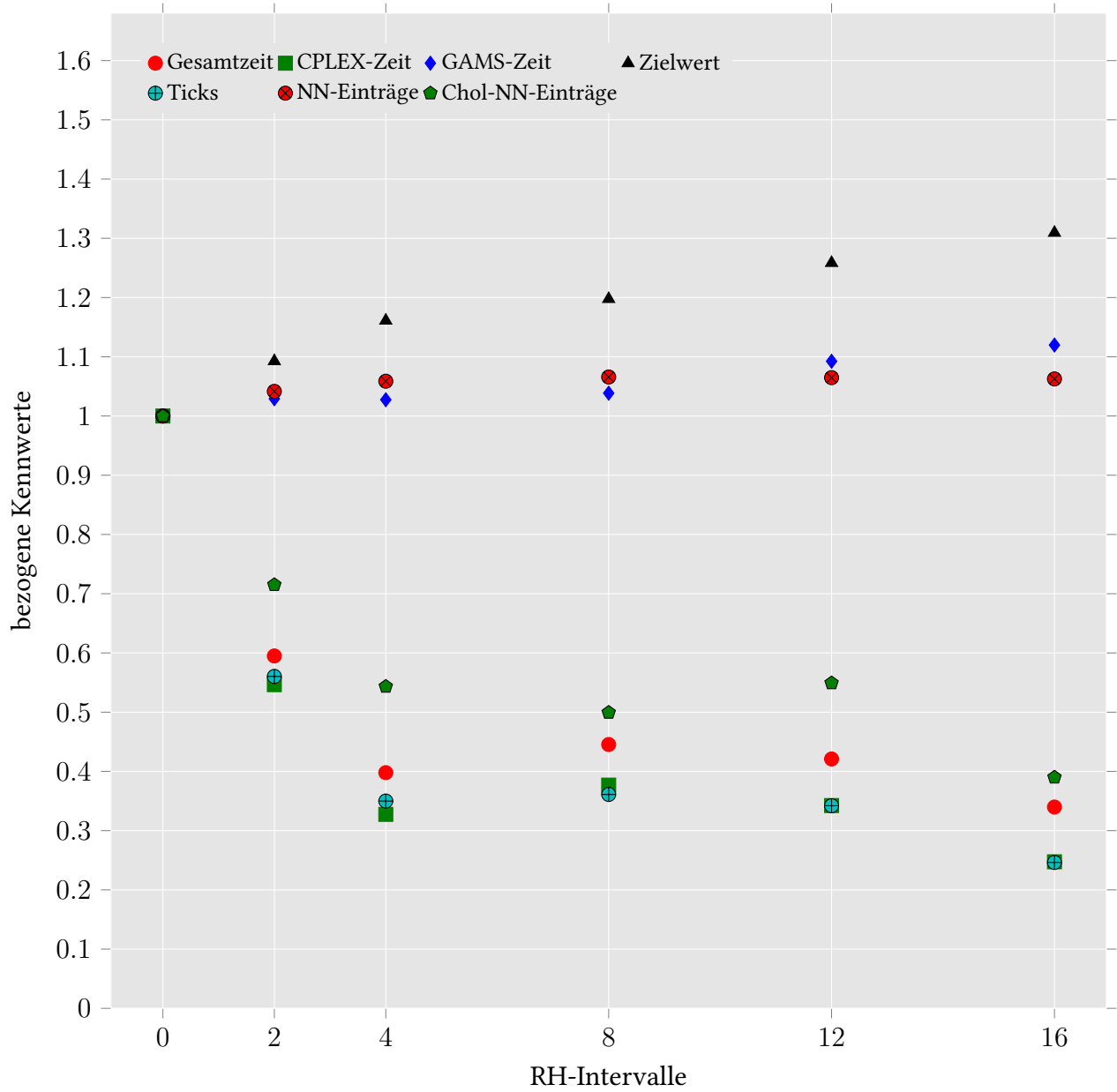


Abbildung 3.15: Auf Referenzfall (ohne *Rolling Horizon*) bezogene Kennwerte bei 365 d/2 h

Abb. 3.13 auf Seite 29 zeigt den Anteil der *GAMS*-Zeit an der Gesamtzeit. Im Referenzfall liegt dieser Anteil bei unter 10 % bzw. 20 % für 6 h und 12 h, bei 128 Intervallen steigt dieser auf fast 90 % an. Da es meist erwünscht ist, die *GAMS*-Zeit niedrig zu halten⁷, zeigt sich, dass die Wahl einer hohen Anzahl an *Rolling Horizon*-Intervallen ungünstig ist.

Abb. 3.14 auf Seite 30 zeigt erneut den Zusammenhang zwischen *GAMS*-Zeit, *CPLEX*-Zeit und den *Rolling Horizon*-Intervallen, hier in absoluten Werten. In dieser Darstellung ist eine Lage nahe des Ursprungs erwünschenswert. Hier ist besonders ersichtlich, wie der Übergang von 64 auf 128 Intervallen in beiden zeitlichen Konfigurationen nur noch längere *GAMS*-Zeiten bewirkt, aber kaum noch Einfluss auf die *CPLEX*-Zeit hat.

Abb. 3.15 auf Seite 31 zeigt den Verlauf diverser Kenngrößen im H_2 -Szenario für den betrachteten Zeitraum eines Jahres mit einer zeitlichen Auflösung von 2h. Alle Werte sind dabei auf den Referenzfall normiert. Hier zeigt sich ein lokales Minimum der Gesamtzeit schon bei 4 Intervallen, ein leichter Anstieg bei 8 und 12 Intervallen und erst ab 16 Intervallen das schon vermutete Minimum der Gesamtzeit. Das Verhältnis der Cholesky-Nichtnulleinträge, der Gesamtzeit, *Ticks* und *CPLEX*-Zeit verläuft dabei zueinander bei allen Intervallen ähnlich. Der insgesamt umfangreichere Rechenaufwand zeigt sich an der leicht ansteigenden Zahl der Nichtnulleinträge. Die *GAMS*-Zeit wächst ebenso an, ist jedoch bei 16 Intervallen nur ungefähr 10 % größer als im Referenzfall. Der Zielwert zeigt einen stetigen Anstieg über der Zahl der *Rolling Horizon*-Intervalle.

Ergebnisdiskussion

Rolling Horizon zeigt sich als vielversprechende Methode, um große Rechenzeiteinsparungen von 50 % und mehr zu ermöglichen. Schon bei wenigen Rolling-Horizon-Intervallen zeigt sich fast eine Halbierung der *CPLEX*-Zeit. Das Optimum der *Rolling Horizon*-Intervalle scheint in den betrachteten Konfigurationen zwischen 4 und 16 zu liegen, wobei die Gründe für einen leichten Anstieg der Gesamtzeit zwischen 4 und 16 Intervallen wie in Abb. 3.15 auf Seite 31 betrachtet genauer untersucht werden sollten. Die Anzahl der Intervalle sollte gleichzeitig mit der zeitlichen Konfiguration verändert werden. Bei steigender Anzahl an *Rolling Horizon*-Intervallen zeigt sich ein geringer Rückgang der *CPLEX*-Zeit, bis schließlich ab 64 *Rolling Horizon*-Intervallen die steigende *GAMS*-Zeit und die wieder leicht ansteigende *CPLEX*-Zeiten für wachsende Gesamtzeiten sorgen. Problematisch zeigt sich der Verlauf des Zielwerts, der schon ab 2 *Rolling Horizon*-Intervallen um 10 % ansteigt (siehe Abb. 3.15 auf Seite 31), was deutlich über den erzielten von Abweichungen des Zielwerts im Testmodell (unter 1 %) von Sebastian Schreck (siehe [9]) liegt. Das kann in der erweiterten Modul- und Technologieauswahl und schlecht gewählter Parametrierung begründet sein. Eine schlecht gewählte Parametrierung für *Rolling Horizon* ist eine, die Ausbauoptimierung betreiben muss und so evtl. in einzelnen Abschnitten große Kapazitäten zubaut, die im weiteren Verlauf große Kosten verursachen. Als Gegenmaßnahme kann bei den beobachteten Rechenzeitgewinnen die Überlappung von den hier gewählten 10 % erhöht werden. Zusätzlich kann eine Heuristik für *Rolling Horizon* eingesetzt werden, bei der in einem ersten Schritt aus einer zeitlich groben Optimierung Stützpunkte für zum Beispiel Speicher gewonnen

⁷Hauptsächlich, weil *GAMS* nicht parallelisierbar ist und somit nur begrenzt beeinflusst werden kann.

werden, die dann im eigentliche Durchlauf mit Rolling Horizon wichtige Informationen bieten, um sich der exakten Lösung, das heißt ohne Aufteilung des Zeithorizonts, anzunähern.

3.3 Set-Order

In diesem Abschnitt soll die Auswirkung von so genannten *Set-Order*-Änderungen auf die gesamte *GAMS*-Zeit und die *CPLEX*-Zeit untersucht werden. Optimierungsproblem können grundsätzlich in zwei Formulierungen beschrieben werden: in Matrizenform und in Form einzelner Gleichungen, die die Koeffizientenmatrix komponentenweise beschreiben. Variablen und Koeffizienten gleicher Art werden in Modellierungssprachen für Optimierungsprobleme gerne als Mengen beschrieben.

Mengen in *GAMS* bestehen so z. B. aus den Zeitschritten, den ausgewählten Knoten oder einer Auswahl an Technologien. Mit diesen Mengen muss *GAMS* dann die Matrix für den Löser präparieren. Begründet durch Aussagen von *GAMS*-Entwicklern kann vermutet werden, dass die Reihenfolge der übergebenen Mengen an *GAMS* Auswirkungen auf *GAMS*-Zeit und auch das Verhalten des Löser haben kann. Die *GAMS*-Zeit wird beeinflusst, indem die stark unterschiedlichen Größen der Mengen (8760 Elemente für die Zeitschritte, einige wenige für Technologiegruppen) den benötigten Speicherbedarf und die benötigte Zeit verändern. *GAMS* behandelt grundsätzlich die übergebenen Mengen in der Reihenfolge, in der sie im Quelltext beschrieben werden. Obwohl die mathematische Formulierung dabei unverändert ist, könnte auch ein Einfluss auf die *CPLEX*-Zeit entstehen, da die Vorbehandlung in *CPLEX* (*Presolve*) großen Einfluss auf das tatsächlich zu berechnende Problem hat. Vorbehandlungen in *CPLEX* und anderen Lösern versuchen das eingegebene Problem zu reduzieren, indem die Dimensionen der Matrix und insbesondere die Anzahl der Nichtnulleinträge reduziert werden. Wichtig für die Komplexität des Problems ist dabei auch die Struktur der Matrix, das heißt ob es dicht besetzte Spalten oder Abschnitte gibt. Aus Speichergründen wird versucht, ein Großteil der Matrix als dünnbesetzte Matrix zu verarbeiten, da diese besonders platzsparend und damit schnell verarbeitet werden können. Teile der Matrix, die davon abweichen, werden von *CPLEX* separat behandelt. Unter anderem diese Vorbehandlungen geschehen in der Standardeinstellung automatisch. Für die Wahl, welche Vorbehandlungen in welcher Kombination gewählt werden sollen und wieviel Zeit für diese verwendet werden sollte, verwendet *CPLEX* diverse Heuristiken, die durch die Form der Eingabe an *CPLEX* veränderlich sein könnte. Während mathematisch also keine Änderung stattfindet, kann durch die beschriebene *CPLEX*-Arbeitsweise ein Unterschied in der Rechenzeit auftreten.

Hier werden nun die Unterschiede zwischen Revision r20778 und r20911 des *INTEEVER*-Projekts in den Szenarien *DG0-FC0* sowie *EHighwaySM* untersucht. Der hauptsächliche Unterschied zwischen beiden Szenarien besteht in den installierten Leistungen der Erzeuger, die im *EHighwaySM*-Szenario meist gesetzt sind und im *DG0-FC0*-Szenario unbestimmt, das heißt auf Null gesetzt sind. Die Ausbauoptimierung erhöht die Komplexität des Optimierungsproblems und sorgt so für einen Unterschied beider Szenarien in der *CPLEX*-Zeit von 60 %. Zwischen diesen Revision wurde die Formulierung in *GAMS* verändert und die Menge *timeModel* an die erste Stelle der *GAMS*-Anweisungen gesetzt.

Die folgenden Teilschritte von *GAMS* wurden aufgrund der geringen Zeitdauern in den folgenden Abbildungen nicht weiter betrachtet:

- compilation
- closedown
- startup

Der Teilschritt *execution time* wird nicht dargestellt, da er stets um ungefähr 0,5 % kleiner als der Wert der gesamten *GAMS*-Zeit ist.

Abb. 3.16 auf der nächsten Seite zeigt die Veränderungen der *EHighwaySM*- und *DG0-FC0*-Szenarien beim Übergang von Revision r20778 auf Revision r20911. Dabei wird von jedem Szenario die zeitliche Auflösung von 12- und 6 Stunden gezeigt. Die wichtigste Kennzahl ist die Gesamtzeit, die hier im *EHighwaySM*-Szenario jeweils sinkt und im *DG0-FC0*-Szenario jeweils steigt. Beim Übergang von 12 auf 6h ergibt sich bei *EHighwaySM* eine deutlich höhere Reduktion der Gesamtzeit als bei 12h, die Zunahme im *DG0-FC0*-Szenario schwächt sich auf nur noch ca. 2 % ab. Die Entwicklung der *GAMS*-Zeit zeigt ein Verhalten, das gegen die Einführung der in r20911 gewählten *Set-Order* spricht: in allen Fällen gibt es ein Ansteigen der *GAMS*-Zeit, das sich bei kleineren Zeitschritten noch erhöht. Ein Teil der *GAMS*-Zeit, die Erstellungszeit, zeigt dazu ein gegenteiliges Verhalten. Diese Reduktion geht mit einer Erhöhung des Speicherbedarfs um 50 % bis 60 % einher, wie es die mittlere Grafik in Abb. 3.16 auf der nächsten Seite zeigt. Der maximale Speicherbedarf⁸ scheint direkt mit der Erstellungsgröße zu korrelieren. Eine genauere Aufschlüsselung der Änderungen verschiedener Kennwerte findet sich in der unteren Teilgrafik. Es zeigt sich, dass die *Set-Order*-Änderungen kaum Einfluss auf die Anzahl der Nichtnulleinträge hat: im *DG0-FC0*-Szenario zeigt sich die maximale Änderung mit 3 %. Größere Unterschiede zeigen sich bei *CPLEX*-Zeit und *Ticks*: im *EHighwaySM*-Szenario zeigt sich eine Reduktion der benötigten *Ticks* um bis zu 74 %. Im *DG0-FC0*-Szenario dreht sich zwischen 12 und 6h die Zunahme von *CPLEX*-Zeit und *Ticks* in eine Reduktion beider Kennwerte um. Wie in der obersten Teilgrafik zu sehen, steigt dennoch in diesem Szenario die Gesamtzeit an. Die Anzahl der Cholesky-Nichtnulleinträge, ein Maß für Komplexität und Speicherbedarf in jedem Iterationsschritt, zeigt im *EHighwaySM*-Szenario ein ähnliches Verhalten wie *Ticks* und *CPLEX*-Zeit. Im *DG0-FC0*-Szenario hingegen ist beidesmal ein Ansteigen zu beobachten und außerdem ein weiteres Ansteigen von 12h auf 6h, was gegenläufig zum Verhalten von *CPLEX*-Zeit und *Ticks* ist. Im *DG0-FC0*-Szenario scheint also bei 6h eine niedrige Zahl an Iterationen nötig zu sein⁹. Die Cholesky-OPs verhalten sich bei beiden Szenarien ähnlich wie die Cholesky-Nichtnulleinträge. Insgesamt scheint also das *EHighwaySM*-Szenario positiver auf die *Set-Order*-Änderungen anzusprechen, weist aber auch Kennwerte auf, die sich deutlich verschlechtern. Insbesondere das Ansteigen der *GAMS*-Zeit ist negativ hervorzuheben, da diese Zeit nur durch wenige Eingriffsmöglichkeiten reduziert werden kann. Da insgesamt eine Zeitreduktion stattfindet und das Verhalten auch unter der neuesten *GAMS*-Version untersucht werden soll, wird in Abschnitt 3.3 auf Seite 36 das *EHighwaySM*-Szenario mit einer größeren zeitlichen Variation untersucht.

⁸Bezeichnung in *CPLEX*: `max heap size`

⁹Die Anzahl der Iterationen wird Abschnitt 3.3 auf Seite 36 betrachtet.

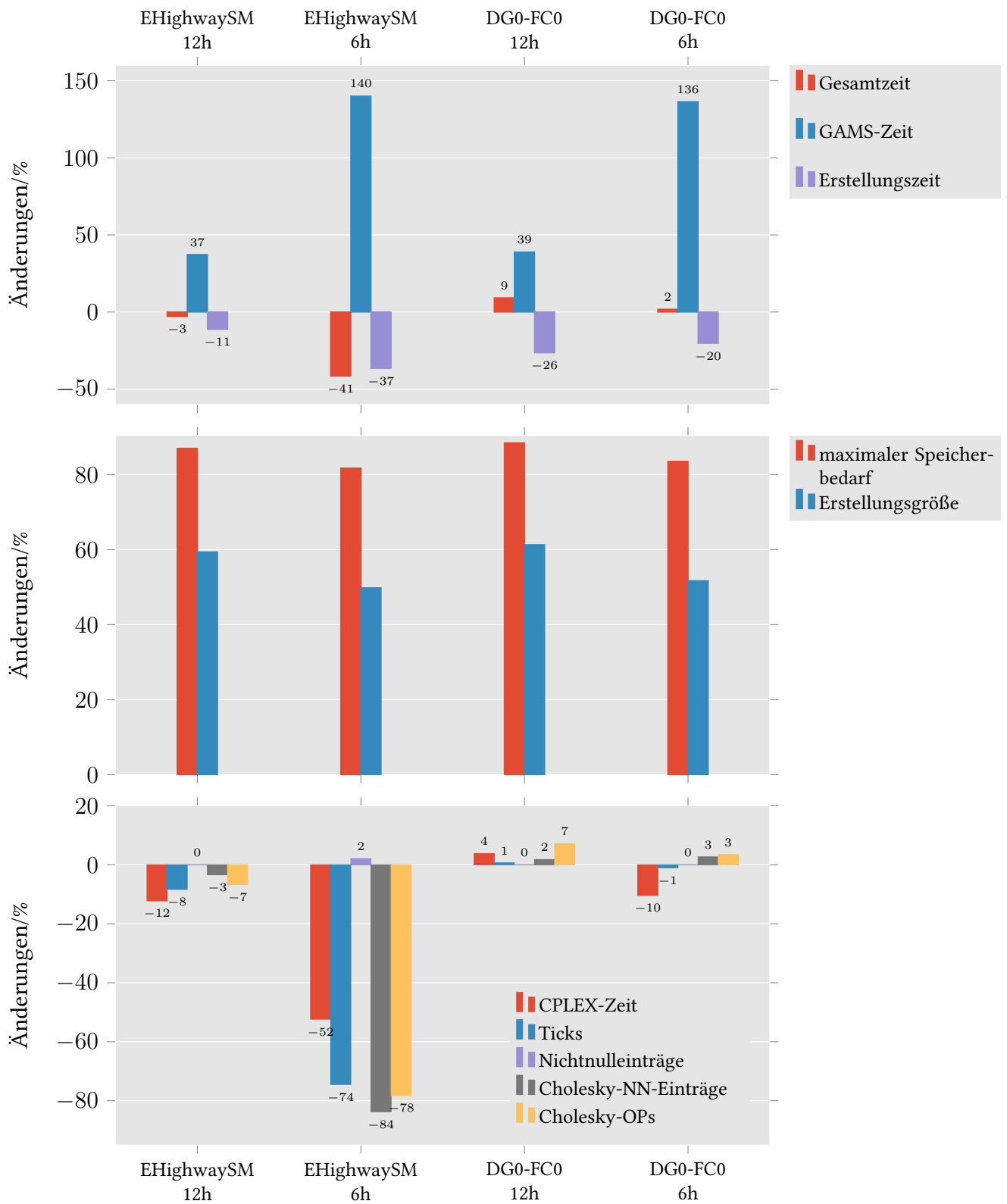


Abbildung 3.16: Veränderungen durch neue *Set-Order* bei den Szenarien *EHighwaySM* und *DG0-FC0* mit 12- und 6-stündlicher Auflösung (365 d) bezogen auf Revision r20778. Dabei wurden die g00-Dateien mit *GAMS* 24.3 erstellt und unter *GAMS* 24.7 ausgeführt.

GAMS 24.8

Da GAMS-Entwickler die Nutzung der neuesten Version empfehlen und bereits Unterschiede zwischen früheren Version festgestellt wurde, wurden die *Set-Order*-Änderungen in diesem Abschnitt mit der neuen GAMS-Version getestet.

Abb. 3.17 auf der nächsten Seite zeigt wie Abbildung Abb. 3.16 auf Seite 35 die Änderungen durch die neue *Set-Order*, hier zusätzlich mit 2-stündlicher Auflösung. Die Auffälligkeiten, wie die starke Reduktion mehrerer Kennwerte bei 6-stündlicher Auflösung in Abb. 3.16 auf Seite 35, lassen sich hier nicht wiederfinden. Das *DG0-FC0*-Szenario ist insgesamt über alle drei Auflösungen langsamer, in der 2-stündlichen Auflösung sinkt die GAMS-Zeit dann erstmals. Im *EHighwaySM*-Szenario ist die Gesamtzeit bei zeitlicher Auflösung von 2h fast identisch¹⁰, die GAMS-Zeit stark reduziert. Bei den Veränderungen des Speicherbedarfs in der mittleren Teilgrafik zeigt sich, dass der erhöhte Speicherbedarf bei mehr Zeitschritten relativ geringer ausfällt und von ca. 60 % auf 40 % fällt. In der aufgeschlüsselten Betrachtung der Komplexitätskenngrößen der unteren Teilgrafik zeigt sich im *DG0-FC0*-Szenario eine Erhöhung der Komplexität bei einer feineren Auflösung. Im *EHighwaySM*-Szenario zeigt überraschend das 12-stündliche Szenario die größten Reduktionen, das 6-stündliche Szenario eine vergleichbare Erhöhung und das 2-stündliche Szenario ein ausgewogenes Verhalten mit geringen Erhöhungen für *CPLEX*-Zeit und *Ticks* und sehr geringen Reduktionen im Aufwand je Iterationsschritt (Cholesky-Nichtnulleinträge und Cholesky-OPs).

Gemittelte Veränderungen durch neue Set-Order Die Betrachtung der Veränderungen durch die neue *Set-Order* sollen in diesem Abschnitt abschließend durch mehrere Zeitschritte und mehrere Start- und Endpunkte untersucht werden (siehe Tabelle 3.3). Damit soll der Parametereinfluss reduziert werden und die Auswirkungen eventueller Ausreißer minimiert werden. Identische Szenarien wurden nach Möglichkeit auch mehrmals ausgeführt, was die Verlässlichkeit der Zeitangaben erhöhen sollte.

Tabelle 3.3: Übersicht über Wahl von zeitlicher Auflösung, erstem und letztem Tag des Optimierungsproblem. Die Kombination von diesen Parametern ergibt (bei Auslassung von Fällen, in denen der Startzeitpunkt vor dem Endzeitpunkt liegt) insgesamt 18 Kombinationen.

Zeitliche Auflösung/h	Erster Tag/d	Letzter Tag/d
2	1	7
6	30	60
12	180	365

Abb. 3.18 auf Seite 38 zeigt die Veränderungen gemittelt über alle ausgewählten zeitlichen Auflösungen und speziell nur über eine zeitliche Auflösung von 2 Stunden. Über alle Zeitschritte wachsen bei beiden Szenarien Gesamt- und GAMS-Zeit. Auf 2-stündliche Auflösung reduziert

¹⁰Die Zahlenwerte in der Grafik sind ohne Nachkommastellen angegeben.

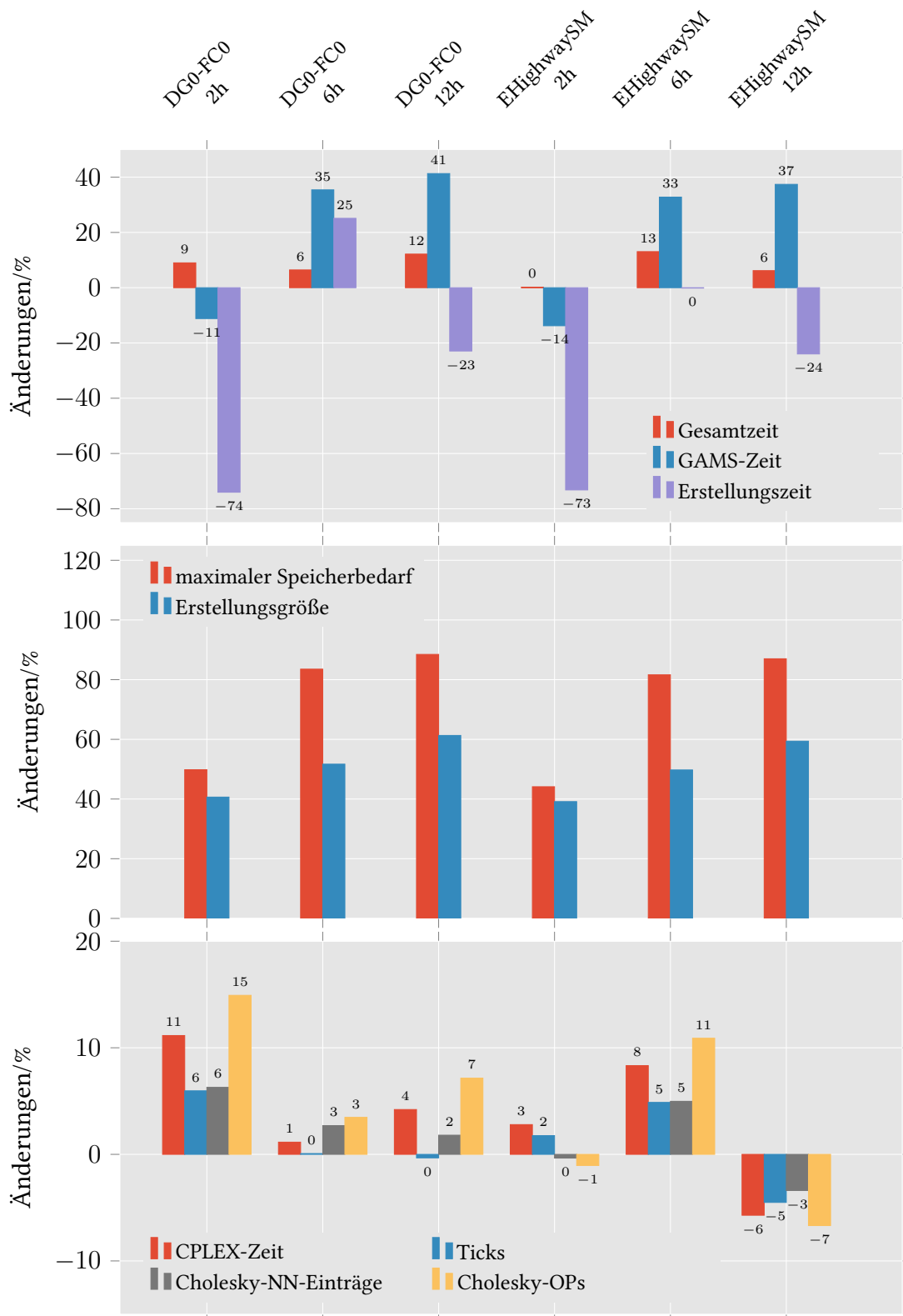


Abbildung 3.17: Veränderungen durch Revision r20911 bei den Szenarien *EHighwaySM* und *DG0-FC0* mit 12-, 6- und 2-stündlicher Auflösung (365 d). Die Nichtnulleinträge weisen keine Unterschiede auf und werden in der unteren Teilgrafik nicht dargestellt. (GAMS-Version 23.8)

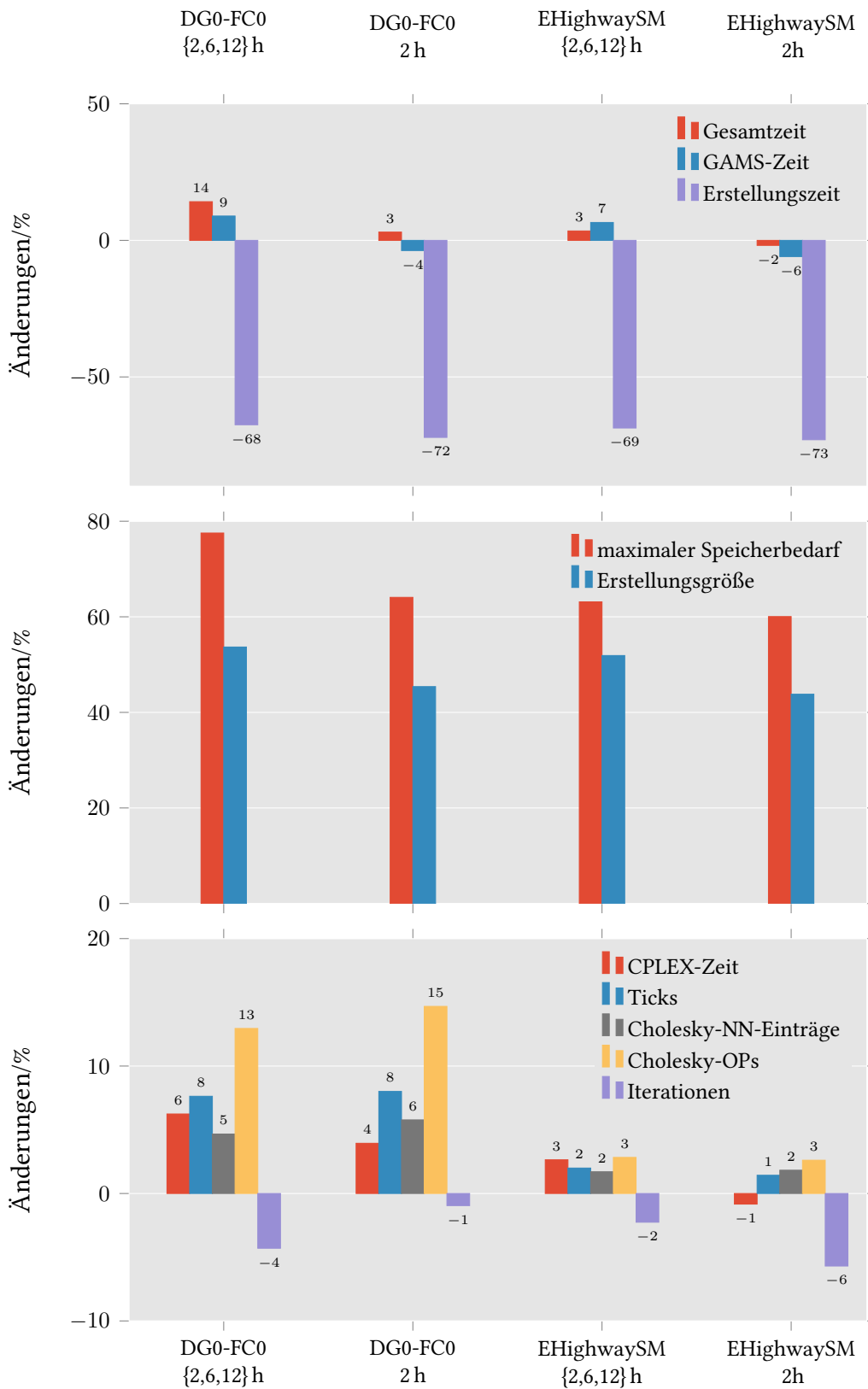


Abbildung 3.18: Veränderungen durch neue *Set-Order* bei den Szenarien *EHighwaySM* und *DG0-FC0* gemittelt über verschiedene zeitliche Konfigurationen bezogen auf Revision r20778. (GAMS-Version 24.8)

zeigt sich eine Reduktion der *GAMS*-Zeit bei beiden Szenarien, ein geringer Anstieg der Gesamtzeit im *DG0-FC0*-Szenario und eine geringe Reduktion im *EHighwaySM*-Szenario. Wie schon gesehen sinkt die Erstellungszeit stets deutlich, ebenso steigt der Speicherbedarf, allerdings relativ geringer mit einer zeitlich höheren Auflösung. In der unteren Teilgrafik zeigt sich, dass die *CPLEX*-Kennwerte für Komplexität ansteigen, die Anzahl der benötigten Iterationen dafür sinkt. Im *DG0-FC0* Szenario steigt die Zahl der Cholesky-OPs stärker an und die Iterationsanzahl sinkt auch im 2-stündlichen Szenario nur gering, was in Kombination die Anstiege aus der oberen Teilgrafik erklärt. Im *EHighwaySM*-Szenario steigen die Komplexitätskenngrößen in geringem Maße an und die Iterationsanzahl sinkt bei 2-stündlichen Szenarien deutlicher.

Analyse der *GAMS*-Zeit

Tabelle 3.4: Übersicht über die 10 längsten Teilschritte der *GAMS*-Zeit im *EHighwaySM*-Szenario (2h/r20911)

Dauer/s	Speicher/GB	Typ	Bezeichnung (Anzahl der Elemente)
515,555	6,025	Assignment	TDCAgg_importNodeToNode (1095000)
507,853	5,967	Assignment	TDCAgg_exportNodeToNode (1095000)
100,478	5,601	Assignment	FuelsAACost_fuelConsumption_Save (2041080)
35,235	6,050	Assignment	TDCAgg_transNodeToNode (500)
31,115	8,276	Solve Fini	OptiMoSystem (72827800)
29,946	5,231	Solve Read	OptiMoSystem
24,947	4,091	Equation	EQ_ECarsSmpl_Balance (709560)
23,491	7,649	Assignment	StorStd_DischargePower_Out (2058600)
22,378	7,779	Assignment	StorStd_StorageLevel_Out (2058600)
17,719	7,518	Assignment	StorStd_ChargePower_Out (2058436)

Tabelle 3.4, Tabelle 3.5 auf der nächsten Seite, Tabelle 3.6 auf der nächsten Seite und Tabelle 3.7 auf Seite 41 zeigen die *GAMS*-Ausgabe der 10 längsten Teilschritt, ihres Speicherbedarfs, Typ und Bezeichnung.

Die beiden langsamsten Teilschritte bleiben in allen Fälle gleich, mit der neuen *Set-Order* (Tabelle 3.4 und Tabelle 3.6 auf der nächsten Seite) steigt der Speicherbedarf allerdings stark an und die benötigte Zeit steigt ebenfalls leicht. Bei der zeitlichen Auflösung von 2 h (Tabelle 3.4 und Tabelle 3.5 auf der nächsten Seite) zeigt sich ab der dritten Zeile ein stärkerer Abfall der weiteren Dauern bei Revision r20911. Dieses Verhalten ist bei zeitlicher Auflösung von 12 h nicht zu beobachten. Außerdem befinden sich weniger Gleichungen in den 10 langsamsten Teilschritten.

Die *Set-Order*-Änderungen scheinen also auf wenige Teilschritte, insbesondere Zuweisungen mit vielen Elementen, negative Auswirkungen, auf die restlichen Schritte jedoch beschleunigenden Einfluss zu besitzen. Bei zeitlich grober Auflösung und kleineren Mengen zeigt sich kein positiver Einfluss.

Tabelle 3.5: Übersicht über die 10 längsten Teilschritte der GAMS-Zeit im *EHighwaySM*-Szenario (2h/r20778)

Dauer/s	Speicher/GB	Typ	Bezeichnung (Anzahl der Elemente)
510,932	3,791	Assignment	TDCAgg_importNodeToNode (1095000)
490,715	3,765	Assignment	TDCAgg_exportNodeToNode (1095000)
121,453	6,307	Equation	Eq_LoadCoverage (254040)
85,761	5,736	Equation	EQ_fCap_TotalPowerGeneration (254040)
84,951	5,511	Equation	EQ_dgs_TotalPowerGeneration (254040)
80,458	6,442	Equation	Eq_TotalPowerGeneration (254040)
50,810	5,824	Equation	EQ_fCap_TotalDemand (254040)
50,220	5,602	Equation	EQ_dgs_TotalDemand (254040)
37,353	2,995	Equation	Eq_ConvBase-TaWCostPC (286)
35,648	1,506	Equation	EQ_tDCAgg_losses (254040)

Tabelle 3.6: Übersicht über die 10 längsten Teilschritte der GAMS-Zeit im *EHighwaySM*-Szenario (12h/r20911)

Dauer/s	Speicher/GB	Typ	Bezeichnung (Anzahl der Elemente)
83,596	1,700	Assignment	TDCAgg_exportNodeToNode (182500)
83,539	1,702	Assignment	TDCAgg_importNodeToNode (182500)
23,449	1,249	Assignment	FuelsAACost_fuelConsumption_Save (2041080)
7,248	1,879	Assignment	ConvBase_generatedPower_Save (2505360)
5,334	4,819	Unload	
4,752	1,338	Solve Fini	OptiMoSystem (11669831)
4,340	3,237	Assignment	StorStd_DischargePower_Out (2058600)
4,268	3,367	Assignment	StorStd_StorageLevel_Out (2058600)
4,110	3,107	Assignment	StorStd_ChargePower_Out (2058600)
3,744	$857 \cdot 10^{-3}$	Solve Read	OptiMoSystem

Tabelle 3.7: Übersicht über die 10 längsten Teilschritte der *GAMS*-Zeit im *EHighwaySM*-Szenario (12h/r20778)

Dauer/s	Speicher/GB	Typ	Bezeichnung (Anzahl der Elemente)
81,046	$971 \cdot 10^{-3}$	Assignment	TDCAgg_importNodeToNode (182500)
80,913	$965 \cdot 10^{-3}$	Assignment	TDCAgg_exportNodeToNode (182500)
3,331	1,071	Solve Fini	OptiMoSystem (11669831)
2,880	$566 \cdot 10^{-3}$	Solve Read	OptiMoSystem
2,799	2,580	Unload	
2,716	1,017	Equation	Eq_LoadCoverage (42340)
2,659	$730 \cdot 10^{-3}$	Assignment	FuelsAACost_fuelConsumption_Save (2041080)
2,513	1,040	Equation	Eq_TotalPowerGeneration (42340)
2,503	$913 \cdot 10^{-3}$	Equation	EQ_dgs_TotalPowerGeneration (42340)
2,503	$945 \cdot 10^{-3}$	Equation	EQ_fCap_TotalPowerGeneration (42340)

Ergebnisdiskussion

Die *Set-Order*-Änderungen zeigen kein eindeutig positiv oder negatives Verhalten auf. Abhängig von Szenario, zeitlicher Konfiguration und verwendeter *GAMS*-Versionen ergeben sich Veränderungen von deutlicher Verschlechterung bis zu leichten Verbesserungen (unter 5 %). Bei kleineren Zeitschritten, also größerer Komplexität, scheint die neue *Set-Order* dabei besser abzuschneiden, was durch die größeren Mengen erklärbar ist. Da besonders bei diesen Konfigurationen, die eine lange Laufzeit benötigen, Verbesserungen erwünscht sind, kann die neue *Set-Order* zumindest bei manchen Parametrierungen eine Möglichkeit sein, die *GAMS*-Zeit zu verringern. Möglichkeiten zur Verbesserung bestehen in einer selektiven Anwendung der *Set-Order*-Änderungen auf einzelne Teile des *GAMS*-Code und unter Berücksichtigung der gewählten zeitlichen Konfiguration.

4 Zusammenfassung und Ausblick

In dieser Arbeit werden verschiedene Beschleunigungsstrategien für optimierende Energiesystemmodelle untersucht, speziell die räumliche Skalierung, *Rolling Horizon* und Änderungen der Formulierung (*Set-Order*). Die räumliche Skalierung in Abschnitt 3.1.1 auf Seite 8 zeigt große Einsparungsmöglichkeiten bei moderaten Änderungen des Zielwerts auf, allerdings mit großen Unterschieden zwischen den Modellen *Transgrid2* und *INTEEVER*. Bei Skalierung über die komplette Knotenanzahl ergeben sich allerdings Teilbereiche mit ungewünschtem Verhalten, in denen kurzzeitig Rechenzeit und Zielwert große Sprünge durchlaufen. Diese Auffälligkeiten müssen beachtet werden und benötigen genauere Untersuchungen; ebenso die Wahl des Clusteringalgorithmus. Der Verlauf der *CPLEX*-Zeit über der Clusteranzahl ist auch stark von der Parametrierung abhängig: schon die Wahl des Leitungsausbaus zeigte im *INTEEVER*-Projekt großen Einfluss. Die Unterschiede zwischen *Transgrid2* und *INTEEVER* lassen darüber hinaus keine allgemeinen Aussagen über den Verlauf der Rechenzeit zu.

Die Verringerung der Modulauswahl in Abschnitt 3.1.2 auf Seite 23 zeigt keine signifikanten Unterschiede auf, was die Signifikanz der Parametrierung unterstreicht. Eine Reduktion der Modulauswahl ohne Evaluierung der Komplexitätseinflüsse der verwendeten Module liefert wie gezeigt keine deutliche Reduktion der Rechenzeit. Eine Betrachtung der Einflüsse von Speicher, bzw. E-Mobilität benötigt wie erwähnt eine passende Parametrierung, welche einerseits einen deutlichen Einfluss auf die Komplexität des Optimierungsproblems aufweist und andererseits nicht die Abhängigkeit so überhöht, dass eine Übertragung auf praktisch vorkommende Speicherszenarien nicht mehr gegeben ist.

Die Untersuchung der *Rolling Horizon*-Methode in Abschnitt 3.2 auf Seite 26 zeigt großes Potential für Rechenzeiteinsparungen von über 50 %, allerdings mit einem deutlichen Anstieg des Zielwerts. Durch Änderungen in der Parametrierung der *Rolling Horizon*-Methode, wie die Erhöhung der Überlappung zwischen den Zeitabschnitten, oder dem Einsatz der Heuristik wie in Abschnitt 3.2 auf Seite 32 beschrieben, könnte der Zielwert auch in einem komplexen Szenario wie *INTEEVER* dem Referenzwert weiter angenähert werden und die Genauigkeit auch bei Einsatz von *Rolling Horizon* erhalten werden.

Zuletzt wird die Umformulierung des *GAMS*-Code in *GAMS* durch *Set-Order*-Änderungen in Abschnitt 3.3 auf Seite 33 untersucht. Diese ist attraktiv, da sie das Optimierungsproblem nicht reduziert und das Potential besitzt, die Zeit in *GAMS* zu verringern, was besonders bei Modellen, in denen nicht die *CPLEX*-Zeit dominiert, interessant ist. Allgemein zeigen sich allerdings keine Einsparungen, sondern ein gestreutes Bild über verschiedene zeitliche Auflösungen und Szenarien. Bei zeitlich feiner Auflösung und entsprechend hoher Komplexität zeigt ein Szenario mit geringer

Ausbauoptimierung leichte Vorteile bei *GAMS*- und *CPLEX*-Zeiten (-6% und -1%). Die *Set-Order*-Änderungen scheinen also auf wenige Teilschritte, insbesondere Zuweisungen mit vielen Elementen, negative Auswirkungen, auf die restlichen Schritte jedoch positiven Einfluss zu haben. Eine selektive Anwendung der *Set-Order*-Änderungen könnte also in deutlicheren Vorteilen resultieren.

Literatur

- [1] Karl-Kiên Cao, Ambros Gleixner und Matthias Miltenberger. “Methoden zur Reduktion der Rechenzeit linearer Optimierungsmodelle in der Energiewirtschaft – Eine Performance-Analyse”. In: *14. Symposium Energieinnovation*. Feb. 2016. URL: <http://elib.dlr.de/103136/>.
- [2] GAMS Development Corporation. *General Algebraic Modeling System (GAMS) Release 24.7*. Washington, DC, USA. 2016. URL: <http://www.gams.com/>.
- [3] IBM. *IBM ILOG CPLEX Optimization Studio CPLEX Users Manual V12.6*.
- [4] Julien F. Marquant, Ralph Evins und Jan Carmeliet. “Reducing Computation Time With a Rolling Horizon Approach Applied To a Milp Formulation of Multiple Urban Energy Hub System”. In: *Procedia Computer Science* 51.nil (2015), S. 2137–2146. DOI: 10.1016/j.procs.2015.05.486. URL: <https://doi.org/10.1016/j.procs.2015.05.486>.
- [5] Carsten Matke, Wided Medjroubi und David Kleinhans. *SciGRID - An Open Source Reference Model for the European Transmission Network (v0.2)*. Juli 2016. URL: <http://www.scigrid.de>.
- [6] Wided Medjroubi u. a. “Open Data in Power Grid Modelling: New Approaches Towards Transparent Grid Models”. In: *Energy Reports* 3.nil (2017), S. 14–21. DOI: 10.1016/j.egyr.2016.12.001. URL: <https://doi.org/10.1016/j.egyr.2016.12.001>.
- [7] Johannes Metzdorf. “Development and implementation of a spatial clustering approach using a transmission grid energy system model”. Betreuer am DLR: Karl-Kiên Cao. Masterarbeit. Universität Stuttgart, Aug. 2016. URL: <http://elib.dlr.de/106325/>.
- [8] Yvonne Scholz. “Renewable energy based electricity supply at low costs: development of the REMix model and application for Europe”. Diss. Universität Stuttgart, 2012, XII, 181 S.
- [9] Sebastian Schreck. *Implementation and Analysis of a Rolling Horizon Approach for the Energy System Model REMix*. 2016.
- [10] *scikit-learn: Overview of clustering methods*. URL: <http://scikit-learn.org/stable/modules/clustering.html>.

Anhang

Tabelle 4.1: Modulauswahl für minimale Module

Modul	Version
___ParameterAndOptions	0.3.6
___ValidateFunctionHelper	0.0.20
_StartScenario	0.1.5
ModuleVersionSummary	0.0.3
PyProcess	0.12.8
SaveGdx	0.1.1
TimeSeries_Global	0.0.1
Fuels	0.18.0
Nodes_SubAndSuper	0.10.10
Techs	0.18.0
Temporal_hoursYear	0.2.11
FuelsAndACost	0.20.0
Transport_DC_Aggregation	0.20.1
convBase	0.20.0
demand_electrical	0.20.0
pollutionAndACost	0.20.0
re_biomass	0.20.1
re_csp	0.20.1
re_fluctuatingNoStor	0.20.0
re_reservoirHydro	0.20.1
--ErrorsAndWarnings	0.5.3
-minCost_standart	0.20.0
freeParameterVariation	0.0.7

Tabelle 4.2: Modulauswahl für maximale Module

Modul	Version
___ParameterAndOptions	0.3.6
___ValidateFunctionHelper	0.0.20
_StartScenario	0.1.5
ModuleVersionSummary	0.0.3
PyProcess	0.12.8
SaveGdx	0.1.1
TimeSeries_Global	0.0.1
Fuels	0.18.0
Nodes_SubAndSuper	0.10.10
Techs	0.18.0
Temporal_hoursYear	0.2.11
__hy_Balance	0.10.20
_heat_demand	0.20.0
FuelsAndACost	0.20.0
Transport_DC_Aggregation	0.20.1
convBase	0.20.0
demand_electrical	0.20.0
eCars_smpl	0.20.1
heat_balance	0.20.0
heat_electricBoiler	0.20.1
heat_pump	0.20.1
hy_ElectrolyzerSimple	0.16.0
hy_FixedAnnualDemand	0.10.21
hy_Storage	0.16.0
pollutionAndACost	0.20.0
re_biomass	0.20.1
re_csp	0.20.1
re_fluctuatingNoStor	0.20.0
re_reservoirHydro	0.20.1
storageStd	0.20.2
--ErrorsAndWarnings	0.5.3
-minCost_standart	0.20.0
freeParameterVariation	0.0.7