



Diplomarbeit

**Matrixrobustheit
Algorithmen, Komplexität und
Anwendung zur
Netzwerküberwachung**

zur Erlangung des akademischen Grades
Diplom-Informatiker

Matthias Brosemann

Betreuer: Dipl.-Inform. Falk Hüffner
Prof. Dr. Rolf Niedermeier
Dr. Jochen Alber (DIgSILENT GmbH)

Abgabe: 16. Februar 2006

Theoretische Informatik I: Komplexitätstheorie
Institut für Informatik
Fakultät für Mathematik und Informatik
Friedrich-Schiller-Universität Jena

Kurzzusammenfassung. In der Elektrotechnik, speziell bei dem Betrieb von Stromnetzen, ist es wichtig, die aktuellen Größen, also Ströme, Spannungen usw. zu kennen bzw. zu berechnen, um Havariefällen vorzubeugen. In großen Verteilernetzen sind z. B. jedoch die Bedürfnisse der Verbraucher nicht hinreichend genug bekannt und zudem variabel. Daher werden Messgeräte eingesetzt, mithilfe derer spezielle Größen im Netz bestimmt werden können.

Diese Diplomarbeit untersucht die Problemstellung, bei der gefragt wird, ob ein gegebenes Netz auch bei einem Ausfall von k beliebigen Messgeräten überwachbar bleibt, d. h. ob die gewünschten Größen noch berechenbar sind. Mithilfe einer Arbeit von Jochen Alber und Markus Pöller wird dieses Problem auf das im Folgenden MATRIXROBUSTHEIT genannte Problem zurückgeführt: Hat eine gegebene, sogenannte Sensitivitätsmatrix nach dem Löschen von k beliebigen Zeilen noch vollen Rang? In der vorliegenden Arbeit werden zur Bestimmung der Komplexität von MATRIXROBUSTHEIT diverse Äquivalenzen zu anderen Problemformulierungen aufgezeigt, z. B. zu dem bekannten MINIMUM DISTANCE aus der Codierungstheorie. Mithilfe dieser wird die NP -Vollständigkeit für den Fall bewiesen, in dem die Sensitivitätsmatrix auf endliche Räume beschränkt ist. Die Komplexität des für die Praxis wichtigeren Falls, bei dem eine Sensitivitätsmatrix auf einem unendlich Raum gegeben ist, bleibt eine offene Fragestellung.

Außerdem werden Lösungsalgorithmen für MATRIXROBUSTHEIT entwickelt. Dazu zählt ein exakter Polynomialzeit-Algorithmus für einen Spezialfall, ein auf Mixed Integer Programming (MIP) basierender, exakter Algorithmus und eine nicht exakt rechnende Polynomialzeit-Heuristik, wobei der MIP-Ansatz und die Heuristik sowohl auf realistischen Daten, generiert aus Stromnetzwerken, als auch auf synthetischen Daten getestet wurde.

Schließlich werden interessante Beziehungen zu weiteren Problemen aufgezeigt, die Ansatzpunkte für fortführende Untersuchungen bieten könnten. In diesem Teil der Arbeit wird zusätzlich eine in der Literatur referenzierte Aussage vorgestellt, die eine Antwort auf die Komplexitätsfrage für MATRIXROBUSTHEIT bzgl. unendlicher Räume hätte geben können, deren Beweis jedoch einen in dieser Arbeit aufgezeigten Fehler beinhaltet.

Abstract. For the management of a power network it is essential to know or be able to calculate the current values, e.g. active and reactive powers, voltages etc. such that, in the event of an unexpected behavior emergency actions can be initiated. But in large distribution network some values change over the time. Thus, measurements are installed to determine the required values.

This diploma thesis analyzes the question whether a network remains observable, i.e. whether the required values are still calculable, even if any k measurements fail. Using a work of Jochen Alber and Markus Pöller this question leads to the problem MATRIX ROBUSTNESS: Does a given, so-called sensitivity matrix have full rank after dropping an arbitrary set of k rows? In this thesis divers equivalences to other problems are presented, e.g. to the well-known MINIMUM DISTANCE from coding theory. Using this equivalence the NP -completeness is proven when restricting the sensitivity matrix to finite fields. The complexity for infinite fields remains open.

Furthermore, algorithms solving MATRIX ROBUSTNESS are developed. This includes an exact polynomial-time algorithm for a special case, an exact algorithm based on mixed integer programming (MIP) and a non-exact polynomial-time heuristic. Both the MIP based algorithm and the heuristic are tested on realistic data generated from power network models and synthetic data sets.

Finally, some interesting relationships to further problems are presented that may provide starting points for future research. In addition, this part of the diploma thesis presents a flaw in a proof in the literature for the case of infinite fields that could have settled the question for the complexity of MATRIX ROBUSTNESS in the case of infinite fields.

Inhaltsverzeichnis

Abbildungsverzeichnis	iii
Tabellenverzeichnis	v
Algorithmenverzeichnis	vii
1 Einleitung	1
1.1 Motivation	1
1.2 Ziele und Inhalt	3
1.3 Observability Analysis und Optimal Meter Placement	5
1.4 Matrixrobustheit	6
1.5 Äquivalente Problemformulierungen	7
2 Komplexität	9
2.1 Matroide	9
2.1.1 Definition	10
2.1.2 Äquivalente Charakterisierungen	12
2.1.3 Dualität	16
2.1.4 Vektor-Matroide	17
2.2 Codierungstheorie	24
2.2.1 Codeabstände	27
2.2.2 Duale Codes, Generator- und Paritätskontrollmatrizen	28
2.3 Die Komplexität von Minimum Distance	33
3 Algorithmen	35
3.1 Matrixrobustheit bei einfacher linearer Abhängigkeit	35
3.2 Modellierung als Mixed Integer Linear Program (MIP)	38
3.2.1 Theoretische Entwicklung	38
3.2.2 Praktische Umsetzung	40
3.2.3 Verbesserungen	41

3.3	Heuristik	42
3.4	Experimentelle Ergebnisse vom MIP-Algorithmus und der Heuristik	44
3.4.1	Testergebnisse für Berechnungen von Stromnetzwerken	47
3.4.2	Testergebnisse für synthetische Daten	48
3.5	Weitere Ansätze	50
4	Verwandte Probleme	51
4.1	Kantenzusammenhang	51
4.2	Dominating Set auf Split-Graphen	56
4.3	Das Nullraum-Problem	59
4.3.1	Die Reduktion von Minimum Circuit auf das Nullraum-Problem	59
4.3.2	Fehlerhafte Reduktion von Clique auf Minimum Circuit	62
5	Zusammenfassung und Ausblick	65
	Literatur	67
	Danksagung	71

Abbildungsverzeichnis

1.1	Ausschnitt aus einem modellierten Stromnetz mit berechnetem Lastfluss . . .	2
1.2	Ausschnitt aus einem modellierten Stromnetz mit Zustandsschätzung	3
1.3	Ergebnisübersicht	4
2.1	Beispielgraph als Grundlage für einen Matroiden	11
2.2	Beispielgraph als Grundlage eines Matroiden	13
3.1	Beispiel zur Heuristik	44
3.2	Netzplan für das Beispiel „Nine Bus System“	45
3.3	Netzplan für das Beispiel „Treelike System“	45
3.4	Netzplan für das Beispiel „MV/LV Distribution“	46
3.5	Laufzeit und Lösungsqualität für Zufallsmatrizen	49
4.1	Beispiel einer Kantenmatrix eines Graphen	52
4.2	Beispiel zur Abbildung zwischen Fluss und Linearkombination	55
4.3	Graph für das Gegenbeispiel zu McCormicks <i>NP</i> -Härtebeweis für MINIMUM CIRCUIT	63
4.4	Spärlichkeitsmuster und Matrix für das Gegenbeispiel zu McCormicks <i>NP</i> - Härtebeweis für MINIMUM CIRCUIT	64

Tabellenverzeichnis

3.1	Eigenschaften der Beispielprobleme	47
3.2	Experimentelle Ergebnisse mit dem MIP	48
3.3	Experimentelle Ergebnisse mit der Heuristik	49

Algorithmenverzeichnis

3.1	Exakte Lösung von MATRIXROBUSTHEIT für Sensitivitätsmatrizen mit ausschließlich paarweise linearen Abhängigkeiten	36
3.2	Berechnung der MINIMALEN ROBUSTEN SUBMATRIX von Sensitivitätsmatrizen mit ausschließlich paarweise linearen Abhängigkeiten	37
3.3	MIP zur (Teil-)Lösung von UMFASSENDSTE HYPEREBENE	41
3.4	Heuristik basierend auf LPs zur Lösung von MATRIXROBUSTHEIT mit polynomialer Laufzeit	43
4.1	Lösung von MATRIXROBUSTHEIT mittels DOMINATING SET	57
4.2	Greedy-Algorithmus für das NULLRAUM-Problem	62

Kapitel 1

Einleitung

1.1 Motivation

„Elektrischer Strom ist heute eine der wichtigsten Methoden, Energie zu transportieren und bereitzustellen. So wird heute die gesamte Beleuchtung, die meisten Haushaltsgeräte und die gesamte Elektronik und Rechnertechnik mit elektrischer Energie betrieben. Autos mit elektrischem Antrieb werden als umweltfreundliche Alternative zu herkömmlichen benzinbetriebenen Gefährten propagiert. Auch medizinische Geräte vom Röntgenapparat über den Kernspintomographen bis hin zum Zahnarztbohrer werden mit elektrischer Energie betrieben.“

So wird in der freien Enzyklopädie Wikipedia [Wik05] die technische Nutzung des Stroms beschrieben und damit die Wichtigkeit von elektrischem Strom aufgezeigt. Denn für die meisten Bewohner der Industrieländer ist ob der vielfältigen Nutzung elektrischer Energie ein Leben ohne Strom nur noch schwer vorstellbar. Als Kernaussage lässt sich festhalten, dass elektrischer Strom derzeit das wichtigste Energietransportmittel ist. Dieser Transport geschieht bekanntlich über Stromleitungen, die zu Stromnetzen zusammen geschaltet sind.

Eine wichtige Aufgabe des Netzbetreibers ist es, den Zustand des Stromnetzes zu überwachen, um Havariefälle frühzeitig zu erkennen und zu vermeiden. Dazu wird als Grundlage das Stromnetz mathematisch als Gleichungssystem modelliert. Dabei müssen zunächst für jeden Knoten die sogenannten Kirchhoff'schen Sätze festgehalten werden, die dem Fluss-erhaltungsgesetz aus der Graphentheorie entsprechen. Als Verbindungen zwischen den Knoten (Zweigen) gibt es verschiedene Elemente, wie z. B. einfache Leitungen oder Transformatoren. Diese erhalten jeweils einen geeigneten Satz an Modellgleichungen. Zusätzlich liegen empirische (zeitabhängige) Daten für die Punkte im Netz vor, wo Leistung eingespeist bzw. abgenommen wird.

Mit den Knotensummen, den Modellgleichungen und den Daten als Einspeise- und Abnahmegleichungen kann durch Lösung des entstehenden Gleichungssystems der sogenannte *Lastfluss* des Stromnetzes berechnet werden. Dabei werden z. B. Variablen wie der Strom an allen Zweigen und die Spannung an allen Knoten bestimmt, wie in Abbildung 1.1 dargestellt ist. Mithilfe des Lastflusses lässt sich überprüfen, ob ein gegebenes Netz unter den angenommenen Werten für Einspeisung und Abnahme keine kritischen Werte, wie z. B. einen zu hohen Strom auf einer Leitung, annimmt.

Im laufenden Betrieb eines Stromnetzes sollte jedoch die Verwendung von empirischen Daten vergangener Messungen eingeschränkt werden, da der aktuelle Zustand des Systems

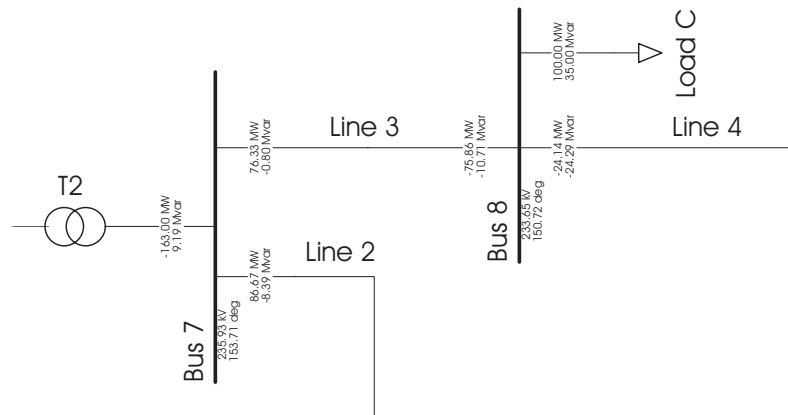


Abbildung 1.1: Ausschnitt aus einem modellierten Stromnetz mit berechnetem Lastfluss. Dargestellt sind zu jedem Zweig (Lines als einfache Leitungen, T2 als Transformator, Load C als Abnehmer) die Wirk- und Blindleistung und zu jedem Knoten (Bus) die Spannung und der Phasenwinkel.

berechnet werden muss. Statt der empirischen Daten für Einspeise- und Abnahmepunkte werden nun Messgeräte eingesetzt, um das Gleichungssystem zu vervollständigen. Da es jedoch sehr kostenaufwändig ist, an jedem Abnahme- bzw. Einspeisepunkt ein Messgerät zu installieren, werden diese an wirtschaftlich besser gewählten Stellen verwendet. Sind genügend Messgeräte vorhanden, könnte dieselbe Berechnung wie beim Lastfluss angesetzt werden, um das Gleichungssystem zu lösen. Allerdings gibt es Messfehler, Fehler in der Übertragung der Messwerte und andere Störfaktoren. Daher wird eine Lösung gesucht, die den Messwerten möglichst gut entspricht. Mathematisch ausgedrückt wird das quadratische Mittel der Abweichungen von gemessenen (calVal_i) und berechneten Werten (meaVal_i) minimiert:

$$f(x) = \sum_i \omega_i \cdot |\text{calVal}_i(x) - \text{meaVal}_i|^2.$$

Der Zustandsvektor x umfasst dabei alle Ströme und Phasenwinkel, sowie alle Variablen, die geschätzt werden sollen. Summiert wird über alle Messgeräte. Das Gewicht ω_i wird verwendet, um verschiedene Messgeräte-Genauigkeiten einfließen zu lassen. Diese Art und Weise den Zustand eines Stromnetzes (näherungsweise) zu berechnen wird *State Estimation* genannt. Sie wurde 1967 von Hodges eingeführt [Hod67] und für sie stehen eine Reihe von Vorgehensweisen zur Verfügung (für einen Überblick und weitere Referenzen siehe [MCR94]). Abbildung 1.2 zeigt denselben Ausschnitt wie Abbildung 1.1 mit dem Ergebnis einer State Estimation.

Eine solche Zustandsschätzung ist jedoch nur möglich, wenn genügend Messgeräte wohlverteilt im Netz vorhanden sind. Man bezeichnet ein Netz als (*vollständig*) *überwachbar*, wenn alle benötigten Zustandsvariablen durch die Messungen mit der oben genannten Formel geschätzt werden können [Cle90].

Die Bestimmung, ob ein Netz überwachbar ist oder nicht, wird auch als *Observability Analysis* bezeichnet. Wird ein System nicht überwachbar, müssen neue Messgeräte installiert werden. Dies kann passieren, wenn Messgeräte ausfallen oder Veränderungen an der Netzwerktopologie, wie z. B. das Teilen eines Busses oder das Abschalten einer Leitung vorgenommen werden. Dann muss entschieden werden, welche Messgeräte an welchen Orten installiert werden sollten, damit das Netz möglichst kostengünstig wieder überwachbar wird. Diese Aufgabenstellung ist als *Optimal Meter Placement* bekannt. Während das

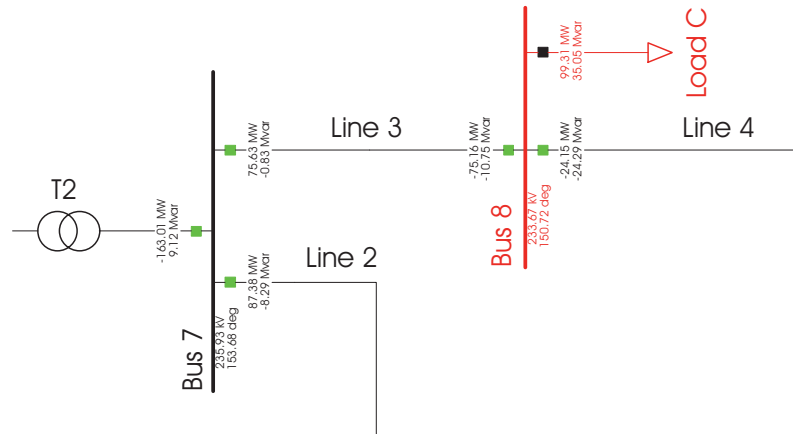


Abbildung 1.2: Ausschnitt aus einem modellierten Stromnetz mit Zustandsschätzung. Rote Farbe steht für eine benötigte Zustandsvariable. Die Quadrate stellen mögliche Positionen für Messgeräte dar, wobei die grün eingefärbten mit solchen bestückt tatsächlich sind.

Optimal Meter Placement eher vereinzelt angewandt wird (bei der Planung eines Netzes oder bei einem Ausfall von Messgeräten), ist die Observability Analysis als Teil des State Estimation-Prozesses in Energie-Management-Systemen [Din04, AP05] integriert.

Die Motivation dieser Arbeit ist es, die Messung eines gegebenen Netzes (vor der Inbetriebnahme) zu einem gewissen Grad ausfallsicher zu machen. In der Literatur gibt es dazu bereits einige, meist heuristische Ansätze, wobei der Ausfall einzelner Messgeräte und einzelner Leitungen, sowie deren Zusammenwirken im Vordergrund stehen [PMCK88, BZZG95, MA00, AM01, HLA03]. Dabei werden vielfältige Arten von Algorithmen vorgestellt, wie genetische Algorithmen [EZ04] oder Simulated Annealing [ATF01]. Nur wenige Arbeiten beschäftigen sich jedoch mit kombinatorisch komplexeren Szenarien, wie der Absicherung gegenüber dem Ausfall von mehreren Messgeräten und Leitungen [DA02].

An dieser Stelle setzt die vorliegende Arbeit an, die die algorithmische Komplexität einer solchen Problemstellung untersucht. Dabei wird ausschließlich auf den Ausfall von Messgeräten eingegangen. Exakt gesprochen soll der Ausfall von bis zu k Messgeräten die Observierbarkeit des Netzes nicht beeinflussen. Als Ausgangspunkt wird dazu eine Sensitivitätsmatrix angenommen, wie sie in [AP05] eingeführt wurde. Dies führt zu den Problemstellungen der MATRIXROBUSTHEIT und der MINIMALEN ROBUSTEN SUBMATRIX, wobei letztere im Sinne von elektrischen Netzwerken als *Robust Meter Placement* bezeichnet werden könnte.

1.2 Ziele und Inhalt

Das Hauptziel der Untersuchungen zur MATRIXROBUSTHEIT ist die Bestimmung der Komplexität des Problems. Dazu werden mehrere, im Sinne der polynomialen Many-One-Reduktion¹ verwandte Problemstellungen betrachtet, die in Abbildung 1.3 dargestellt sind.

¹Eine Menge A ist many-one-reduzierbar auf eine Menge B , kurz $A \leq_m B$, wenn eine berechenbare, total definierte Funktion f existiert, so dass $x \in A \Leftrightarrow f(x) \in B$. Die Funktion f heißt polynomiale Many-One-Reduktion, wenn sie in Polynomialzeit berechenbar ist. Dies wird dann als $A \leq_m^p B$ geschrieben.

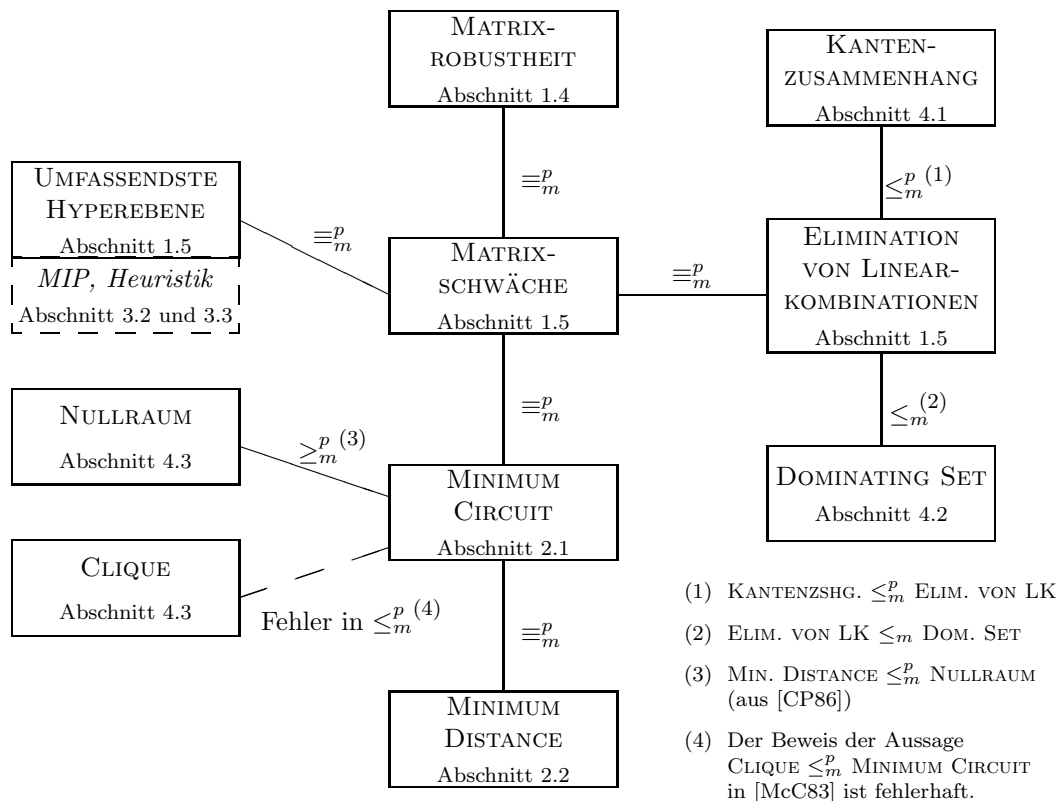


Abbildung 1.3: Ergebnisübersicht. Die Symbole \leq_m^p und \equiv_m^p bezeichnen die polynomiale Many-One-Reduktion bzw. Äquivalenz. Bis auf die Problemstellung UMFASSENDSTE HYPEREBENE, zu der Lösungsalgorithmen entwickelt werden, sind alle Probleme Teil der Komplexitätsuntersuchung von MATRIXROBUSTHEIT.

Zunächst wird in diesem Kapitel das Problem MATRIXROBUSTHEIT definiert und dazu verschiedene einfache Äquivalenzen vorgestellt, nämlich zu MATRIXSCHWÄCHE, UMFASSENDSTE HYPEREBENE und ELIMINATION VON LINEARKOMBINATIONEN.

Ausgehend von MATRIXSCHWÄCHE wird anschließend in Kapitel 2 über MINIMUM CIRCUIT aus der Matroidtheorie als Zwischenschritt die Äquivalenz zu MINIMUM DISTANCE aus der Codierungstheorie gezeigt. Damit können vorhandene Komplexitätsbetrachtungen zu MINIMUM DISTANCE auf MATRIXROBUSTHEIT übertragen werden. Eins dieser Ergebnisse ist die *NP*-Vollständigkeit von MINIMUM DISTANCE.

Wegen der *NP*-Vollständigkeit beschäftigt sich Kapitel 3 mit der Entwicklung eines Algorithmus für MATRIXROBUSTHEIT, der auf Mixed Integer Linear Programs (MIP) basiert. Außerdem wird eine davon abgeleitete Heuristik angegeben. Beide betrachten die zu MATRIXROBUSTHEIT äquivalente Problemstellung UMFASSENDSTE HYPEREBENE.

Neben dem Beweis der *NP*-Härte von MATRIXROBUSTHEIT und der Entwicklung von Algorithmen dazu gibt es eine Reihe von weiteren Ergebnissen, die für die zukünftige Forschung von Interesse sein könnten. Darunter fallen die Zusammenhänge zu den Problemen KANTENZUSAMMENHANG und DOMINATING SET in Graphen. Weiterhin kann das sogenannte NULLRAUM-Problem, welches zum Lösen von Gleichungssystemen verwendet wird, auf MINIMUM CIRCUIT reduziert werden. Der in der Literatur zitierte Beweis zur *NP*-Härte von MINIMUM CIRCUIT, der in diesem Zusammenhang in [McC83] geführt wird, ist allerdings fehlerhaft. Dieser Fehler wird zusammen mit den genannten Reduktionen in Kapitel 4

vorgestellt. Dank der Äquivalenz von MINIMUM CIRCUIT zu MINIMUM DISTANCE bleibt die getroffene Aussage zur Komplexität von NULLRAUM jedoch erhalten.

1.3 Observability Analysis und Optimal Meter Placement

Um eine Zustandsschätzung eines Stromnetzes zu ermöglichen, müssen genügend Datenquellen (Messgeräte) auf die richtige Art und Weise im Netz verteilt vorhanden sein. Mit der sogenannten Observability Analysis wird dies überprüft. Methoden zum Optimal Meter Placement dagegen versuchen möglichst kostengünstig in nicht überwachbaren Netzwerken Positionen für zusätzliche Messgeräte zu finden, so dass die Zustände (Variablen, die geschätzt werden sollen) überwachbar sind.

In Abschnitt 1.1 wurde die folgende Definition gegeben: Ein Netzwerk ist genau dann observierbar, wenn die State Estimation möglich ist. Diese Definition ist jedoch abhängig von der Art und Weise, wie die State Estimation durchgeführt wird. Daher gibt es verschiedene Definitionen für Observierbarkeit, die letztendlich angewandt werden.

Für die erste Definition wird die sogenannte Measurement-Jacobi-Matrix H oder die damit eng verwandte Gain-Matrix G herangezogen, welche vollen Rang besitzen und wohlkonditioniert sein müssen, damit die State Estimation durchführbar ist. Dies wird auch als *numerische Observierbarkeit* bezeichnet [Bre96, Gou05, HI01]. Ein Vorteil der numerischen Observierbarkeit ist, dass die dazu notwendigen Routinen (Dreieckszerlegung von Matrizen, Gauß-Elimination) bereits in einem State Estimator implementiert sind.

Aufgrund der Algorithmen, die zur Bestimmung der oben definierten Observierbarkeit herangezogen werden, entstehen u. a. durch die limitierte Wortlänge in den Rechnern numerische Fehler. Dies kann zu Folge haben, dass das Ergebnis nicht korrekt ist. Daher gibt es weitere Ansätze Observierbarkeit zu definieren, deren Umsetzungen ohne numerische Operationen auskommen. Eine davon ist die sogenannte *topologische Observierbarkeit*. Sie gilt genau dann, wenn sich in dem als Graph interpretierten Netz ein Spannbaum² mit einer Messwertzuordnung finden lässt [BLAB05]. Eine Messwertzuordnung ist eine Zuordnung von je einem Messgerät zu jeder Kante des Spannbaums, wobei folgende Eigenschaften gelten müssen:

- Verschiedenen Kanten werden verschiedene Messgeräte zugeordnet.
- Ist das Messgerät für eine Kante eine Einspeisungsmessung (Injection Measurement), so muss es an einem der Endpunkte der Kante installiert sein.
- Ist das Messgerät für eine Kante eine Flussmessung (Flow Measurement), so muss es den Fluss der Kante entsprechenden Leitung messen.

Zu diesem recht spezialisiertem Ansatz lassen sich einige Arbeiten finden [MT91, JKBT03, KKCD03, BLAB05, JBT⁺05], wobei Ausnahmen zu verzeichnen sind, die zur Bestimmung der topologischen Observierbarkeit auch numerische Methoden verwenden [KK03].

Die dritte Variante ist eine Abschwächung der numerischen Überwachbarkeit und wird *symbolische* bzw. *algebraische Observierbarkeit* genannt. Sie gilt genau dann, wenn die Measurement-Jacobi-Matrix H vollen Rang besitzt. Die Kondition der Matrix wird nicht

²Ein Spannbaum ist in der Graphentheorie ein Teilgraph eines ungerichteten Graphen, der ein Baum (kreisfrei und zusammenhängend) ist und alle Knoten des Ausgangsgraphen enthält.

berücksichtigt. Dadurch ist es möglich nicht-numerische Verfahren zu entwickeln [Che90, BL98].

In dieser Arbeit wird ein neuer Ansatz verfolgt. Er basiert auf der Sensitivitätsmatrix aus [AP05], die es erlaubt alle Arten von Messungen abzudecken. Die Definition der Observierbarkeit entspricht jedoch der algebraischen. Allerdings lassen sich die oben genannten Arbeiten zur algebraischen Observierbarkeit nicht übertragen, da nun eine Matrix anderen Ursprungs verwendet wird als jene, auf denen diese Arbeiten aufbauen. Außerdem sollen nun zusätzlich multiple Ausfälle von Messgeräten berücksichtigt werden.

1.4 Matrixrobustheit

Als Ansatzpunkt für die Untersuchung der Komplexität der Robustheit einer Netzwerküberwachung werden Sensitivitätsmatrizen verwendet, die wie folgt definiert sind.

Definition 1.1

Gemessen seien m Größen $Y = (y_1, \dots, y_m)$ mit denen n Zustände $X = (x_1, \dots, x_n)$ geschätzt werden sollen. Die Sensitivitätsmatrix ist definiert als

$$S_X^Y := \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \dots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \dots & \frac{\partial y_m}{\partial x_n} \end{pmatrix}.$$

Dabei bezeichnet $\frac{\partial y_i}{\partial x_j}$ die Sensitivität von y_i bzgl. x_j .

Die Sensitivität einer Größe y_i bzgl. eines Zustands x_j gibt an, wie eine Veränderung des Zustands x_j das Messgerät y_i beeinflusst. Da die Beeinflussung nicht linear ist, wird die Sensitivität numerisch durch infinitesimale Störungen bestimmt.

In [AP05] werden nun die folgenden Aussagen getroffen. Ist ein Messgerät-Sensitivitätsvektor

$$\text{sens}_X(y) := \left(\frac{\partial y}{\partial x_1}, \dots, \frac{\partial y}{\partial x_n} \right)$$

eines Messgeräts y als Linearkombination von anderen Messgerät-Sensitivitätsvektoren darstellbar, so enthält $\text{sens}_X(y)$ keine „neuen Informationen“ gegenüber den anderen Vektoren. Eine Menge y_{i_1}, \dots, y_{i_p} enthält somit genau dann redundante Messungen, wenn die Messgerät-Sensitivitätsvektoren zu y_{i_1}, \dots, y_{i_p} linear abhängig sind.

Wenn andererseits die Zustand-Sensitivitätsvektoren

$$\text{sens}^Y(x_{i_1}) := \left(\frac{\partial y_1}{\partial x_{i_1}}, \dots, \frac{\partial y_m}{\partial x_{i_1}} \right)$$

und $\text{sens}^Y(x_{i_2})$ für die zwei Zustände x_{i_1} und x_{i_2} linear abhängig sind, so kann zwischen den beiden Zuständen nicht unterschieden werden. Sie sind somit nur als Gruppe überwachbar, jedoch nicht individuell. Allgemein gesprochen ist eine Menge von Zuständen x_{i_1}, \dots, x_{i_q} genau dann (individuell) überwachbar, wenn die korrespondierenden Zustand-Sensitivitätsvektoren linear unabhängig sind.

Aufgrund dieser Beobachtungen wird in [AP05] Observierbarkeit wie folgt definiert, wobei jedoch beachtet werden muss, dass die unvermeidliche Diskretisierung der Sensitivitäten einen Einfluss auf die linearen Abhängigkeiten von Vektoren hat.

Definition 1.2

Eine Menge X von n Zuständen ist durch eine Menge Y von m gemessenen Größen genau dann überwachbar, wenn die $m \times n$ -Sensitivitätsmatrix S_X^Y den Rang n besitzt.

Mit diesem Hintergrundwissen kann nun das Problem auf einfache Art und Weise formuliert werden, wobei zwischen einer Entscheidungs- und einer Optimierungsvariante unterschieden werden kann. Da sehr einfach getestet werden kann, ob eine Matrix vollen Rang besitzt, wird im Folgenden davon ausgegangen, dass eine Sensitivitätsmatrix vollen Rangs vorliegt.

Problem: MATRIXROBUSTHEIT

Instanz: Eine $m \times n$ -Sensitivitätsmatrix M über einem beliebigen Körper K mit vollem Rang n , $m \geq n$ und eine positive ganze Zahl k .

Frage: Ist die Sensitivitätsmatrix robust gegenüber dem Löschen von k Zeilen, d. h. bleibt der Rang der Sensitivitätsmatrix stets beim Löschen von k beliebigen Zeilen erhalten?

Problem: MINIMALE ROBUSTE SUBMATRIX

Instanz: Eine $m \times n$ -Sensitivitätsmatrix M über einem beliebigen Körper K mit vollem Rang n , $m \geq n$ und eine positive ganze Zahl k .

Ziel: Finde die kleinste $m' \times n$ -Submatrix der Sensitivitätsmatrix (also $m' \leq m$), die robust gegenüber dem Löschen von k Zeilen ist.

Diese Arbeit konzentriert sich auf die Untersuchung von MATRIXROBUSTHEIT, während auf das Problem der MINIMALEN ROBUSTEN SUBMATRIX nur in einem einfachen Spezialfall in Abschnitt 3.1 eingegangen wird.

1.5 Äquivalente Problemformulierungen

Ausgehend von dem Entscheidungsproblem MATRIXROBUSTHEIT werden im Folgenden einige weitere Formulierungsvarianten festgehalten, die äquivalent im Sinne des Many-One-Reduktionsbegriffes sind.

Komplement Statt der Frage, ob die Sensitivitätsmatrix robust gegenüber dem Löschen von k Zeilen ist, kann auch geprüft werden, ob sie es nicht ist, was genau dann der Fall ist, wenn k Zeilen der Sensitivitätsmatrix ausgewählt werden können, so dass sich nach deren Löschen der Rang der Sensitivitätsmatrix verringert.

Problem: MATRIXSCHWÄCHE

Instanz: Eine $m \times n$ -Sensitivitätsmatrix M über einem beliebigen Körper K mit vollem Rang n , $m \geq n$ und eine positive ganze Zahl k .

Frage: Gibt es k Zeilen der Sensitivitätsmatrix, so dass die Submatrix, die durch Löschen dieser k Zeilen entsteht, nicht mehr den Rang n hat?

Untersuchung der linearen Abhängigkeiten Es bezeichne $ZV(M)$ die Menge der Zeilenvektoren einer Matrix M . Die Verringerung des Ranges der Sensitivitätsmatrix tritt offensichtlich genau dann ein, wenn es einen Vektor $x \in X$ aus einer Menge zu löschender Vektoren $X \subseteq ZV(M)$ gibt, so dass x nicht als Linearkombination von Vektoren aus $ZV(M) \setminus X$ dargestellt werden kann. Somit ist die Frage, die in MATRIXSCHWÄCHE gestellt wird, äquivalent zu der folgenden Problemstellung.

Problem: ELIMINATION VON LINEARKOMBINATIONEN

Instanz: Eine $m \times n$ -Sensitivitätsmatrix M über einem beliebigen Körper K mit Rang n , $m \geq n$ und eine positive ganze Zahl k .

Frage: Gibt es eine Zeile, die nach dem Löschen von $k - 1$ anderen Zeilen der Sensitivitätsmatrix nicht mehr als Linearkombination der verbliebenen Vektoren dargestellt werden kann?

Finden einer Hyperebene Da die Sensitivitätsmatrix vollen Rang besitzt, spannen ihre Zeilenvektoren einen n -dimensionalen Raum auf. Reduziert sich der Rang um eins, so spannen die Zeilenvektoren nur noch einen $(n - 1)$ -dimensionalen Raum auf. Somit ist es äquivalent zum Problem MATRIXSCHWÄCHE zu fragen, ob es eine Hyperebene gibt, die mindestens $n - k$ Zeilenvektoren enthält. Denn die restlichen k Vektoren sind diejenigen, die gelöscht werden müssen, damit der Rang der Matrix fällt.

Problem: UMFASSENDSTE HYPEREBENE

Instanz: Eine $m \times n$ -Sensitivitätsmatrix M über einem beliebigen Körper K mit Rang n , $m \geq n$ und eine positive ganze Zahl k .

Frage: Gibt es eine Hyperebene in dem Raum K^n , die mindestens $n - k$ viele Zeilenvektoren aus M enthält?

Die hier vorgestellten Problemformulierungen werden im Folgenden genutzt, um die Komplexität von MATRIXROBUSTHEIT zu bestimmen. Während UMFASSENDSTE HYPEREBENE Grundlage der in Kapitel 3 entwickelten Algorithmen ist, wird MATRIXSCHWÄCHE als Basis für den Beweis von Äquivalenzen zu Problemen aus der Matroid- und der Codierungstheorie in Kapitel 2 und ELIMINATION VON LINEARKOMBINATIONEN als Ausgangspunkt für Beziehungen zu graphentheoretischen Problemen in Kapitel 4 verwendet.

Kapitel 2

Komplexität

Günstig für Komplexitätsbetrachtungen zu einem neuen Problem, wie in diesem Falle MATRIXROBUSTHEIT bzw. MATRIXSCHWÄCHE, ist die Reduktion auf geläufigere, evtl. bereits untersuchte Problemstellungen.

Dieses Kapitel zeigt die Äquivalenz zwischen MATRIXSCHWÄCHE und einer verallgemeinerten Version des Problems MINIMUM DISTANCE aus der Codierungstheorie auf, wodurch u. a. die NP -Vollständigkeit von MINIMUM DISTANCE auf MATRIXROBUSTHEIT mit endlichem Körper K übertragen werden kann. Die Äquivalenz beider Probleme wird unter Verwendung des Problems MINIMUM CIRCUIT aus der Matroidtheorie als Zwischenschritt gezeigt.

In Abschnitt 2.1 findet sich der Beweis der Äquivalenz von MATRIXSCHWÄCHE und MINIMUM CIRCUIT, der viel Vorarbeit bedarf. Die Äquivalenz zwischen MINIMUM CIRCUIT und MINIMUM DISTANCE ist dagegen relativ einfach. Neben dem Beweis hierzu erläutert Abschnitt 2.2 zudem die genaue Herkunft und Namensgebung von MINIMUM DISTANCE. Der letzte Abschnitt dieses Kapitels enthält einen Überblick zu den Komplexitätsuntersuchungen zu MINIMUM DISTANCE und entsprechende Aussagen zur Komplexität von MATRIXROBUSTHEIT.

2.1 Matroide

In diesem Abschnitt wird gezeigt, dass das Problem der MATRIXSCHWÄCHE äquivalent zu dem MINIMUM CIRCUIT-Problem ist, welches wie folgt definiert ist.

Problem: MINIMUM CIRCUIT

Instanz: Eine $m \times n$ -Matrix M über einem beliebigen Körper K und eine positive ganze Zahl k .

Frage: Gibt es eine k -elementige, linear abhängige Menge von Spaltenvektoren in M ?

Der Begriff Minimum Circuit entspringt der Matroidtheorie, in der minimale (linear) abhängige Mengen auch als Kreise bezeichnet werden. Im Folgenden wird eine Einführung in diese Theorie gegeben, da Matroide und der zugehörige Dualitätsbegriff das zentrale Hilfsmittel im Beweis der Äquivalenz von MATRIXSCHWÄCHE und MINIMUM CIRCUIT

sind. Der Definition von Matroiden folgen Sätze zu äquivalenten Charakterisierungen. Diese werden zur Einführung des Begriffs des dualen Matroiden benötigt. Schließlich wird gezeigt, dass die Klasse der Vektormatroide (Matroide, die aus Matrizen gebildet werden) abgeschlossen bzgl. der Dualität ist. Damit kann die oben genannte Äquivalenz im Theorem 2.29 gezeigt werden.

Der Inhalt dieses Kapitels setzt sich hauptsächlich aus Matroid-Theorie und linearer Algebra zusammen. Die hier vorgestellten Erkenntnisse über Matroide lassen sich u. a. in [Oxl04] oder [Wel76] nachlesen, die beide einen umfassenden Einblick in die Mächtigkeit der Matroid-Theorie geben. Im Folgenden wird jedoch hauptsächlich auf [Oxl04] Bezug genommen (Quellenangabe der Beweise). Auf der anderen Seite sind die in Abschnitt 2.1.4 benötigten Aussagen der linearen Algebra bekannte Ergebnisse, die in den meisten ausführlicheren Werken über lineare Algebra, z. B. [Ant04] zu finden sind. Die vorgestellte Theorie wird schließlich angewandt, um die Äquivalenz zwischen MATRIXSCHWÄCHE und MINIMUM CIRCUIT zu beweisen.

2.1.1 Definition

Ein Matroid stellt ein System \mathcal{U} von Mengen über einer Grundmenge S dar, die als unabhängig gelten sollen. Dabei müssen einige Eigenschaften gelten, die diese Unabhängigkeit charakterisieren.

Definition 2.1

Ein Paar (S, \mathcal{U}) mit $S \supseteq \bigcup \mathcal{U}$ heißt Matroid, falls die folgenden drei Axiome gelten:

- (I1) $\emptyset \in \mathcal{U}$
- (I2) $A \in \mathcal{U}, B \subseteq A \Rightarrow B \in \mathcal{U}$
- (I3) $A, B \in \mathcal{U}, |B| > |A| \Rightarrow \exists a \in B \setminus A : A \cup \{a\} \in \mathcal{U}$

Für den Umgang mit Matroiden werden nun eine Reihe von Bezeichnungen eingeführt.

Bezeichnung 2.2

Sei $M = (S, \mathcal{U})$ ein Matroid. Dann gelten folgende Bezeichnungen:

1. Die Mengen S und \mathcal{U} werden auch mit $S(M)$ bzw. $\mathcal{U}(M)$ bezeichnet.
2. Jedes $A \in \mathcal{U}$ heißt unabhängige Menge von M .
3. Jedes $A \subseteq S$ mit $A \notin \mathcal{U}$ heißt abhängige Menge von M .
4. Eine Menge $B \in \mathcal{U}$ heißt Basis von M , falls B eine maximale unabhängige Menge von M ist, d.h. $\forall B' \in \mathcal{U} : B \subseteq B' \Rightarrow B = B'$.
Die Menge aller Basen von M wird auch mit \mathcal{B} bzw. $\mathcal{B}(M)$ bezeichnet.
5. Eine Menge $C \subseteq S$ heißt Kreis von M , falls C eine minimale abhängige Menge von M ist, d.h. $\forall C' \subset C \notin \mathcal{U} : C' \in \mathcal{U}$.
Die Menge aller Kreise von M wird auch mit \mathcal{C} bzw. $\mathcal{C}(M)$ bezeichnet.

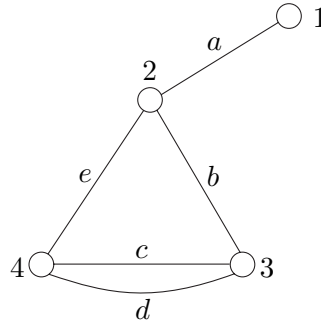


Abbildung 2.1: Beispielgraph als Grundlage für einen Matroiden

An dieser Stelle wird deutlich, dass Matroide das Konzept der linearen Unabhängigkeit aus der Vektorraumtheorie verallgemeinern, aus der offensichtlich die Begriffe der unabhängigen und abhängigen Mengen, sowie der Ausdruck Basis stammen. Zudem entspricht Axiom (I3) dem Austauschsatz von Steinitz, mit dessen Hilfe der Hauptsatz für endlich-dimensionale Vektorräume bewiesen werden kann.

Der Begriff Kreis kommt allerdings aus der Sprache der Graphentheorie. Denn auch Graphen können als Matroid aufgefasst werden, wie das folgende Beispiel verdeutlicht.

Beispiel 2.3

Gegeben sei der Graph (V, E) in Abbildung 2.1. Dieser Graph wird nun als Matrix mit $|V|$ Spalten und $|E|$ Knoten modelliert, indem jedem Knoten eine Spalte zugeordnet wird. Für jede Kante gibt es eine Zeile mit -1 an der Position des Knotens mit dem kleineren Index und 1 an der Position des Knotens mit dem größeren Index. Die restlichen Einträge in einer Zeile sind 0 . Für das Beispiel ergibt sich die folgende Matrix.

	a	b	c	d	e
1	-1	0	0	0	0
2	1	-1	0	0	-1
3	0	1	-1	-1	0
4	0	0	1	1	1

Fasst man die linear unabhängigen Mengen der Spalten dieser Matrix nun als Matroid auf (aufgrund der Eigenschaften von linear unabhängigen Mengen sind die Matroid-Axiome erfüllt), ergeben sich folgende Grundmenge S , sowie die unabhängigen und abhängigen Mengensysteme \mathcal{U} und $\mathcal{P}(S) \setminus \mathcal{U}$. Graphentheoretisch entspricht $\mathcal{P}(S) \setminus \mathcal{U}$ der Menge aller nicht kreisfreien Kantenmengen des Graphen. Der Ausdruck $\mathcal{P}(S)$ bezeichnet die Potenzmenge von S .

$$\begin{aligned}
 S &:= \{a, b, c, d, e\} \\
 \mathcal{U} &:= \left\{ \begin{array}{l} \emptyset, \\ \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \\ \{a, b\}, \{a, c\}, \{a, d\}, \{a, e\}, \{b, c\}, \{b, d\}, \{b, e\}, \{c, e\}, \{d, e\}, \\ \{a, b, c\}, \{a, b, d\}, \{a, b, e\}, \{a, c, e\}, \{a, d, e\} \end{array} \right\} \\
 \mathcal{P}(S) \setminus \mathcal{U} &= \left\{ \begin{array}{l} \{a, b, c, d, e\}, \\ \{a, b, c, d\}, \{a, b, c, e\}, \{a, b, d, e\}, \{a, c, d, e\}, \{b, c, d, e\}, \\ \{a, c, d\}, \{b, c, d\}, \{b, c, e\}, \{b, d, e\}, \{c, d, e\}, \\ \{c, d\} \end{array} \right\}
 \end{aligned}$$

Nun kann ein Beispiel für die Funktionsweise von Axiom (I3) gegeben werden. Betrachtet man die unabhängigen Mengen $A := \{a, b, c\}$ und $B := \{d\}$, so kann beispielsweise b zu B hinzugenommen werden, um eine weitere unabhängige Menge zu erhalten. Allerdings erfüllt nicht jedes Element aus A diese Eigenschaft, da $\{c, d\}$ eine abhängige Menge ist.

In dem Matroid (S, \mathcal{U}) ist $\mathcal{B} = \{\{a, b, c\}, \{a, b, d\}, \{a, b, e\}, \{a, c, e\}, \{a, d, e\}\}$ die Menge der Basen. Diese sind alle gleichmächtig, was für die Kreise nicht gilt:

$$\mathcal{C} = \{\{c, d\}, \{b, c, e\}, \{b, d, e\}\}.$$

Der Leser wird bereits bemerkt haben, dass die Menge der Kreise \mathcal{C} des Matroiden der Menge der einfachen Kreise des Graphen entspricht. Dieser Zusammenhang begründet die Namensgebung für die Menge \mathcal{C} , soll hier aber nicht weiter vertieft werden. Als Nachschlagewerk empfiehlt sich [Oxl04].

2.1.2 Äquivalente Charakterisierungen

Ein Matroid kann nicht nur durch seine unabhängigen Mengen, also \mathcal{U} , charakterisiert werden, sondern u. a. auch durch seine Kreise bzw. durch seine Basen. Für die Einführung des Dualitätsbegriffes wird die Charakterisierung durch Basen benötigt, welche nach einiger Vorarbeit in Korollar 2.13 gezeigt wird. Die Charakterisierung durch Kreise wird in dieser Arbeit nicht vollständig gezeigt, da lediglich die in der äquivalenten Charakterisierung verwendeten Eigenschaften der Menge der Kreise benötigt werden. In Kapitel 1.1 von [Oxl04], wo auch Matroide über die unabhängigen Mengen eingeführt werden, wird die vollständige Charakterisierung durch die Kreismenge vorgestellt.

Theorem 2.4

Die Menge der Kreise \mathcal{C} eines Matroiden M besitzt folgende Eigenschaften:

$$(C1) \quad \emptyset \notin \mathcal{C}$$

$$(C2) \quad C_1, C_2 \in \mathcal{C} \wedge C_1 \subseteq C_2 \quad \Rightarrow \quad C_1 = C_2$$

$$(C3) \quad C_1, C_2 \in \mathcal{C} \wedge C_1 \neq C_2 \wedge e \in C_1 \cap C_2 \quad \Rightarrow \quad \exists C_3 \in \mathcal{C} : C_3 \subseteq (C_1 \cup C_2) \setminus \{e\}$$

Beweis: (z. T. entnommen aus [Oxl04, Lemma 1.1.3])

Die Eigenschaft (C1) folgt direkt aus dem Axiom (I1) und (C2) aus der Minimalität von Kreisen. Angenommen $(C_1 \cup C_2) \setminus \{e\}$ enthält keinen Kreis, ist also unabhängig. Dann ist $(C_1 \cup C_2) \setminus \{e\} \in \mathcal{U}(M)$. Wegen $C_1 \neq C_2$ kann nach (C2) ein $f \in C_2 \setminus C_1$ gewählt werden. Da C_2 eine minimale abhängige Menge ist, folgt $C_2 \setminus \{f\} \in \mathcal{U}(M)$. Wähle nun eine maximale Teilmenge I von $C_1 \cup C_2$, die $C_2 \setminus \{f\}$ enthält und unabhängig ist. Damit ist $f \notin I$, da I sonst als Widerspruch zu (I2) einen Kreis enthalten würde. Zudem gibt es ein $g \in C_1$, das nicht in I enthalten ist, da I nicht C_1 enthalten kann. Wegen $f \in C_2 \setminus C_1$ sind g und f verschieden. Somit gilt

$$|I| \leq |(C_1 \cup C_2) \setminus \{f, g\}| = |C_1 \cup C_2| - 2 < |(C_1 \cup C_2) \setminus \{e\}|.$$

Nach Axiom (I3) kann nun I ergänzt werden, was der Maximalität von I widerspricht. \square

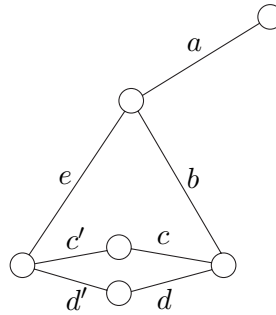


Abbildung 2.2: Beispielgraph als Grundlage eines Matroiden. Modifikation von Abbildung 2.1.

Beispiel 2.5

Um die Eigenschaft (C3) zu verdeutlichen, rekapitulieren wir den Matroiden aus Beispiel 2.3. Darin sind $C_1 := \{c, d\}$ und $C_2 := \{b, c, e\}$ zwei Kreise, wobei c in beiden Kreisen vorkommt. Nach (C3) muss es nun einen Kreis geben, der Teilmenge von $(C_1 \cup C_2) \setminus \{e\}$ ist. Dies ist genau der dritte Kreis des Matroiden, nämlich $\{b, d, e\}$.

Der auf diese Weise gewonnene Kreis kann i. A. allerdings sehr wohl eine echte Teilmenge von $(C_1 \cup C_2) \setminus \{e\}$ sein. Betrachten wir dazu eine leichte Modifikation des Beispiels 2.3, dargestellt in Abbildung 2.2. Hierfür lautet die Kreismenge

$$\mathcal{C} = \{\{c', c, d, d'\}, \{b, c, c', e\}, \{b, d, d', e\}\}.$$

Nun kann (C3) auf $C'_1 := \{c', c, d, d'\}$ und $C'_2 := \{b, c, c', e\}$, sowie die Kante c angewendet werden. Die Menge $C'_3 := \{b, d, d', e\} \subset \{b, c', d, d', e\}$ ist dann der gesuchte Kreis.

Bemerkung 2.6

Wie bereits angedeutet, genügt eine Menge \mathcal{C} , die (C1)–(C3) erfüllt, um einen Matroiden zu charakterisieren. Dies wird u. a. in [Oxl04] in Theorem 1.1.4 bewiesen.

Aus dem im vorhergehenden Theorem gewonnenen Ergebnis kann nun die folgende Aussage als technisches Hilfsmittel für weitere Beweise bzgl. der Charakterisierung durch Basen abgeleitet werden.

Lemma 2.7

Sei I eine unabhängige Menge in einem Matroiden M und $e \in S(M)$, so dass $I \cup \{e\}$ abhängig ist. Dann besitzt M einen eindeutig bestimmten Kreis in $I \cup \{e\}$ und dieser Kreis enthält e .

Beweis: ([Oxl04, Proposition 1.1.6])

Aufgrund der Abhängigkeit enthält $I \cup \{e\}$ einen Kreis und jeder Kreis in $I \cup \{e\}$ muss e enthalten, da sonst I nicht unabhängig wäre. Seien C und C' zwei verschiedene Kreise in $I \cup \{e\}$. Nach (C3) enthält $(C \cup C') \setminus \{e\}$ einen Kreis. Wegen $(C \cup C') \setminus \{e\} \subseteq I$ ist dies ein Widerspruch. Somit gilt die Gleichheit zwischen C und C' . \square

Neben der Charakterisierung durch die Kreismenge kann ein Matroid auch durch die Menge seiner Basen dargestellt werden. Um dies zu beweisen, werden die nachfolgenden Aussagen bis inklusive Lemma 2.11 benötigt, die in Kapitel 1.2 von [Oxl04] enthalten sind. Begonnen wird mit dem zum Hauptsatz über Vektorräume analogen Ergebnis, dass alle Basen gleichmächtig sind.

Lemma 2.8

Seien B_1 und B_2 zwei Basen eines Matroids M . Dann ist $|B_1| = |B_2|$.

Beweis: ([Oxl04, Lemma 1.2.1])

Angenommen $|B_1| < |B_2|$, so kann B_1 nach Axiom (I3) aus Definition 2.1 zu einer größeren unabhängigen Menge ergänzt werden, was der Maximalitätseigenschaft von Basen widerspricht. Somit gilt $|B_1| \geq |B_2|$. Analog kann $|B_1| \leq |B_2|$ gezeigt werden. \square

Das obige Lemma rechtfertigt die folgende Definition, die auch aus der Vektorraumtheorie übernommen worden ist.

Definition 2.9

Die *Mächtigkeit einer Basis* eines Matroids M nennt man den Rang von M .

Wie schon bei der Kreismenge reichen einige wenige Eigenschaften der Basismenge eines Matroiden aus, um diesen zu charakterisieren. Bei der Charakterisierung durch die Menge der Basen sind von den nachfolgend eingeführten Aussagen sogar nur (B1) und (B2) ausreichend. Die Eigenschaft (B3) stellt ein technisches Hilfsmittel dar, das bei der Einführung von dualen Matroiden genutzt wird.

Theorem 2.10

Die Menge der Basen \mathcal{B} eines Matroids M besitzt folgende Eigenschaften:

(B1) $\mathcal{B} \neq \emptyset$

(B2) $B_1, B_2 \in \mathcal{B} \wedge x \in B_1 \setminus B_2 \Rightarrow \exists y \in B_2 \setminus B_1 : (B_1 \setminus \{x\}) \cup \{y\} \in \mathcal{B}$

(B3) $B_1, B_2 \in \mathcal{B} \wedge x \in B_2 \setminus B_1 \Rightarrow \exists y \in B_1 \setminus B_2 : (B_1 \setminus \{y\}) \cup \{x\} \in \mathcal{B}$

Beweis: ([Oxl04, Lemma 1.2.2 und Lemma 2.1.2])

Die Eigenschaft (B1) folgt direkt aus (I1).

Seien B_1, B_2 und x wie in (B2) gewählt. Sowohl $B_1 \setminus \{x\}$, als auch B_2 sind nach (I2) unabhängige Mengen. Zudem ist nach Lemma 2.8 $|B_1 \setminus \{x\}| < |B_2|$. Wegen (I3) gibt es ein $y \in B_2 \setminus (B_1 \setminus \{x\})$, so dass $(B_1 \setminus \{x\}) \cup \{y\} \in \mathcal{U}(M)$. Offensichtlich ist $y \in B_2 \setminus B_1$. Da $(B_1 \setminus \{x\}) \cup \{y\}$ unabhängig ist, muss sie in einer maximalen unabhängigen Menge B'_1 enthalten sein. Nach Lemma 2.8 gilt $|B'_1| = |B_1| = |(B_1 \setminus \{x\}) \cup \{y\}|$. Somit ist $(B_1 \setminus \{x\}) \cup \{y\} = B'_1$, woraus folgt, dass $(B_1 \setminus \{x\}) \cup \{y\}$ eine Basis von M ist. Die Menge $\mathcal{B}(M)$ erfüllt also Eigenschaft (B2).

Seien B_1, B_2 und x wie in (B3) gewählt. Nach Theorem 2.7 enthält $B_1 \cup \{x\}$ einen eindeutig bestimmten Kreis C . Da C abhängig und B_2 unabhängig sind, kann ein $y \in C \setminus B_2$ gewählt werden. Offensichtlich ist $y \in B_1 \setminus B_2$. Zudem enthält $(B_1 \setminus \{y\}) \cup \{x\}$ keine Kreise und ist demnach unabhängig. Schließlich ist die Menge $(B_1 \setminus \{y\}) \cup \{x\}$ eine Basis, da sie genauso viele Elemente wie B_1 enthält. \square

Um die Aussage zu gewinnen, dass eine Menge \mathcal{B} mit den Eigenschaften (B1) und (B2) als äquivalente Charakterisierung eines Matroiden dienlich ist, muss nun noch gezeigt werden, wie mit \mathcal{B} ein Matroid definiert werden kann. Dazu wird zunächst bewiesen, dass die Gleichmächtigkeit von Basen bereits aus den Eigenschaften (B1) und (B2) folgt.

Lemma 2.11

Sei S eine Menge und \mathcal{B} eine Menge von Teilmengen von S , die (B1) und (B2) erfüllt. Dann sind die Mengen aus \mathcal{B} gleichmächtig.

Beweis: ([Oxl04, Lemma 1.2.4])

Angenommen B_1 und B_2 sind zwei verschiedene Mengen aus \mathcal{B} mit $|B_1| > |B_2|$, so dass $|B_1 \setminus B_2|$ minimal über alle Paare aus \mathcal{B} ist. Offensichtlich kann ein $x \in B_1 \setminus B_2$ gewählt werden. Nach (B2) kann zudem ein $y \in B_2 \setminus B_1$ gewählt werden, so dass $(B_1 \setminus \{x\}) \cup \{y\} \in \mathcal{B}$. Folglich ist $|(B_1 \setminus \{x\}) \cup \{y\}| = |B_1| > |B_2|$ und $|((B_1 \setminus \{x\}) \cup \{y\}) \setminus B_2| < |B_1 \setminus B_2|$, was im Widerspruch zur Wahl von B_1 und B_2 steht. \square

Mit dieser Aussage kann nun die Richtigkeit der folgenden Matroid-Konstruktion bewiesen werden.

Theorem 2.12

Sei S eine Menge und \mathcal{B} eine Menge von Teilmengen von S , die (B1) und (B2) erfüllt. Definiere \mathcal{U} wie folgt:

$$\mathcal{U} := \{X \subseteq S \mid \exists B \in \mathcal{B} : X \subseteq B\}$$

Dann ist (S, \mathcal{U}) ein Matroid.

Beweis: ([Oxl04, Theorem 1.2.3])

Die Eigenschaft (I1) folgt sofort aus (B1) und (I2) aus der Definition von \mathcal{U} .

Angenommen (I3) gilt nicht für (S, \mathcal{U}) . Dann können $I_1, I_2 \in \mathcal{U}$ mit $|I_1| < |I_2|$ gewählt werden, so dass gilt:

$$\forall e \in I_2 \setminus I_1 : I_1 \cup \{e\} \notin \mathcal{U} \quad (2.1)$$

Per Definition gibt es $B_1, B_2 \in \mathcal{B}$ mit $I_1 \subseteq B_1$, sowie $I_2 \subseteq B_2$. Dabei seien B_1 und B_2 so gewählt, dass $|B_2 \setminus (I_2 \cup B_1)|$ minimal ist. Offensichtlich gilt $I_2 \setminus B_1 \subseteq I_2 \setminus I_1$. Aufgrund von (2.1) gilt auch die umgekehrte Inklusion, woraus folgt:

$$I_2 \setminus B_1 = I_2 \setminus I_1 \quad (2.2)$$

Angenommen $B_2 \setminus (I_2 \cup B_1)$ ist nicht leer. Dann kann ein x aus dieser Menge gewählt werden. Nach (B2) gibt es ein Element $y \in B_1 \setminus B_2$, so dass $(B_2 \setminus \{x\}) \cup \{y\} \in \mathcal{B}$. Aber dann ist

$$|((B_2 \setminus \{x\}) \cup \{y\}) \setminus (I_2 \cup B_1)| < |B_2 \setminus (I_2 \cup B_1)|$$

und somit ein Widerspruch zur Wahl von B_2 . Damit ist $B_2 \setminus (I_2 \cup B_1)$ leer und somit $B_2 \setminus B_1$ in $I_2 \setminus B_1$ enthalten. Die umgekehrte Inklusion ist trivial. Mit (2.2) kann nun gefolgert werden:

$$B_2 \setminus B_1 = I_2 \setminus I_1 \quad (2.3)$$

Angenommen $B_1 \setminus (I_1 \cup B_2)$ ist nicht leer. Dann kann Eigenschaft (B2) genutzt werden, um zu einem x aus dieser Menge ein $y \in B_2 \setminus B_1$ zu finden, so dass $(B_1 \setminus \{x\}) \cup \{y\} \in \mathcal{B}$. Wegen $I_1 \cup \{y\} \subseteq (B_1 \setminus \{x\}) \cup \{y\}$ ist auch $I_1 \cup \{y\} \in \mathcal{U}$. Da $y \in B_2 \setminus B_1$, ist nach (2.3) $y \in I_2 \setminus I_1$, was der Annahme widerspricht, dass (I3) nicht gilt. Somit muss $B_1 \setminus (I_1 \cup B_2)$ leer sein und demnach ist $B_1 \setminus B_2 = I_1 \setminus B_2$. Mit $I_1 \setminus B_2 \subseteq I_1 \setminus I_2$ folgt:

$$B_1 \setminus B_2 \subseteq I_1 \setminus I_2 \quad (2.4)$$

Nach Lemma 2.11 ist $|B_1| = |B_2|$ und somit $|B_1 \setminus B_2| = |B_2 \setminus B_1|$. Mit den Gleichungen (2.4) und (2.3) ergibt sich $|I_1 \setminus I_2| \geq |I_2 \setminus I_1|$, also $|I_1| \geq |I_2|$. Dies ist schließlich ein Widerspruch, weswegen (I3) für (S, \mathcal{U}) gilt. \square

Als wichtigstes Ergebnis dieses Abschnittes lässt sich die Charakterisierung eines Matroiden durch die Menge seiner Basen festhalten.

Korollar 2.13

Sei \mathcal{B} eine Menge von Teilmengen einer Menge S . Dann ist \mathcal{B} die Menge aller Basen eines Matroiden auf S genau dann, wenn sie die Eigenschaften (B1) und (B2) erfüllt.

Beweis:

Diese Aussage folgt aus Theorem 2.10 und Theorem 2.12. □

Oft wird eine Art sehr einfacher Matroide z. B. für Spezialfälle benötigt. In diesen Matroiden sind genau die Teilmengen der Grundmenge Basen, deren Mächtigkeit gleich dem Rang des Matroiden ist. Damit ist die „Grenze“ zwischen abhängigen und unabhängigen Mengen bereits durch den Rang gegeben.

Lemma 2.14

Es seien m und n zwei natürliche Zahlen, so dass $m \leq n$. Es sei zudem S eine n -elementige Menge und \mathcal{B} die Menge aller m -elementigen Teilmengen von S . Dann ist \mathcal{B} eine Basis des Matroids $U_{m,n}$ über der Grundmenge S mit den unabhängigen Mengen

$$\mathcal{U} := \{X \subseteq S \mid |X| \leq m\}.$$

Beweis: ([Oxl04, Example 1.2.7])

(B1) ist erfüllt, da $0 \leq m \leq n$. Seien $B_1, B_2 \in \mathcal{B}$ und $x \in B_1 \setminus B_2$. Dann existiert ein $y \in B_2 \setminus B_1$. Es gilt $|(B_1 \setminus \{x\}) \cup \{y\}| = |B_1| = m$ und somit $(B_1 \setminus \{x\}) \cup \{y\} \in \mathcal{B}$. Nun kann Theorem 2.12 angewandt werden, was den Beweis abschließt. □

Definition 2.15

Der Matroid $U_{m,n} := (S, \mathcal{U})$ mit \mathcal{U} wie in Lemma 2.14 ist der Einheitsmatroid vom Rang m auf einer n -elementigen Menge.

2.1.3 Dualität

Ein sehr interessanter Aspekt der Matroidtheorie ist die Existenz einer Dualitätstheorie. Sie erweitert sowohl die Begriffe der Orthogonalität in Vektorräumen, als auch die Dualität von planaren Graphen. Ersteres wird in dem folgenden Kapitel deutlich, während auf den Zusammenhang zur Dualität von planaren Graphen hier nicht weiter eingegangen wird. Dieser Zusammenhang wird in [Oxl04] und [Wel76] diskutiert.

Der duale Matroid wird über die sogenannten Co-Basen eingeführt. In der Definition wird bereits die Verbindung zu dem MATRIXSCHWÄCHE-Problem deutlich, wenn man abhängige Mengen im dualen Matroiden betrachtet.

Definition 2.16

Der duale Matroid $M^* = (S, \mathcal{U}^*)$ des Matroids $M = (S, \mathcal{U})$ ist definiert durch die Menge der unabhängigen Mengen $\mathcal{U}^* := \{X \subseteq S : \exists \text{Basis } B \text{ von } M \text{ mit } B \cap X = \emptyset\}$.

Theorem 2.17

Der duale Matroid M^* des Matroids M ist wiederum ein Matroid und

$$\mathcal{B}^* = \{S(M) \setminus B : B \in \mathcal{B}(M)\}$$

ist die Menge seiner Basen. Diese Basen werden auch als Co-Basen von M bezeichnet.

Beweis: ([Oxl04, Theorem 2.1.1])

Da $\mathcal{B}(M)$ nicht leer ist, ist auch $\mathcal{B}^*(M)$ nicht leer. Somit wird (B1) von $\mathcal{B}^*(M)$ erfüllt.

Seien $B_1^*, B_2^* \in \mathcal{B}^*(M)$ und $x \in B_1^* \setminus B_2^*$. Weiterhin seien $B_1 = S(M) \setminus B_1^*$ und $B_2 = S(M) \setminus B_2^*$. Dann sind $B_1, B_2 \in \mathcal{B}(M)$ und $B_1^* \setminus B_2^* = B_2 \setminus B_1$. Nach (B3) und wegen $x \in B_2 \setminus B_1$ gibt es ein Element $y \in B_1 \setminus B_2$, so dass $(B_1 \setminus \{y\}) \cup \{x\} \in \mathcal{B}(M)$. Folglich ist $y \in B_2^* \setminus B_1^*$ und $S(M) \setminus ((B_1 \setminus \{y\}) \cup \{x\}) \in \mathcal{B}^*(M)$. Weiterhin gilt das Folgende:

$$S(M) \setminus ((B_1 \setminus \{y\}) \cup \{x\}) = ((S(M) \setminus B_1) \setminus \{x\}) \cup \{y\} = B_1^* \setminus \{x\} \cup \{y\}$$

Damit gilt (B2) für M^* . Schließlich kann Theorem 2.12 angewandt werden, wonach \mathcal{B}^* die Menge der Basen eines Matroiden und M^* ein Matroid sind. \square

Beispiel 2.18

Erinnern wir uns an Beispiel 2.3. Dort ist der Matroid $M = (S, \mathcal{U})$ gegeben durch:

$$S := \{a, b, c, d, e\}$$

$$\mathcal{U} := \left\{ \begin{array}{l} \emptyset, \\ \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \\ \{a, b\}, \{a, c\}, \{a, d\}, \{a, e\}, \{b, c\}, \{b, d\}, \{b, e\}, \{c, e\}, \{d, e\}, \\ \{a, b, c\}, \{a, b, d\}, \{a, b, e\}, \{a, c, e\}, \{a, d, e\} \end{array} \right\}$$

Daraus ergibt sich die Menge der Basen $\mathcal{B} = \{\{a, b, c\}, \{a, b, d\}, \{a, b, e\}, \{a, c, e\}, \{a, d, e\}\}$. Das Mengensystem der unabhängigen Mengen des dualen Matroiden ist folglich nach Definition 2.16:

$$\mathcal{U}^* := \left\{ \begin{array}{l} \emptyset, \\ \{d\}, \{e\}, \{c\}, \{b\}, \\ \{d, e\}, \{c, e\}, \{c, d\}, \{b, d\}, \{b, c\} \end{array} \right\}$$

2.1.4 Vektor-Matroide

In diesem Abschnitt konzentrieren wir uns auf Matroide, bei denen die Grundmenge durch Spalten einer Matrix gebildet wird und bei denen das Mengensystem der unabhängigen Mengen durch die Übertragung des Begriffs der linearen Unabhängigkeit entsteht.

Dieser Abschnitt gliedert sich wie folgt. Nach der Definition von Vektor-Matroiden (siehe auch Kapitel 1.1 in [Oxl04]), die auf einer Matrix basieren, wird die Gauß-Äquivalenz von Matrizen erläutert, unter der der zur Matrix korrespondierende Matroid unverändert bleibt. Diese Äquivalenz wird benötigt, um eine Standardrepräsentation des Matroiden zu erhalten, mithilfe derer anschließend gezeigt werden kann, dass die Dualität von Matroiden abgeschlossen bzgl. der Klasse der Vektor-Matroide ist (Kapitel 2.2 in [Oxl04]). Damit sind schließlich alle Grundlagen für den Beweis der Äquivalenz der Probleme MATRIXSCHWÄCHE und MINIMUM CIRCUIT gezeigt.

Theorem 2.19

Sei S die Menge von Spalten-Bezeichnungen einer $m \times n$ -Matrix A über einem Körper K , und sei \mathcal{U} die Menge der Teilmengen X von S , für die die Menge von durch X gekennzeichneten Spalten linear unabhängig in dem Vektorraum K^m ist. Dann ist (S, \mathcal{U}) ein Matroid.

Beweis: ([Oxl04, Proposition 1.1.1])

Die Axiome (I1) und (I2) folgen direkt aus den entsprechenden Eigenschaften der linearen Unabhängigkeit. Seien I_1 und I_2 zwei linear unabhängige Mengen von S mit $|I_1| < |I_2|$. Sei W der Unterraum von K^m , der von $I_1 \cup I_2$ aufgespannt wird. Dann ist $\dim(W) \geq |I_2|$. Angenommen, $I_1 \cup \{e\}$ ist linear abhängig für alle $e \in I_2 - I_1$. Dann ist W in dem von I_1 aufgespannten Raum enthalten. Also ist $|I_2| \leq \dim(W) \leq |I_1| < |I_2|$; ein Widerspruch. Demzufolge muss es ein $e \in I_2 - I_1$ geben, so dass $I_1 \cup \{e\} \in \mathcal{U}$. Also ist auch das dritte Axiom erfüllt. \square

Definition 2.20

Den Matroiden, der wie in Theorem 2.19 aus einer Matrix A gewonnen wird, bezeichnet man mit $M[A]$. Er wird auch Vektor-Matroid von A genannt.

Ein Vektor-Matroid ist zwar bis auf Isomorphie (unterschiedliche Bezeichnungen der Spalten) eindeutig, jedoch können verschiedene Matrizen denselben Matroiden ergeben, wie im Folgenden mittels der Gauß-Äquivalenz beschrieben wird. Die Gauß-Äquivalenz ist ein wichtiges Hilfsmittel in der linearen Algebra um Gleichungssysteme zu lösen [Ant04]. Der Hintergrund dabei ist, dass die Matrix vereinfacht wird, während grundlegende Eigenschaften erhalten bleiben. Hier wird u. a. gezeigt, dass die Operationen der Gauß-Äquivalenz auf Matrizen von Vektor-Matroiden angewandt werden können, ohne dass sich der Matroid ändert. Dabei ist zu beachten, dass ein Vektor-Matroid nicht durch die Vektoren an sich definiert wird, sondern über die Relation der linearen Unabhängigkeit. Somit ist es grundsätzlich möglich, dass einzelne Vektoren geändert werden, der Matroid jedoch erhalten bleibt. Mögliche Operationen dafür sind in dem folgenden Theorem aufgelistet.

Theorem 2.21

Sei A eine $m \times n$ -Matrix über dem Körper K . Dann bleibt $M[A]$ unverändert, wenn eine der folgenden Operationen auf A getätigt werden:

1. Vertauschung von zwei Zeilen.
2. Multiplikation einer Zeile mit einem Nicht-Null-Element von K .
3. Ersetzung einer Zeile durch die Summe dieser Zeile mit einer anderen.
4. Löschung einer Zeile, die nur aus Nullen besteht, sofern sie nicht die einzige Zeile ist.
5. Vertauschung von zwei Spalten (die Spaltenbezeichnungen werden auch vertauscht).

Beweis:

Es seien die Spalten von A mit e_1, \dots, e_n benannt, wobei e_i^j die j -te Komponente des

Vektors e_i bezeichne. Nach Definition ist die Menge der unabhängigen Teilmengen von $S := \{e_1, \dots, e_n\}$ gegeben durch:

$$\begin{aligned} \mathcal{U}(M[A]) &= \{X \subseteq S \mid \text{die Spalten aus } X \text{ sind linear unabhängig}\} \\ &= \{X \subseteq S \mid \forall a_x \in K : \sum_{x \in X} a_x \cdot x = 0 \Leftrightarrow \forall x \in X : a_x = 0\} \\ &= \{X \subseteq S \mid \forall a_x \in K : (\forall j \in \{1, \dots, m\} : \sum_{x \in X} a_x \cdot x^j = 0) \Leftrightarrow \forall x \in X : a_x = 0\} \end{aligned}$$

Anhand dieser Darstellung werden nun im Folgenden die einzelnen Beweise geführt. Dabei sei $X = \{x_1, \dots, x_l\} \subseteq S$ mit $1 \leq l \leq n$ und $x_i = e_{f(i)}$ mit $f : \{1, \dots, l\} \rightarrow \{1, \dots, n\}$ injektiv.

1. Da in der obigen Darstellung $\sum_{x \in X} a_x \cdot x^j = 0$ für jede Zeile j einzeln geprüft wird, hat die Vertauschung zweier Zeilen keinen Einfluss auf $\mathcal{U}(M[A])$.
2. Es sei $b \in K - \{0\}$ und $j \in \{1, \dots, m\}$. Dann gilt:

$$\sum_{x \in X} a_x \cdot x^j = 0 \Leftrightarrow \sum_{x \in X} a_x \cdot b \cdot x^j = 0$$

Folglich wird $\mathcal{U}(M[A])$ bei der Multiplikation einer Zeile mit einem Nicht-Null-Element von K nicht verändert.

3. Es seien $p, q \in \{1, \dots, m\}$ mit $p \neq q$. Dann gilt wegen $q \in \{1, \dots, p-1, p+1, \dots, m\}$:

$$\begin{aligned} &\forall j \in \{1, \dots, m\} : \sum_{x \in X} a_x \cdot x^j = 0 \\ \Leftrightarrow &\forall j \in \{1, \dots, p-1, p+1, \dots, m\} : \sum_{x \in X} a_x x^j = 0 \wedge \sum_{x \in X} a_x x^p + \sum_{x \in X} a_x x^q = 0 \\ \Leftrightarrow &\forall j \in \{1, \dots, p-1, p+1, \dots, m\} : \sum_{x \in X} a_x x^j = 0 \wedge \sum_{x \in X} a_x (x^p + x^q) = 0 \end{aligned}$$

Ergo lässt die Ersetzung einer Zeile durch die Summe dieser Zeile mit einer anderen $\mathcal{U}(M[A])$ unverändert.

4. Es sei $p \in \{1, \dots, m\}$ eine Zeile von A mit nur Null-Einträgen. Dann gilt:

$$\begin{aligned} &\forall j \in \{1, \dots, m\} : \sum_{x \in X} a_x \cdot x^j = 0 \\ \Leftrightarrow &\forall j \in \{1, \dots, p-1, p+1, \dots, m\} : \sum_{x \in X} a_x \cdot x^j = 0 \quad \wedge \quad \sum_{x \in X} a_x \cdot 0 = 0 \\ \Leftrightarrow &\forall j \in \{1, \dots, p-1, p+1, \dots, m\} : \sum_{x \in X} a_x \cdot x^j = 0 \end{aligned}$$

Demzufolge kann eine Zeile mit nur Null-Einträgen gelöscht werden, ohne dass sich $\mathcal{U}(M[A])$ verändert.

5. Es ist offensichtlich, dass die Position der Spalten für den Matroiden keine Rolle spielt.

□

Definition 2.22

Entsteht aus einer Matrix M eine Matrix M' durch Anwendung einer der Operationen 1–3 aus Theorem 2.21, so schreibt man auch $M \mapsto_G M'$. Die Gauß-Äquivalenz \equiv_G , die auf dem Raum der $n \times m$ -Matrizen über einem Körper K definiert ist, ist die transitive Hülle von \mapsto_G .

Theorem 2.23

Die Relation \equiv_G ist eine Äquivalenzrelation, d. h. es gilt:

1. $L \equiv_G L$
2. $L \equiv_G M \Rightarrow M \equiv_G L$
3. $L \equiv_G M \wedge M \equiv_G N \Rightarrow L \equiv_G N$

Beweis:

1. $L \equiv_G L$ folgt aus der Definition mit $r = 1$ und $M_1 = L$.
2. Ist $L \equiv_g M$, so existieren $L = L_1, \dots, L_r = M$ mit $L_i \mapsto_G L_{i+1}$ für alle $i = 1, \dots, r - 1$. Dann gilt auch $L_{i+1} \mapsto_G L_i$ für alle $i = r - 1, \dots, 1$, da:
 - (a) Die Vertauschung zweier Zeilen ist symmetrisch.
 - (b) Die Multiplikation mit einem Nicht-Null-Element kann mit dem dazu multiplikativ inversen Element rückgängig gemacht werden.
 - (c) Wurde eine Zeile durch die Summe dieser Zeile mit einer anderen ersetzt, so kann von der neuen Zeile die zweite Zeile abgezogen und die neue durch das Resultat ersetzt werden, um den Ausgangszustand zu erhalten.
3. Ist $L \equiv_G M$ und $M \equiv_G N$, dann gilt $L = L_1, \dots, L_r = M$ mit $L_i \mapsto_G L_{i+1}$ für alle $i = 1, \dots, r$ und $M = M_1, \dots, M_s = N$ mit $M_i \mapsto_G M_{i+1}$ für alle $i = 1, \dots, s$. Daraus kann eine neue Transformationsfolge $L = L_1, \dots, L_r = M = M_1, \dots, M_s = N$ gebildet werden, weswegen gilt: $L \equiv_G N$.

□

Analog zum bekannten Gauß-/Jordan-Verfahren (siehe z. B. [Ant04]) kann mithilfe der Operationen aus Theorem 2.21 die erzeugende Matrix eines Vektor-Matroids in eine Form überführt werden, die Standardrepräsentation genannt wird.

Definition 2.24

Eine $m \times n$ -Matrix $S = (s_{i,j})$ heißt Stufenmatrix mit der Stufenzahl r , falls entweder $r = 0$ und $S = 0$ sind oder $r \geq 1$ und es r Zahlen j_1, \dots, j_r gibt mit $1 \leq j_1 < \dots < j_r \leq n$ derart, dass gilt:

1. $s_{1,j_1} \neq 0, \dots, s_{r,j_r} \neq 0$
2. $s_{i,j} = 0$ für $i = 1, \dots, r$ und $j < j_i$
3. $s_{i,j} = 0$ für $i \geq r + 1$ und $j = 1, \dots, n$

Gilt außerdem noch

$$4. s_{1,j_1} = \dots = s_{r,j_r} = 1 \text{ und } s_{k,j_i} = 0 \text{ für } k < i \text{ und } i = 1, \dots, r$$

so heißt S normierte Stufenmatrix. Das r -Tupel (j_1, \dots, j_r) wird Typ der Stufenmatrix S genannt.

Bemerkung 2.25

Eine Stufenmatrix besitzt die folgende Form, wobei jedes $*$ für eine beliebige Zahl und \times für eine beliebige Zahl ungleich Null steht:

$$\begin{pmatrix} 0 & \cdots & 0 & \times & * & \dots & * \\ 0 & \dots & \dots & 0 & \times & * & \dots & * \\ \vdots & & & & & \ddots & & \\ 0 & \dots & \dots & \dots & 0 & \times & * & \dots & * \\ & & & & & & & & 0 \end{pmatrix}$$

Dabei befindet sich die erste Zahl ungleich Null (\times) in der i -ten Zeile an der Position j_i . In einer normierten Stufenmatrix sind zudem diese ersten Nicht-Null-Einträge gleich 1.

Theorem 2.26

Zu jeder $m \times n$ -Matrix M gibt es eine normierte $m \times n$ -Stufenmatrix S mit $M \equiv_G S$.

Beweis:

Der Beweis wird als vollständige Induktion über der Anzahl n der Spalten geführt.

(IA) Es sei $n = 1$, also $M = (\alpha_1, \dots, \alpha_m)^T$. Dann ist $M = (0, \dots, 0)^T$ oder $M \equiv_G (1, 0, \dots, 0)^T$. Letzteres kann erreicht werden, indem eine Nicht-Null-Zeile (α_i) auf Position 1 getauscht und anschließend per Multiplikation mit $1/\alpha_i$ normiert wird. Alle weiteren Nicht-Null-Zeilen (α_j) werden ersetzt durch $-\alpha_j \cdot (1) + (\alpha_j)$.

(IS) Es sei $n \geq 2$ und $M = (M', \vec{b})$. Also $M' \equiv_G S'$ mit S' als normierte Stufenmatrix vom Typ (j_1, \dots, j_r) . Folglich ist $M = (M', \vec{b}) \equiv_G (S', \vec{c})$ mit $c = (c_1, \dots, c_m)^T$, wobei dieselbe Folge von Gauß-Operationen angewandt wird, die auch M' in S' überführen. Man erhält eine Matrix der Form:

$$\begin{pmatrix} 0 & \cdots & 0 & 1 & * & \cdots & * & 0 & * & \cdots & * & 0 & * & \cdots & * & \left| \begin{matrix} c_1 \\ c_2 \\ \vdots \\ c_r \\ \vdots \\ c_m \end{matrix} \right. \\ 0 & \dots & \dots & 0 & 1 & * & \cdots & * & 0 & * & \cdots & * & 0 & * & \cdots & * \\ \vdots & & & & & & & \ddots & & & & & & & & \\ 0 & \dots & \dots & \dots & 0 & 1 & * & \cdots & * & & & & & & & \\ & & & & & & & & & & & & & & & 0 \end{pmatrix}$$

Fall 1: $c_{r+1} = \dots = c_m = 0$
 Es folgt, dass (S', \vec{c}) bereits eine normierte Stufenmatrix ist.

Fall 2: $c_i \neq 0$ für $i \in \{r+1, \dots, m\}$

Hierbei kann die Zeile i mit der Zeile $r+1$ getauscht werden, sofern $i \neq r+1$ ist. Danach wird diese Zeile normiert und alle anderen Zeilen, wie im Induktionsanfang durch Summen ersetzt, um $M \equiv_G (S', \vec{c}) \equiv_G (S', \vec{e})$ mit $\vec{e} = (0, \dots, 0, 1, 0, \dots, 0)^T$ zu erhalten. Diese Operationen haben keinen Einfluss auf S' , da alle Elemente bis auf das letzte jeder Zeile 0 sind. \square

Mit diesem Wissen lässt sich nun die Standardrepräsentation eines Vektor-Matroiden einführen. Dafür wird die Bezeichnung $[A|B]$ für eine $m \times p$ -Matrix A mit den Einträgen $a_{i,j}$ und eine $m \times q$ -Matrix B mit den Einträgen $b_{i,j}$ verwendet, die die folgende Matrix darstellt:

$$[A|B] := \begin{pmatrix} a_{1,1} & \cdots & a_{1,p} & b_{1,1} & \cdots & b_{1,q} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,p} & b_{m,1} & \cdots & b_{m,q} \end{pmatrix}$$

Korollar 2.27

Eine $m \times n$ -Matrix A über einem Körper K kann mit den Operationen aus Theorem 2.21 in eine Matrix der Form $[I_r|D]$ überführt werden, wobei I_r die r -dimensionale Einheitsmatrix und D eine $r \times (n-r)$ -Matrix ist. Dabei bleibt der Vektor-Matroid $M[A]$ erhalten. Die Form $[I_r|D]$ wird Standardrepräsentation eines Vektor-Matroiden genannt.

Beweis:

Nach Theorem 2.26 kann A mit den Operationen 1–3 in eine Stufenmatrix S vom Rang r überführt werden. Aus dieser Stufenmatrix werden mit Operation 4 alle Null-Zeilen $r+1, \dots, m$ gelöscht und anschließend mit Operation 5 die Spalten so vertauscht, dass die resultierende Stufenmatrix den Typ $(1, \dots, r)$ besitzt. \square

Aus der Standardrepräsentation eines Vektor-Matroiden kann mit geringem Aufwand eine erzeugende Matrix des dualen Matroiden gewonnen werden, wie das folgende Theorem zeigt.

Theorem 2.28

Ist M der Vektor-Matroid von $[I_r|D]$, so ist der dazu duale Matroid M^* der Vektor-Matroid von $[-D^T|I_{n-r}]$.

Beweis: ([Oxl04, Theorem 2.2.8])

Es sei B eine Basis von M . Zunächst wird gezeigt, dass $S(M) \setminus B$ eine Basis von $[-D^T|I_{n-r}]$ ist. Im Folgenden werden Spalten und Zeilen von D neu angeordnet. Dies wirkt sich auf $-D^T$ nur in Form einer entsprechenden Neuordnung von Zeilen und Spalten aus.

Durch eine solche Umordnung kann angenommen werden, dass es ein $t \in \{0, \dots, r\}$ gibt, so dass $B = \{e_{r-t+1}, \dots, e_r, e_{r+1}, \dots, e_{2r-t}\}$ ist. Damit kann $[I_r|D]$ wie folgt in Blöcke unterteilt werden:

$$\left[\begin{array}{c|c|c|c} e_1 \cdots e_{r-t} & e_{r-t+1} \cdots e_r & e_{r+1} \cdots e_{2r-t} & e_{2r-t+1} \cdots e_n \\ \hline I_{r-t} & 0 & D_1 & D_2 \\ \hline 0 & I_t & D_3 & D_4 \end{array} \right]$$

Die Matrix

$$\left[\begin{array}{c|c} 0 & D_1 \\ \hline I_t & D_3 \end{array} \right]$$

ist vom Rang r , da B eine Basis ist. Daher besitzt D_1 und somit auch $-D_1^T$ den Rang $r - t$. Die Partition von $[-D^T|I_{n-r}]$ analog zur Partition von $[I_r|D]$ sieht wie folgt aus:

$$\left[\begin{array}{c|c|c|c} e_1 \cdots e_{r-t} & e_{r-t+1} \cdots e_r & e_{r+1} \cdots e_{2r-t} & e_{2r-t+1} \cdots e_n \\ \hline -D_1^T & -D_3^T & I_{r-t} & 0 \\ \hline -D_2^T & -D_4^T & 0 & I_{n-(2r-t)} \end{array} \right]$$

Die zu $S(M) \setminus B$ korrespondierende Submatrix ist

$$\left[\begin{array}{c|c} -D_1^T & 0 \\ \hline -D_2^T & I_{n-(2r-t)} \end{array} \right]$$

und ihr Rang ist gleich der Summe der Ränge von $I_{n-(2r-t)}$ und $-D_1^T$, also $n - r$. Somit ist $S(M) \setminus B$ eine Basis des Vektor-Matroids von $[-D^T|I_{n-r}]$.

Diese Schlussfolgerungen lassen sich auch umkehren, weswegen jede Basis dieses Matroiden auf diese Weise gewonnen werden kann. Somit ist $[-D^T|I_{n-r}]$ eine Repräsentation für M^* .

Zu beachten bleibt noch, dass bisher der Rang r weder 0 noch n sein durfte. Ist $r = 0$, so ist $M \cong U_{0,m}$, also $M^* \cong U_{n,n} \cong M[I_n]$. Ist $r = n$, so ist $M \cong U_{n,n}$, also $M^* \cong U_{0,n} \cong M[0_n]$ wobei 0_n die $n \times n$ -Null-Matrix ist. \square

Aus den oben gewonnenen Erkenntnissen lässt sich ein polynomialer Algorithmus ableiten, der aus einer gegebenen Matrix A die Matrix des zu $M[A]$ dualen Matroiden berechnet. Damit lässt sich sowohl das Problem der MATRIXSCHWÄCHE auf das Problem MINIMUM CIRCUIT reduzieren, als auch umgekehrt.

Theorem 2.29

Das Problem MATRIXSCHWÄCHE ist äquivalent zum Problem MINIMUM CIRCUIT. Die Matrix des einen Problems kann in Polynomialzeit in eine korrespondierende Matrix des anderen Problems transformiert werden.

Beweis:

Da in einem Vektor-Matroiden die Spalten der zugrunde liegenden Matrix die Objekte von Interesse sind, betrachten wir zunächst die Sensitivitätsmatrix M in ihrer transponierten Form $A := M^T$. Nun lässt sich äquivalent zu MATRIXSCHWÄCHE fragen: Gibt es k Elemente aus $S(M[A])$, so dass nach deren Löschen der resultierende Matroid nicht mehr den Rang n hat? Diese Äquivalenz gilt aufgrund der Definition von Vektor-Matroiden, siehe Theorem 2.19.

Wenn nun k solche Elemente vorliegen — z. B. als Menge L —, so dass sich der Rang des Matroiden $M[A]$ verringert, dann muss jede Basis von $M[A]$ einen nicht-leeren Durchschnitt mit L besitzen, da ansonsten der Rang nicht fällt. Gibt es andererseits eine Menge L , die mit jeder Basis einen nicht-leeren Durchschnitt besitzt, so können die darin

enthaltenen Elemente gelöscht werden, um den Rang des Matroiden zu verringern. Somit ist es äquivalent zu fragen, ob es eine k -elementige Teilmenge L von $S(M[A])$ gibt, so dass für alle $B \in \mathcal{B}(M[A])$ gilt: $B \cap L \neq \emptyset$.

Eine solche Menge L ist nun jedoch nach Definition 2.16 genau eine k -elementige, abhängige Menge in dem zu $M[A]$ dualen Matroiden. Nach Theorem 2.28 ist der zu einem Vektor-Matroiden duale Matroid wiederum ein Vektor-Matroid. Sei A^* die zugrunde liegende Matrix dieses dualen Vektor-Matroiden, dann ist MATRIXSCHWÄCHE äquivalent zu MINIMUM CIRCUIT mit der Matrix A^* .

Schließlich bleibt noch die Betrachtung der Laufzeit der verwendeten Matrixoperationen. Die Transposition einer Matrix kann in polynomialer Zeit durchgeführt werden. Anhand der Beweise zu Theorem 2.26 und Korollar 2.27 kann ein entsprechender Algorithmus gewonnen werden, der die Standardrepräsentation einer Matrix berechnet, womit sich nach Theorem 2.28 die Matrix des dualen Vektor-Matroiden aufstellen lässt. Damit sind alle verwendeten Operationen in polynomialer Zeit ausführbar. \square

Mit der Äquivalenz zu dem bekannten Problem MINIMUM CIRCUIT lassen sich nun weitere Untersuchungen zur Komplexität von MATRIXROBUSTHEIT auf dieses Problem zurückführen. Der folgende Abschnitt zeigt z. B. auf der Basis von MINIMUM CIRCUIT die Äquivalenz zu einem weiteren Problem, mithilfe derer schließlich die NP -Vollständigkeit von MATRIXROBUSTHEIT gezeigt wird. Weiterhin basieren die Betrachtungen in Abschnitt 4.3 auf MINIMUM CIRCUIT.

2.2 Codierungstheorie

Im letzten Abschnitt wurde die Äquivalenz der Probleme MATRIXSCHWÄCHE und MINIMUM CIRCUIT gezeigt. Diese Äquivalenz wird in diesem Abschnitt auf MINIMUM DISTANCE aus der Codierungstheorie erweitert, um komplexitätstheoretische Ergebnisse zu letzterem auf MATRIXROBUSTHEIT anwenden zu können.

Codes werden hauptsächlich genutzt, um Daten über einen verrauschten Kanal zu übermitteln. Dazu werden diese Daten codiert, indem Redundanz hinzugefügt wird, so dass selbst bei einigen Fehlern durch die Übertragung die ursprüngliche Nachricht wiederhergestellt werden kann. Ein Code ist eine Menge von Codewörtern, durch die die Nachricht dargestellt wird. Im Allgemeinen ist ein Code eine beliebige Menge. Die Betrachtungen in dieser Arbeit konzentrieren sich jedoch auf lineare Codes, welche einen Unterraum eines Vektorraums darstellen.

Da eine Codierung meist mit Computern realisiert wird, betrachtet man hauptsächlich lineare Codes über endlichen, im Speziellen binären Körpern. Um eine Äquivalenz zu MATRIXROBUSTHEIT zu gewinnen wird der hier vorgestellte Teil der Codierungstheorie, welcher aus [LX04] entnommen wurde, auf beliebige Körper verallgemeinert.

Definition 2.30

Ein linearer Code C der Länge n über einem Körper K ist ein Unterraum von K^n . Die Elemente von C werden Codewörter genannt.

Die oben erwähnte Redundanz in Codes wird mittels des sogenannten Hamming-Abstands bzw. mit dem Hamming-Gewicht gemessen.

Definition 2.31

Es sei K ein Körper und $x, y \in K^n$, sowie C ein Code über K .

1. Der Hamming-Abstand von x und y ist definiert als:

$$d(x, y) := d(x_1, y_1) + \cdots + d(x_n, y_n) \quad \text{mit} \quad d(x_i, y_i) := \begin{cases} 1 & \text{falls } x_i \neq y_i \\ 0 & \text{falls } x_i = y_i \end{cases}$$

2. Der minimale Codeabstand von C ist der minimale Abstand zwischen allen Codewörtern aus C :

$$d(C) := \min\{d(x, y) \mid x, y \in C \wedge x \neq y\}$$

3. Das Hamming-Gewicht von x ist der Hammingabstand von x zum Null-Codewort:

$$wt(x) := d(x, 0)$$

4. Analog zum minimalen Codeabstand wird das minimale Hamming-Gewicht für einen Code C definiert:

$$wt(C) := \min\{wt(x) \mid x \in C \setminus \{0\}\}$$

Beispiel 2.32

Gegeben ist der Code C , der durch die Vektoren $a := (0, 1, 0, 2)$ und $b := (5, 1, 0, 3)$ in dem Vektorraum \mathbb{Z}^4 aufgespannt wird. Dabei sind beispielsweise

$$\begin{aligned} d(a, b) &= 1 + 0 + 0 + 1 = 2 \quad \text{und} \\ wt(b) = d(b, 0) &= 1 + 1 + 0 + 1 = 3. \end{aligned}$$

Um den minimalen Codeabstand bzw. das minimale Hamming-Gewicht zu bestimmen untersuchen wir einige Eigenschaften von C . Dazu seien im Folgenden mit $*$ ein beliebiger Wert und mit \times ein Nicht-Null-Wert bezeichnet. Da a und b eine Basis für C darstellen, können alle gültigen Codewörter als Linearkombination von a und b dargestellt werden. Damit lassen sich drei Eigenschaften festhalten:

1. Codewörter der Struktur $(0, *, *, *)$ sind stets ein Vielfaches von a . Die entstehenden Wörter sind somit entweder das Null-Wort oder von der Form $(0, \times, 0, \times)$. Folgende Arten von Codewörter sind also nicht möglich: $(0, 0, 0, \times)$, $(0, 0, \times, 0)$, $(0, 0, \times, \times)$, $(0, \times, 0, 0)$, $(0, \times, \times, 0)$ und $(0, \times, \times, \times)$.
2. Codewörter der Struktur $(*, 0, *, 0)$ sind bis auf das Null-Wort nicht möglich. Dies sind im Speziellen Codewörter der Arten $(0, 0, \times, 0)$, $(\times, 0, 0, 0)$ und $(\times, 0, \times, 0)$.
3. Da die dritte Komponente eines Codewortes stets Null ist, gibt es keine Wörter der Formen $(0, 0, \times, 0)$, $(0, 0, \times, \times)$, $(0, \times, \times, 0)$, $(0, \times, \times, \times)$, $(\times, 0, \times, 0)$, $(\times, 0, \times, \times)$, $(\times, \times, \times, 0)$, sowie $(\times, \times, \times, \times)$.

Somit sind nur Codewörter der folgenden Arten in C enthalten:

$$\begin{aligned} (0, 0, 0, 0) & \text{ erzeugt durch } 0 \cdot a + 0 \cdot b \\ (0, \times, 0, \times) & \text{ erzeugt durch } x \cdot a + 0 \cdot b, \text{ mit } x \neq 0 \\ (\times, 0, 0, \times) & \text{ erzeugt durch } x \cdot c, \text{ mit } x \neq 0 \text{ und } c := b - a = (5, 0, 0, 1) \\ (\times, \times, 0, 0) & \text{ erzeugt durch } x \cdot d, \text{ mit } x \neq 0 \text{ und } d := 2b - 3a = (10, -1, 0, 0) \\ (\times, \times, 0, \times) & \text{ z. B. } a + b = (5, 2, 0, 5) \end{aligned}$$

Offensichtlich ist das minimale Hamming-Gewicht $wt(C) = 2$ und der minimale Codeabstand höchstens 2. Der minimale Codeabstand könnte gleich 1 sein, wenn es ein Codewort x des letzten Typs und ein Codewort y vom Typ $(0, \times, 0, \times)$, $(\times, 0, 0, \times)$ oder $(\times, \times, 0, 0)$ gäbe, die in den beiden Komponenten übereinstimmen, die in y nicht Null sind. Dies ist aber nicht möglich, da die Linearkombination $x = p \cdot a + q \cdot b$ durch die beiden Nicht-Null-Komponenten von y bestimmt ist und damit $y = x$.

Die Übereinstimmung des minimalen Codeabstands und des minimalen Hamming-Gewichts ist allerdings kein Zufall, wie in Abschnitt 2.2.1 gezeigt wird.

Mithilfe der obigen Definitionen kann nun eine im Sinne von beliebigen Grundkörpern verallgemeinerte Version von MINIMUM DISTANCE definiert werden.

Problem: GENERALIZED MINIMUM DISTANCE

Instanz: Eine $m \times n$ -Matrix M über einem beliebigen Körper K und eine positive, ganze Zahl ω .

Frage: Gibt es einen Vektor $x \in K^n$ ungleich dem Nullvektor mit $wt(x) \leq \omega$, so dass $Mx = 0$?

Ein einfacher Vergleich mit MINIMUM CIRCUIT offenbart die Äquivalenz beider Probleme.

Problem: MINIMUM CIRCUIT

Instanz: Eine $m \times n$ -Matrix M über einem beliebigen Körper K und eine positive ganze Zahl k .

Frage: Gibt es eine k -elementige, linear abhängige Menge von Spaltenvektoren in M ?

Theorem 2.33

Die Probleme MINIMUM CIRCUIT und GENERALIZED MINIMUM DISTANCE sind äquivalent.

Beweis:

Es sei $M = (m_1^T, \dots, m_n^T)$, $x = (x^1, \dots, x^n)$ und $\omega = k > 0$, dann gilt:

$$\begin{aligned} \exists x \neq 0 : \quad & wt(x) \leq \omega \wedge Mx = 0 \\ \Leftrightarrow \exists x \neq 0 : \quad & wt(x) \leq \omega \wedge x^1 \cdot m_1 + \dots + x^n \cdot m_n = 0 \\ \Leftrightarrow \exists \{x_{i_1}, \dots, x_{i_\omega}\} \subseteq K \setminus \{0\} : \quad & x_{i_1} \cdot m_{i_1} + \dots + x_{i_\omega} \cdot m_{i_\omega} = 0 \\ \Leftrightarrow \exists S \subseteq S(M) \text{ mit } |S| \leq \omega : S \text{ ist linear abhängig} \\ \Leftrightarrow \exists S \subseteq S(M) \text{ mit } |S| = \omega : S \text{ ist linear abhängig} \end{aligned}$$

□

Mit diesem Ergebnis lassen sich komplexitätstheoretische Betrachtungen zu MINIMUM DISTANCE auf MATRIXROBUSTHEIT übertragen, wie in Abschnitt 2.3 gezeigt wird. Der restliche Teil dieses Abschnittes offenbart die Herkunft des Namens Minimum Distance und dabei die Äquivalenz zu einem weiteren Problem.

2.2.1 Codeabstände

Wie bereits erwähnt, wird die Redundanz eines Codes mit dem Hamming-Gewicht bzw. dem Hamming-Abstand gemessen. Der Zusammenhang dieser beiden Maße wird im Folgenden vorgestellt. Dabei werden Beweise aus [LX04] auf Codes über beliebigen Körpern erweitert.

Während sich das Hamming-Gewicht besser für Algorithmen und komplexitätstheoretische Untersuchungen eignet, da es eine Funktion mit nur einem Argument ist, besitzt der Hamming-Abstand, die Eigenschaften einer *Metrik*, d. h. es gilt:

1. $d(x, y) = 0 \Leftrightarrow x = y$
2. $d(x, y) = d(y, x)$
3. $d(x, y) + d(y, z) \geq d(x, z)$

Lemma 2.34

Der Hamming-Abstand d über Codewörtern des Codes C über K^n ist eine Metrik von K^n . Außerdem gilt für $x, y \in K^n$: $0 \leq d(x, y) \leq n$.

Beweis: ([LX04, Proposition 2.3.3])

Seien im Folgenden $x, y, z \in K^n$. Dann gilt:

$$d(x, y) = 0 \Leftrightarrow d(x_1, y_1) + \dots + d(x_n, y_n) = 0 \Leftrightarrow x_1 = y_1 \wedge \dots \wedge x_n = y_n \Leftrightarrow x = y$$

$$d(x, y) = d(x_1, y_1) + \dots + d(x_n, y_n) = d(y_1, x_1) + \dots + d(y_n, x_n) = d(y, x)$$

$$\begin{aligned} d(x_i, y_i) + d(y_i, z_i) &= \begin{cases} 0 & \text{falls } x_i = y_i = z_i \\ 1 & \text{falls } (x_i = y_i \wedge y_i \neq z_i) \vee (x_i \neq y_i \wedge y_i = z_i) \\ 2 & \text{falls } x_i \neq y_i \wedge y_i \neq z_i \end{cases} \\ &\geq \begin{cases} 0 & \text{falls } x_i = z_i \text{ (} x_i = y_i = z_i \text{ oder } x_i \neq y_i \wedge y_i \neq z_i) \\ 1 & \text{falls } x_i \neq z_i \text{ (möglich, wenn nicht } x_i = y_i = z_i) \end{cases} \\ &= d(x_i, z_i) \end{aligned}$$

□

Im Folgenden wird die Gleichheit des minimalen Codeabstands und des minimalen Hamming-Gewichts gezeigt. Diese Übereinstimmung begründet die Verwendung des Hamming-Gewichts in der Problemstellung GENERALIZED MINIMUM DISTANCE.

Lemma 2.35

Sei K ein Körper und $x, y \in K^n$. Dann gilt $d(x, y) = wt(x - y)$.

Beweis: ([LX04, Lemma 4.3.3] auf beliebige Grundkörper verallgemeinert)

Die Behauptung ergibt sich aus folgender Aussage:

$$d(x_i, y_i) = \begin{Bmatrix} 1 & \text{falls } x_i \neq y_i \\ 0 & \text{falls } x_i = y_i \end{Bmatrix} = \begin{Bmatrix} 1 & \text{falls } x_i - y_i \neq 0 \\ 0 & \text{falls } x_i - y_i = 0 \end{Bmatrix} = d(x_i - y_i, 0)$$

□

Theorem 2.36

Für einen Code C über einem Körper K gilt $d(C) = wt(C)$.

Beweis: ([LX04, Theorem 4.3.8] auf beliebige Grundkörper verallgemeinert)

Es seien $x, y \in K$. Nach Lemma 2.35 gilt $d(x, y) = wt(x - y)$. Weiterhin existieren $x', y' \in K$ mit $d(x', y') = d(C)$ und $z \in C - \{0\}$ mit $wt(C) = wt(z)$. Damit folgt:

$$d(C) = d(x', y') = wt(x' - y') \geq wt(C) \quad (\text{da } x' - y' \in C)$$

$$wt(C) = wt(z) = d(z, 0) \geq d(C) \quad (\text{da } z, 0 \in C)$$

□

2.2.2 Duale Codes, Generator- und Paritätskontrollmatrizen

Lineare Codes sind Unterräume eines Vektorraums, die von Basen aufgespannt werden. Daher können solche Basen zusammengefasst als Matrizen analog zur Matroidtheorie (vgl. Vektor-Matroiden, Theorem 2.19) verwendet werden, um einen linearen Code zu definieren.

Definition 2.37

Eine Generatormatrix für einen linearen Code C über einem Körper K ist eine Matrix G aus $K^{m \times n}$, deren Zeilen eine Basis für C bilden.

Mittels Generatormatrizen kann nun das Problem GENERALIZED MINIMUM DISTANCE* definiert werden, welches die Herkunft des Namens Minimum Distance verdeutlicht.

Problem: GENERALIZED MINIMUM DISTANCE*

Instanz: Eine Generatormatrix G über einem beliebigen Körper K und eine positive, ganze Zahl d .

Frage: Besitzt der durch seine Generatormatrix G definierte Code C einen minimalen Codeabstand $d(C) \leq d$?

Im Folgenden wird unter Verwendung von auf beliebige Körper verallgemeinerten Aussagen aus [LX04] die Äquivalenz der Probleme GENERALIZED MINIMUM DISTANCE* und GENERALIZED MINIMUM DISTANCE gezeigt. Die Matrizen beider Probleme werden dabei auf analoge Art und Weise ineinander transformiert, wie dies bei MATRIXSCHWÄCHE und MINIMUM CIRCUIT der Fall ist.

Dafür wird wie auch in der Matroidtheorie ein Dualitätsbegriff benutzt. Dabei wird das zu einem linearen Code C , der einen Unterraum eines Vektorraums K^n darstellt, orthogonale Komplement betrachtet, welches wiederum ein Unterraum von K^n und somit auch ein linearer Code ist. Dieser Code wird als der duale Code von C bezeichnet.

Definition 2.38

Es sei K ein Körper und $S \subseteq K^n$. Das orthogonale Komplement von S ist definiert als

$$S^\perp := \{x \in K^n \mid \forall s \in S : \langle x, s \rangle = 0\}.$$

Mit $\langle \cdot, \cdot \rangle$ sei sowohl hier als auch im Folgenden das euklidische Skalarprodukt bezeichnet.

Bemerkung 2.39

Ist $[\cdot]$ der lineare Hüllenoperator, so ist für einen Körper K und $S \subseteq K^n$ das orthogonale Komplement orthogonal auf dem gesamten Unterraum $[S]$, d. h. $\forall s \in S \forall s' \in S^\perp : \langle s, s' \rangle = 0$. Außerdem gilt:

$$\dim([S]) + \dim(S^\perp) = n$$

Ist K ein Körper, so gilt für einen Unterraum bzw. linearen Code C über K die Gleichung $C = [C]$.

Definition 2.40

Sei C ein linearer Code über einem Körper K . Der duale Code von C ist das orthogonale Komplement von C , also C^\perp .

Beispiel 2.41

Betrachten wir erneut den linearen Code, der durch die Vektoren $a := (0, 1, 0, 2)$ und $b := (5, 1, 0, 3)$ in dem Vektorraum \mathbb{Z}^4 aufgespannt wird. Dann spannen beispielsweise die Vektoren $c := (0, 0, 0, 1)$ und $d := (-1, -10, 0, 5)$ den dualen Code auf, da c und d linear unabhängig sind und beide senkrecht zu sowohl a als auch b stehen.

Da der duale Code wiederum ein Code ist, kann auch für einen solchen eine Generatormatrix gefunden werden, welche dann Paritätskontrollmatrix des ursprünglichen Codes genannt wird.

Definition 2.42

Eine Paritätskontrollmatrix H für einen linearen Code C über einem Körper K ist eine Generatormatrix für den dualen Code C^\perp .

Paritätskontrollmatrizen werden oft benutzt, um zu verifizieren, dass ein Vektor ein Codewort ist. Ist H eine Paritätskontrollmatrix und x ein mögliches Codewort, so kann

$$H \cdot x = 0?$$

für diesen Test verwendet werden, wie das folgende Lemma u. a. belegt.

Lemma 2.43

Sei C ein linearer Code über einem Körper K mit der $k \times n$ -Generatormatrix G . Dann ist $v \in K^n$ ein Element von C^\perp genau dann, wenn v orthogonal zu jeder Zeile von G ist, d. h.

$$v \in C^\perp \Leftrightarrow vG^T = 0.$$

Insbesondere ist eine $(n - k) \times n$ -Matrix H eine Paritätskontrollmatrix für C genau dann, wenn die Zeilen von H linear unabhängig sind und $HG^T = 0$ gilt.

Beweis: ([LX04, Lemma 4.5.4] auf beliebige Grundkörper verallgemeinert)

Es bezeichne r_i die i -te Zeile von G . Somit kann jedes $c \in C$ als Linearkombination der r_i geschrieben werden:

$$c = \lambda_1 r_1 + \dots + \lambda_k r_k \quad \text{mit } \lambda_1, \dots, \lambda_k \in K$$

Es sei $v \in C^\perp$. Dann gilt $\langle v, c \rangle = 0$ für alle $c \in C$, insbesondere für alle $r_i, i = 1, \dots, n$, also $vG^T = 0$. Wenn umgekehrt $\langle v, r_i \rangle = 0$ für alle $r_i, i = 1, \dots, n$ ist, so folgt für ein beliebiges $c \in C$ mit $c = \lambda_1 r_1 + \dots + \lambda_k r_k$

$$\langle v, c \rangle = \lambda_1 \langle v, r_1 \rangle + \dots + \lambda_k \langle v, r_k \rangle = 0$$

aufgrund der Bilinearität von $\langle \cdot, \cdot \rangle$.

Für den Beweis der zweiten Aussage sei zunächst H eine Paritätskontrollmatrix für C . Dann sind die Zeilen von H per Definition linear unabhängig. Da die Zeilen von H Codewörter von C^\perp sind, folgt nach der ersten Aussage, dass $HG^T = 0$ ist.

Gilt $HG^T = 0$, so zeigt die erste Aussage, dass die Zeilen von H und damit auch der Zeilenraum von H in C^\perp enthalten sind. Sind die Zeilen von H linear abhängig, so hat der Zeilenraum die Dimension $n - k$, also ist der Zeilenraum von H gleich C^\perp . Mit anderen Worten ist H eine Paritätskontrollmatrix von C . \square

Bemerkung 2.44

Eine äquivalente Formulierung zu Lemma 2.43 ist die folgende:

Sei C ein linearer Code über einem Körper K mit der $(n-k) \times n$ -Paritätskontrollmatrix H . Dann ist $v \in K^n$ ein Element von C genau dann, wenn v orthogonal zu jeder Zeile von H ist, d. h.:

$$v \in C \Leftrightarrow vH^T = 0$$

Insbesondere ist eine $k \times n$ -Matrix G eine Generatormatrix für C genau dann, wenn die Zeilen von G linear unabhängig sind und $GH^T = 0$ gilt.

Analog zu Vektor-Matroiden (vgl. Theorem 2.21 bis Korollar 2.27) werden auch für die Generator- und Paritätskontrollmatrizen Standardrepräsentationen eingeführt. Dazu wird zunächst bewiesen, dass sich die Codes u. a. unter Anwendung der Gauß-Äquivalenz auf die zugrunde liegenden Matrizen nicht ändern.

Theorem 2.45

Sei G eine Generatormatrix für einen Code C über dem Körper K . Dann bleibt C unverändert, wenn eine der folgenden Operationen auf G getätigt wird:

1. Vertauschung von zwei Zeilen.
2. Multiplikation einer Zeile mit einem Nicht-Null-Element von K .
3. Ersetzung einer Zeile durch die Summe dieser Zeile mit einer anderen.
4. Vertauschung von zwei Spalten.

Beweis:

Es ist allgemein bekannt, dass die Operationen 1–3 die Basis-Eigenschaften nicht verändern. Operation 4 dagegen verändert zwar den Code, allerdings ist das Resultat ein zum ursprünglichen Code isomorpher Code, d. h. Eigenschaften wie der minimale Codeabstand sind identisch. \square

Korollar 2.46

Eine $k \times n$ -Generatormatrix G über einem Körper K kann mit den Operationen aus Theorem 2.45 in eine Generatormatrix der Form $[I_k|D]$ überführt werden, wobei D eine $k \times (n - k)$ -Matrix ist. Dabei bleibt der Code C (bis auf Isomorphie) erhalten.

Beweis:

Nach Theorem 2.26 kann G mit den Operationen 1–3 in eine Stufenmatrix S vom Rang k überführt werden (G hat vollen Rang). Aus dieser Stufenmatrix werden anschließend die Spalten so vertauscht, dass die resultierende Stufenmatrix den Typ $(1, \dots, k)$ besitzt. \square

Definition 2.47

Die Gestalt $[I_k|X]$ einer Generatormatrix wird als Normalform dieser bezeichnet.

Die Gestalt $[Y|I_{n-k}]$ einer Paritätskontrollmatrix wird als Normalform dieser bezeichnet.

Mithilfe der Normalformen kann nun gezeigt werden, dass die Generator- und die Paritätskontrollmatrizen eines Codes exakt auf dieselbe Art und Weise ineinander umgerechnet werden können, wie die zugrunde liegenden Matrizen zweier dualer Vektor-Matroiden (vgl. Theorem 2.28).

Theorem 2.48

Ist $G = [I_k|D]$ die Generatormatrix eines Codes C , so ist $H = [-D^T|I_{n-k}]$ eine Paritätskontrollmatrix für C .

Beweis: ([LX04, Theorem 4.5.9] auf beliebige Grundkörper verallgemeinert)

Es gilt $HG^T = D - D = 0$. Außerdem ist offensichtlich, dass die Zeilen von H linear unabhängig sind. Also folgt nach Lemma 2.43, dass H eine Paritätskontrollmatrix für C ist. \square

Mit dem Lemma 2.43 wurde auch die Grundlage für das folgende Theorem gelegt, dass Aufschluss über den Zusammenhang von linearer Unabhängigkeit und minimalem Codeabstand gibt. Damit kann schließlich die Äquivalenz von GENERALIZED MINIMUM DISTANCE und GENERALIZED MINIMUM DISTANCE* gezeigt werden.

Theorem 2.49

Es sei C ein linearer Code und H eine Paritätskontrollmatrix für C . Dann gelten folgende zwei Aussagen:

1. $d(C) \geq d$ genau dann, wenn beliebige $d - 1$ Spalten von H linear unabhängig sind.
2. $d(C) \leq d$ genau dann, wenn H d Spalten besitzt, die linear abhängig sind.

Beweis: ([LX04, Theorem 4.5.6] auf beliebige Grundkörper verallgemeinert)

Nach Lemma 2.43 bzw. nach Bemerkung 2.44 enthält C ein Codewort $v = (v_1, \dots, v_n)$ mit Gewicht $wt(v) = e > 0$ (wobei die Nicht-Null-Koordinaten v_{i_1}, \dots, v_{i_e} seien) genau dann, wenn

$$0 = vH^T = v_{i_1}c_{i_1}^T + \dots + v_{i_e}c_{i_e}^T.$$

Dies gilt genau dann, wenn es e Spalten in H gibt (nämlich c_{i_1}, \dots, c_{i_e}), die linear abhängig sind. Damit folgt:

$$\begin{aligned} wt(C) \geq d &\Leftrightarrow C \text{ enthält kein Nicht-Null-Codewort } v \text{ mit } wt(v) \leq d - 1 \\ &\Leftrightarrow \text{beliebige } \leq d \text{ Spalten von } H \text{ sind linear unabhängig.} \end{aligned}$$

Dies beweist die erste Aussage. Die zweite gilt nach folgenden Äquivalenzen:

$$\begin{aligned} wt(C) \leq d &\Leftrightarrow C \text{ enthält ein Nicht-Null-Codewort } v \text{ mit } wt(v) \leq d \\ &\Leftrightarrow H \text{ enthält } \leq d \text{ Spalten (und damit genau } d), \text{ die linear abhängig sind.} \end{aligned}$$

□

Korollar 2.50

Es sei C ein linearer Code und H eine Paritätskontrollmatrix für C . Dann sind die beiden folgenden Aussagen äquivalent:

1. $wt(C) = d$
2. beliebige $d-1$ Spalten von H sind linear unabhängig und H enthält d linear abhängige Spalten.

Beweis: ([LX04, Corollary 4.5.7] auf beliebige Grundkörper verallgemeinert)

Die Behauptung folgt direkt aus Theorem 2.49.

□

Korollar 2.51

Die Probleme **MINIMUM CIRCUIT** und **GENERALIZED MINIMUM DISTANCE*** sind many-one-äquivalent¹. Die Matrizen beider Probleme können in Polynomialzeit ineinander transformiert werden.

Beweis:

Die Äquivalenz der Probleme ergibt sich aus der zweiten Aussage von Theorem 2.49. Nach Korollar 2.46 kann eine Generatormatrix in Polynomialzeit in ihre Normalform transformiert werden. Dies ist in analoger Weise auch für die Paritätskontrollmatrix möglich. Theorem 2.48 beschreibt den Zusammenhang zwischen Generatormatrix und Paritätskontrollmatrix. Die Normalformen sind offensichtlich in Polynomialzeit ineinander überführbar.

□

Aufgrund der Äquivalenz von **GENERALIZED MINIMUM DISTANCE*** und **MINIMUM CIRCUIT**, sowie der Aussage von Theorem 2.33 und der erläuterten Einfachheit des Hamming-Gewichts (besitzt nur ein Funktionsargument) gegenüber dem Hamming-Abstand wird im Allgemeinen nur noch das Problem (**GENERALIZED**) **MINIMUM DISTANCE** betrachtet.

Eine interessante Tatsache, die sich aus den obigen Aussagen ergibt, zeigt sich, wenn man die Äquivalenzen zwischen **MATRIXSCHWÄCHE**, **MINIMUM CIRCUIT** und **GENERALIZED MINIMUM DISTANCE*** genauer beleuchtet. Für die Äquivalenz zwischen **MATRIXSCHWÄCHE** und **MINIMUM CIRCUIT** ist exakt dieselbe Veränderung der Eingabe notwendig wie für die Äquivalenz zwischen **MINIMUM CIRCUIT** und **GENERALIZED MINIMUM DISTANCE***. Da zudem die Operation $[I_k|D] \rightarrow [-D^T|I_{n-k}]$ durch einfaches Vertauschen von Spalten und nochmalige Anwendung rückgängig gemacht wird, gilt die Äquivalenz zwischen **MATRIXSCHWÄCHE** und **GENERALIZED MINIMUM DISTANCE*** ohne Veränderung der Eingabe.

¹Zwei Mengen A und B sind many-one-äquivalent, wenn sowohl $A \leq_m B$, als auch $B \leq_m A$ gilt. Dies wird auch als $A \equiv_m B$ geschrieben. Sind beide Reduktionen zudem polynomial, so schreibt man $A \equiv_m^p B$.

2.3 Die Komplexität von Minimum Distance

Da MINIMUM DISTANCE ein Problem der Codierungstheorie ist, werden meist die binären Zahlen bzw. im allgemeineren Falle ein endlicher Körper zugrunde gelegt. Um eine Äquivalenz zu MATRIXROBUSTHEIT zu gewinnen, musste jedoch die auf beliebige Körper verallgemeinerte Variante GENERALIZED MINIMUM DISTANCE verwendet werden.

Trotz des Unterschiedes können Aussagen zu MINIMUM DISTANCE verwendet werden, da es ein Spezialfall von MATRIXROBUSTHEIT ist. Daher gilt die *NP*-Vollständigkeit, die Alexander Vardy in [Var97] für MINIMUM DISTANCE auf endlichen Körpern zeigt, auch für MATRIXROBUSTHEIT. Ob MATRIXROBUSTHEIT auch *NP*-vollständig ist, wenn nur unendliche Körper betrachtet werden, bleibt offen.

Weitere Ergebnisse zur Komplexität von MINIMUM DISTANCE liefern Dumer et al. in [DMS03]. Sie zeigen unter anderem, dass MINIMUM DISTANCE nicht mit konstantem Faktor von randomisierten Polynomialzeit-Algorithmen (RP) approximiert werden kann, sofern $RP \neq NP$. Im Gegensatz zu Vardys Ergebnis, welches im Folgenden erläutert wird, wird hierauf in dieser Arbeit nicht näher eingegangen.

Die in den meisten Fällen verwendete Form von MINIMUM DISTANCE lautet wie folgt:

Problem: MINIMUM DISTANCE

Instanz: Eine binäre $m \times n$ -Matrix M und eine positive, ganze Zahl ω .

Frage: Gibt es einen Vektor $x \in \mathbb{F}_2^n$ ungleich dem Nullvektor mit einem Gewicht von höchstens ω , so dass $Mx = 0$?

Dabei bezeichnet \mathbb{F}_2 den eindeutig bestimmten Körper mit zwei Elementen. Vardys Beweis baut hauptsächlich auf dem *NP*-Vollständigkeitsbeweis für das Problem MAXIMUM-LIKELIHOOD DECODING aus [BMvT78] auf, welches wie folgt definiert ist.

Problem: MAXIMUM-LIKELIHOOD DECODING

Instanz: Eine binäre $m \times n$ -Matrix H , ein Vektor $s \in \mathbb{F}_2^m$ und eine positive, ganze Zahl ω .

Frage: Gibt es einen Vektor $x \in \mathbb{F}_2^n$ mit einem Gewicht von höchstens ω , so dass $Hx = s$?

Dazu wird MAXIMUM-LIKELIHOOD DECODING zunächst etwas eingeschränkt, was aber die *NP*-Vollständigkeit nicht verletzt. So kann laut Vardy angenommen werden, dass man nur $x \neq 0$ sucht, alle Spalten von M verschieden sind, sowie eine Matrix M mit vollem Zeilenrang vorliegt. Außerdem werden die Spalten der Paritätskontrollmatrix H aus MAXIMUM-LIKELIHOOD DECODING als Elemente des endlichen Raums mit 2^m Elementen $\text{GF}(2^m)$ betrachtet. Dies ergibt die Variante von SUBSET SUM für endliche Körper.

Problem: FINITE-FIELD SUBSET SUM

Instanz: Eine positive, ganze Zahl m , eine Menge von $n \leq 2^m$ verschiedenen Elementen $\alpha_1, \dots, \alpha_n \in \text{GF}(2^m)$, ein Element $\beta \in \text{GF}(2^m)$ ungleich Null und eine positive, ganze Zahl $\omega \leq m - 1$.

Frage: Gibt es eine nicht-leere Teilmenge $\{\alpha_{i_1}, \dots, \alpha_{i_\delta}\}$ von $\{\alpha_1, \dots, \alpha_n\}$, so dass

$$\alpha_{i_1} + \dots + \alpha_{i_\delta} = \beta$$

und $\delta \leq \omega$?

Ausgehend von MINIMUM DISTANCE über dem Körper $\text{GF}(2^m)$ beweist Vardy nun die folgende Aussage.

Theorem 2.52

Die Existenz eines Polynomialzeit-Algorithmus für MINIMUM DISTANCE impliziert die Existenz eines Polynomialzeit-Algorithmus für FINITE-FIELD SUBSET SUM.

Beweisidee:

Für FINITE-FIELD SUBSET SUM sei die Eingabe $\alpha_1, \dots, \alpha_n, \beta$ und ω gegeben. Daraus wird eine Reihe von Matrizen A_1, \dots, A_ω konstruiert, die als Paritätskontrollmatrizen für die Codes C_1, \dots, C_ω über $\text{GF}(2^m)$ aufgefasst werden. Diese Matrizen sind so konstruiert, dass der Codeabstand von C_δ gleich $\delta + 1$ ist, wenn $\alpha_{i_1} + \dots + \alpha_{i_\delta} = \beta$ für beliebige i_1, \dots, i_δ . Im anderen Fall ist der Codeabstand von C_δ gleich $\delta + 2$.

Angenommen Φ ist ein Polynomialzeit-Algorithmus für MINIMUM DISTANCE. Dann rufen wir Φ mit $M := A_\delta$, $\omega = \delta + 1$ und $\delta = 1, \dots, \omega$ auf. Gibt Φ für eine dieser Eingaben „Ja“ zurück, so gibt es $i_1, \dots, i_{\delta'}$ mit $\alpha_{i_1} + \dots + \alpha_{i_{\delta'}} = \beta$ und die Antwort auf FINITE-FIELD SUBSET SUM lautet ebenfalls „Ja“. Liefert Φ für keine der Eingaben „Ja“, wird FINITE-FIELD SUBSET SUM mit „Nein“ beantwortet.

Die Konstruktion der Matrizen A_i und der Beweis ihrer Eigenschaften ist ein wesentlicher Teil von Vardys Arbeit. Ein weiterer Teil beschäftigt sich mit der NP -Vollständigkeit für binäre Codes und die Erweiterung auf beliebige endliche Körper.

Zusammenfassend für dieses Kapitel lässt sich sagen, dass MATRIXROBUSTHEIT mit endlichem Körper NP -vollständig ist und nicht mit konstantem Faktor von randomisierten Polynomialzeit-Algorithmen approximiert werden kann, sofern $RP \neq NP$. Ob dies auch für MATRIXROBUSTHEIT mit unendlichem Körper gilt, bleibt offen. Durch den Bezug zu MINIMUM DISTANCE und MINIMUM CIRCUIT bieten sich jedoch einige Ansatzpunkte, die Komplexität von MATRIXROBUSTHEIT genauer, im Speziellen für unendliche Körper zu bestimmen. Die Betrachtung von MATRIXROBUSTHEIT auf unendlichen Körpern ist von Bedeutung, da die Sensitivitätsmatrix auf solchen beruht.

Kapitel 3

Algorithmen

Das Problem der MATRIXROBUSTHEIT ist, wie im vorhergehenden Kapitel gezeigt wurde, für endliche Körper NP -vollständig, was die Hoffnungen auf einen schnellen Polynomialzeit-Algorithmus schwinden lässt. Im Abschnitt 3.2 werden daher ein Algorithmus mit exponentialer Laufzeit für den Worst-Case angegeben und in 3.4 die Ergebnisse von einigen Beispielrechnungen präsentiert. Dieselben Beispiele werden dort auch auf die in Abschnitt 3.3 vorgestellte Heuristik angewandt. Diese besitzt zwar eine polynomiale Laufzeit, jedoch können keine Aussagen zu der Ergebnisqualität gemacht werden. Im Folgenden Abschnitt wird zunächst ein Spezialfall des MATRIXROBUSTHEIT-Problems betrachtet, der exakt in polynomialer Zeit gelöst werden kann.

3.1 Matrixrobustheit bei einfacher linearer Abhängigkeit

Inspiziert von der Pseudo-Rang-Berechnung in [AP05] werden im Folgenden Instanzen des MATRIXROBUSTHEIT-Problems betrachtet, bei denen ausschließlich die paarweise lineare Abhängigkeit auftritt. Das heißt, wenn eine Menge von Vektoren linear abhängig ist, so sind diese stets paarweise linear abhängig. Dazu kann der folgende Sachverhalt festgehalten werden.

Lemma 3.1

Die paarweise lineare Abhängigkeit ist eine Äquivalenzrelation. Insbesondere bildet eine Menge von Vektoren Klassen von paarweise linear abhängigen Vektoren.

Beweis:

Eine Äquivalenzrelation muss die folgenden drei Eigenschaften besitzen:

Reflexivität Ein Vektor x ist linear abhängig von sich selbst, da gilt: $x = 1 \cdot x$.

Transitivität Es seien die Vektoren x und y , sowie y und z linear abhängig, d. h. $x = c \cdot y$ und $y = d \cdot z$ mit $c, d \in \mathbb{R}$. Dann gilt $x = cd \cdot z$, also ist x linear abhängig von z .

Symmetrie Ist x linear abhängig von y mit $x = c \cdot y$, so ist y linear abhängig von x mit $y = \frac{1}{c} \cdot x$.

□

Mit dieser Beobachtung kann das Problem nun genauer analysiert werden.

Input: $n \times m$ matrix M , integer $k > 0$

Output: **true** or **false** $\triangleright M$ ist robust bzw. ist nicht robust bzgl. paarw. lin. Abh.

```

1: function MATRIXROBUSTNESSPLD( $M, k$ )
2:   initialize all rows of  $M$  as 'unmarked'
3:   for  $i \leftarrow 1$  to  $m$  do
4:     if row  $i$  is not marked then  $\triangleright$  Zeile  $i$  wurde noch nicht klassifiziert
5:        $l \leftarrow 0$   $\triangleright$  Variable  $l$  speichert Kardinalität der aktuellen Klasse
6:       for  $j \leftarrow i + 1$  to  $m$  do  $\triangleright$  alle Zeilen  $\leq i$  wurden schon klassifiziert
7:         if row  $i$  is linear dependent on row  $j$  then
8:            $l \leftarrow l + 1$ 
9:           mark row  $l$ 
10:        end if
11:      end for
12:      if  $l \leq k$  then  $\triangleright$  aktuelle Klasse hat nicht genügend Elemente
13:        return false
14:      end if
15:    end if
16:  end for
17:  return true
18: end function

```

Algorithmus 3.1: Exakte Lösung von MATRIXROBUSTHEIT für Sensitivitätsmatrizen mit ausschließlich paarweise linearen Abhängigkeiten. Die Zeilenvektoren der Sensitivitätsmatrix werden klassifiziert, indem zu allen Zeilenvektorpaaren die (paarweise) lineare Abhängigkeit bestimmt wird.

Lemma 3.2

Damit eine Sensitivitätsmatrix robust gegenüber dem Löschen von k Zeilen ist, muss jede Äquivalenzklasse von paarweise linear abhängigen Vektoren über mindestens $k + 1$ Zeilen verfügen.

Beweis: Beim Ausfall von k Messgeräten ist im Worst-Case offensichtlich nur eine Äquivalenzklasse betroffen. Sind höchstens k Zeilen in dieser Klasse, so wird die gesamte Klasse gelöscht, was zu einer Verringerung des Rangs der Sensitivitätsmatrix führt. Also muss jede Klasse über mindestens $k + 1$ Zeilen verfügen, damit sie nicht vollständig eliminiert wird. \square

Aufgrund der erläuterten Einfachheit, die MATRIXROBUSTHEIT unter Annahme von ausschließlich paarweiser linearer Abhängigkeiten besitzt, lassen sich leicht Algorithmen mit polynomialer Laufzeit für die Probleme MATRIXROBUSTHEIT und MINIMALE ROBUSTE SUBMATRIX entwerfen.

Nach Lemma 3.2 muss lediglich die Größe l_i jeder Äquivalenzklasse von paarweise linear abhängigen Vektoren bestimmt und $l_i \leq k$ getestet werden. Algorithmus 3.1 demonstriert dieses Vorgehen.

Da als Eingabe eine $m \times n$ -Sensitivitätsmatrix vorliegt, lässt sich die lineare Abhängigkeit zweier Vektoren in $O(n)$ bestimmen. Die Gesamtlaufzeit ergibt sich zu $O(m^2 \cdot n)$. Die Laufzeit lässt sich verringern zu $O(m \cdot c \cdot n)$, wobei c die Anzahl von Äquivalenzklassen bezeichnet. Dazu wird von jeder Klasse der erste gefundene Vektor in eine verkettete Liste eingefügt und stets nur die lineare Abhängigkeit zu diesen Repräsentanten geprüft.

Input: $m \times n$ matrix M , integer $k > 0$ \triangleright M muss robust bzgl. k sein, sonst Fehler
Output: submatrix of M \triangleright kleinste bzgl. paarw. lin. Abh. robuste Submatrix von M

```

1: function MINIMALROBUSTSUBMATRIXPLD( $M, k$ )
2:   initialize all rows of  $M$  as 'unmarked'
3:   for  $i \leftarrow 1$  to  $m$  do
4:     if row  $i$  not marked then  $\triangleright$  Zeile  $i$  wurde noch nicht klassifiziert
5:        $l \leftarrow 0$   $\triangleright$  Variable  $l$  speichert Kardinalität der aktuellen Klasse
6:       for  $j \leftarrow i + 1$  to  $m$  do  $\triangleright$  alle Zeilen  $\leq i$  wurden schon klassifiziert
7:         if row  $i$  is linear dependent on row  $j$  then
8:           if  $l \leq k$  then
9:              $l \leftarrow l + 1$ 
10:            mark row  $j$ 
11:           else  $\triangleright$  Klasse besitzt  $k + 1$  Elemente
12:             delete row  $j$   $\triangleright$  weitere sind überflüssig
13:           end if
14:         end if
15:       end for
16:       if  $l < k$  then  $\triangleright$  aktuelle Klasse hat nicht genügend Elemente
17:         return error "The input matrix is not robust."
18:       end if
19:     end if
20:   end for
21:   return  $M$ 
22: end function

```

Algorithmus 3.2: Berechnung der MINIMALEN ROBUSTEN SUBMATRIX von Sensitivitätsmatrizen mit ausschließlich paarweise linearen Abhängigkeiten

Mit geringfügigen Modifikationen an Algorithmus 3.1 lässt sich auch die MINIMALE ROBUSTE SUBMATRIX mit Laufzeit $O(m^2 \cdot n)$ für Sensitivitätsmatrizen mit ausschließlich paarweise linear abhängigen Vektoren berechnen, wie Algorithmus 3.2 zeigt.

Bemerkung 3.3

Bezeichnet man eine Menge von linear abhängigen Vektoren, deren echte Teilmengen nicht linear abhängig sind, als *Kreise* (vgl. Bezeichnung 2.2), so ist offensichtlich, dass bei ausschließlich paarweiser linearer Abhängigkeit alle möglichen Kreise eine Kardinalität von zwei besitzen. Erlaubt man auch andere lineare Abhängigkeiten, so zeigt das folgende Beispiel, dass die Kreise dann unterschiedliche Kardinalitäten annehmen können.

Seien $a := (1, 0, 0)$, $b := (0, 1, 1)$, $c := (1, 1, 1)$, $d := (0, -1, 1)$ und $e := (1, 2, 0)$. Dann sind $A := \{a, b, c\}$ wegen $c = a + b$ und $B := \{c, d, e\}$ wegen $c = d + e$ zwei linear abhängige Mengen. Diese sind aufgrund der Tatsache, dass es keine paarweisen linearen Abhängigkeiten gibt, Kreise. Wegen $a = -b + c = -b + d + e$ ist $\{a, b, d, e\}$ eine abhängige Menge und offensichtlich auch ein Kreis, da alle dreielementigen Teilmengen davon linear unabhängig sind.

Dies deutet an, dass die Menge aller Kreise schnell recht kompliziert werden kann, wenn man mehr als paarweise lineare Abhängigkeit erlaubt, was eine Erklärung für die Komplexität von MATRIXROBUSTHEIT ist und auch dafür, dass die obigen einfachen Algorithmen nicht auf allgemeinere Problemstellungen erweitert werden können.

3.2 Modellierung als Mixed Integer Linear Program (MIP)

Die Konstrukte eines Mixed Integer Linear Programs, kurz MIP sind Verallgemeinerungen derer eines Linear Programs (LP). In einem LP wird über Variablen ein Lösungsraum definiert, der durch die Angabe von Ungleichungen eingeschränkt wird. Betrachtet werden diese Einschränkungen als Halbräume in einem der Variablenanzahl entsprechend dimensioniertem Raum. Schneidet man alle Halbräume, so ergibt sich der gültige Lösungsraum. In diesem Raum kann in Polynomialzeit eine gegebene lineare Zielfunktion minimiert werden, wie Leonid Khachiyan durch die Angabe eines Algorithmus [Kha79] bewiesen hat. Dieser besitzt jedoch unpraktikabel hohe Laufzeiten, die mit der projektiven Methode von Narendra Karmarkar [Kar84] vermieden werden. Eine detailliertere Einführung in Linear Programming gibt z. B. [CLRS01].

Alle Variablen eines LPs sind stets aus den reellen Zahlen zu wählen. Mixed Integer Linear Programs dagegen erlauben auch die Beschränkung auf ganzzahlige Variablen bzw. mit entsprechenden Ungleichungen auf binäre Variablen. Allerdings können MIP-Probleme i. A. nicht in Polynomialzeit gelöst werden, wie die folgende einfache Reduktion von dem Problem 3-SAT zeigt.

Theorem 3.4

Das Lösen von MIPs ist NP-hart.

Beweis:

Der Beweis wird mit einer Reduktion von 3-SAT¹ auf MIP geführt. Die zu minimierende Zielfunktion wird als konstant angenommen, so dass zur Lösung des MIP lediglich eine zulässige Lösung, d. h. eine Lösung aus dem Lösungsraum gesucht werden muss.

Sei ϕ eine Instanz von 3-SAT mit den Variablen x_1, \dots, x_n , woraus eine Instanz ψ für MIP mit den binären Variablen y_1, \dots, y_n und z_1, \dots, z_n konstruiert wird, wobei y_i und z_i zu den Literalen x_i bzw. \bar{x}_i korrespondieren. Um zu garantieren, dass $y_i + z_i = 1$ ist, werden folgenden Ungleichungen eingeführt:

$$\forall 1 \leq i \leq n : \quad y_i + z_i \geq 1 \quad \wedge \quad y_i + z_i \leq 1$$

Weiterhin gibt es für jede Klausel in der 3-SAT-Instanz eine Ungleichung in der MIP-Instanz, wobei die Summe der zu den Literalen korrespondierenden Variablen größer oder gleich 1 sein soll, also wird z. B. für $(x_1 \vee x_3 \vee \bar{x}_4)$ die Ungleichung $y_1 + y_3 + z_4 \geq 1$ festgelegt.

Offensichtlich gilt nach dieser Konstruktion $\phi \in 3\text{-SAT} \Leftrightarrow \psi$ ist mittels MIP lösbar. \square

Das Lösen von MIP-Problemen ist NP-vollständig, da es NP-Algorithmen dafür gibt. Aufgrund der vielseitigen Einsetzbarkeit von MIPs sind relativ gute Löser vorhanden [Fou05].

3.2.1 Theoretische Entwicklung

Eine Anwendung von MIPs ist die MATRIXROBUSTHEIT, wie das folgende Theorem zeigt.

¹Das Problem 3-SAT besitzt als Eingabe einen logischen Ausdruck in konjunktiver Normalform mit genau drei Literalen pro Disjunktion, also z. B. $(x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee \bar{x}_3 \vee x_4)$. Die Frage ist, ob es eine erfüllende (so dass der Ausdruck wahr bzw. 1 wird) Belegung $x_i \mapsto \{0, 1\}$ gibt.

Theorem 3.5

Das Problem der MATRIXROBUSTHEIT kann mithilfe eines MIP gelöst werden. Ist eine $m \times n$ -Sensitivitätsmatrix gegeben, so beträgt die Laufzeit $O(2^m \cdot (n \cdot m)^{O(1)})$.

Beweis:

Statt der MATRIXROBUSTHEIT wird das dazu äquivalente Problem der UMFASSENDSTEN HYPEREBENE betrachtet (siehe Abschnitt 1.5).

Problem: UMFASSENDSTE HYPEREBENE

Instanz: Eine $m \times n$ -Sensitivitätsmatrix M über einem beliebigen Körper K mit Rang n , $m \geq n$ und eine positive ganze Zahl k .

Frage: Gibt es eine Hyperebene in dem Raum K^n , die mindestens $n - k$ viele Zeilenvektoren aus M enthält?

Es soll also eine Hyperebene von K^n gefunden werden, die möglichst viele Vektoren enthält. Dazu wird für jede Zeile der Sensitivitätsmatrix (im Folgenden mit y_i bezeichnet) eine binäre Variable d_i eingeführt. Dabei soll gelten:

$$d_i = 0 \quad \Leftrightarrow \quad y_i \text{ liegt in der Hyperebene}$$

Mit dieser Voraussetzung muss lediglich die Summe der d_i minimiert werden:

$$\text{minimiere}(d_1 + \dots + d_m) \tag{3.1}$$

Die gesuchte Hyperebene wird durch ihren Normalenvektor dargestellt, der im Folgenden mit x bezeichnet wird. Die einzige Bedingung, die dieser Vektor erfüllen muss, ist, dass er nicht der Nullvektor ist. Diese Forderung wird weiter unten betrachtet und umgesetzt. Um nun zu testen, ob ein Vektor in der Hyperebene liegt, kann das Skalarprodukt $\langle \cdot, \cdot \rangle$ verwendet werden, da gilt:

$$\langle y_i, x \rangle = 0 \quad \Leftrightarrow \quad y_i \text{ steht senkrecht zu } x$$

Nun muss diese Tatsache mit den Variablen d_i gekoppelt werden. Das geschieht mithilfe von je zwei Nebenbedingungen pro Zeile:

$$\langle y_i, x \rangle - d_i \leq 0 \tag{3.2}$$

$$-1 \cdot \langle y_i, x \rangle - d_i \leq 0 \tag{3.3}$$

Da die Länge der Vektoren für Lagebetrachtungen nicht relevant sind, kann vorausgesetzt werden, dass $\|y_i\| \leq 1$ und $\|x\| \leq 1$ ist. Somit gilt $-1 \leq \langle y_i, x \rangle \leq 1$. Nun können die folgenden Aussagen gemacht werden, woraus die oben gewünschte Eigenschaft der Variablen d_i folgt:

$$\langle y_i, x \rangle > 0 \quad \Rightarrow \quad d_i > 0 \quad \text{wegen (3.2)}$$

$$\langle y_i, x \rangle = 0 \quad \Rightarrow \quad d_i = 0 \quad \text{wegen (3.1), (3.2) und (3.3)}$$

$$\langle y_i, x \rangle < 0 \quad \Rightarrow \quad d_i > 0 \quad \text{wegen (3.3)}$$

Die Voraussetzung $\|y_i\| \leq 1$ kann o. B. d. A. als gegeben angenommen werden, während $\|x\| \leq 1$ dagegen explizit gefordert werden muss, aber durch folgende einfache Bedingungen für jede Komponente von x erreicht werden kann:

$$-\frac{1}{n} \leq x_i \leq \frac{1}{n} \quad (3.4)$$

Nun bleibt noch die Voraussetzung $\|x\| > 0$ (also $x \neq 0$) umzusetzen. Dies kann allerdings nicht durch einfache Ungleichungen erreicht werden, da es keine konvexe Eigenschaft ist. Eine Möglichkeit wäre es, quadratische Programmierung zu verwenden, um dies umzusetzen. Jedoch sind entsprechende Löser kaum verfügbar und die zusätzlichen Möglichkeiten machen sie wesentlich langsamer als MIP-Löser. Daher wird hier der Ansatz gewählt, den Algorithmus n mal zu durchlaufen und in jedem Durchlauf eine Komponente des Vektors x fest, größer als Null, aber die Ungleichung (3.4) nicht verletzend zu wählen. Dies kann z. B. mit der Einschränkung $x_i = 1/n$ erreicht werden.

Insgesamt werden n reelle Variablen für den Normalenvektor und m binäre Variablen verwendet. Somit ist ein Algorithmus mit der Laufzeit $O(2^m \cdot (n \cdot m)^{O(1)})$ verfügbar, der im folgenden Abschnitt explizit formuliert wird. Die angegebene Laufzeit ergibt sich, wenn jede mögliche Belegung der m binären Variablen ausprobiert wird und da LPs in polynomialer Zeit gelöst werden können. \square

3.2.2 Praktische Umsetzung

Das in Theorem 3.5 entwickelte MIP wurde in der Modellierungssprache GNU MathProg umgesetzt, deren Definition in [Mak05] nachgelesen werden kann. Algorithmus 3.3 zeigt den Strukturteil des MIPs, d. h. alle verwendeten Konstrukte bis auf die Parameter- und Variablenbelegung. Das Zeichen `%` dient als Platzhalter. Laut des Beweises von Theorem 3.5 soll das MIP n -mal gelöst werden, wobei `%` sukzessiv durch die Zahlen 1 bis n ersetzt wird. Zu beachten ist weiterhin, dass die Voraussetzung $\|y_i\| \leq 1$ im MIP durch die Variablen r_i umgesetzt wird. Dabei wird jedes Vorkommen von $y_{j,i}$ durch r_i dividiert. Da die Ungleichung

$$\sum_{i \in N} |y_{j,i}| \geq \sqrt{\sum_{i \in N} y_{j,i}^2},$$

gilt, wird die linke, schneller zu berechnende Formel in der Ungleichung statt der Norm von y_i (rechte Seite) verwendet, um die Voraussetzung zu implementieren.

Da das MIP mehrfach für dasselbe Problem gelöst werden muss, wird für diese Aufgabe ein Python-Skript eingesetzt, das die Sensitivitätsmatrix aus einer Datei im CSV-Format (Comma Separated Values, siehe [Sha05]; hier mit dem Tabulator als Trennzeichen) einliest. Eine Einführung und für die hier verwendeten Zwecke ausreichende Übersicht der Programmiersprache Python ist in dem Tutorial [vR05] zu finden. Das Skript generiert im einfachsten Fall nacheinander die n MIPs und startet das Programm GLPSOL aus dem Programmpaket GNU Linear Programming Kit, Version 4.8 (zur Anwendung von GLPSOL siehe [Mak05]), welches das jeweilige MIP löst. Nach jeder Berechnung wird die Ausgabe von GLPSOL eingelesen. Damit kann dann das globale Minimum über alle Teillösungen bestimmt und die zugehörigen Werte ausgegeben werden. Dazu zählen der Normalenvektor x , und bei welchen Zeilen d_j ungleich Null ist, also welche Zeilen gelöscht werden müssen, damit die Sensitivitätsmatrix im Rang fällt.

- 1: **define parameter** n ▷ Spaltenanzahl der Sensitivitätsmatrix
- 2: **define parameter** m ▷ Zeilenanzahl der Sensitivitätsmatrix

- 3: **define set** $N := \{1, \dots, n\}$ ▷ Menge Spaltenindizes
- 4: **define set** $M := \{1, \dots, m\}$ ▷ Menge Zeilenindizes

- 5: **define parameter** $\forall j \in M \forall i \in N: y_{j,i}$ ▷ $j \cdot i$ Variablen für die Sensitivitätsmatrix

- 6: **define parameter** $\forall j \in M: r_j \leftarrow \sum_{i \in N} |y_{j,i}|$ ▷ Divisor für y_j ; Ziel: $\|y_j\| \leq 0$

- 7: **define binary variable** $\forall j \in M: d_j$ ▷ binäre Variablen d_j für die Zeilen
- 8: **define variable** $\forall i \in N: x_i$ ▷ Komponenten x_i des Normalenvektors

- 9: **minimize** $\sum_{j \in M} d_j$

- 10: **subject to** $\forall j \in M: \sum_{i \in N} \left(\frac{y_{j,i}}{r_j} \cdot x_i\right) - d_j \leq 0$ ▷ $\langle y_i, x \rangle - d_i \leq 0$
- 11: **subject to** $\forall j \in M: \sum_{i \in N} \left(\frac{y_{j,i}}{r_j} \cdot x_i\right) + d_j \geq 0$ ▷ $-1 \cdot \langle y_i, x \rangle - d_i \leq 0$
- 12: **subject to** : $x_{\%} = \frac{1}{n}$ ▷ % steht für den aktuellen Schleifenindex
- 13: **subject to** $\forall i \in N: x_i \geq -\frac{1}{n}$
- 14: **subject to** $\forall i \in N: x_i \leq \frac{1}{n}$

Algorithmus 3.3: MIP zur (Teil-)Lösung von UMFASSENDSTE HYPEREBENE. Dargestellt ist der sogenannte Strukturteil des MIPs, die Belegung der Parameter n, m und der Variablen $y_{j,i}$ erfolgt in dem hier nicht abgebildeten Datenteil.

Da Zeilen in der Matrix, deren Komponenten alle Null sind, keinen Einfluss auf den Rang der Sensitivitätsmatrix haben, werden diese vom Skript während des Preprocessings identifiziert und bei der Problemlösung ignoriert. Außerdem wird der Spärlichkeitsfaktor (Anteil der Nulleinträge) der Sensitivitätsmatrix zur manuellen Abschätzung der Laufzeit des Programms bestimmt.

Offensichtlich löst der hier implementierte Algorithmus MATRIXROBUSTHEIT über einem unendlichen Körper, was speziell für die Beispielrechnungen, die in Abschnitt 3.4 vorgestellt werden, notwendig ist. Die Frage nach der Komplexität von MATRIXROBUSTHEIT für unendliche Körper ist jedoch offen.

3.2.3 Verbesserungen

Kleinstes erreichbares Minimum Wird bei einem MIP festgestellt, dass es ausreicht, eine einzelne Zeile der Sensitivitätsmatrix zu löschen, so kann die Berechnung eingestellt werden, da es kein besseres Ergebnis gibt.

Weiterverwendung von Teillösungen Da das absolute Minimum von allen n MIPs bestimmt werden soll und diese sukzessive gelöst werden, können bereits gewonnene Werte weiterverwendet werden. So wird, sofern bereits eine Teillösung berechnet wurde, diese in den folgenden Teilproblemen benutzt, um deren Lösungsräume einzuschränken und so den Lösungsprozess zu beschleunigen. Um dieses Vorgehen zu realisieren, wird die zusätzliche Bedingung $\sum_{j \in M} d_j \leq \%$ im MIP eingeführt. Dabei wird % durch das derzeit beste Minimum ersetzt. Falls ein solches noch nicht bestimmt wurde, wird die gesamte Bedingung ignoriert.

Paarweise lineare Abhängigkeiten Eine wesentlich tiefgreifendere Änderung am MIP stellt die Elimination von paarweise linearen Abhängigkeiten dar. Der Ansatzpunkt hierfür ist der folgende: Wenn ein Vektor aus einer Klasse von paarweise linear abhängigen Vektoren gelöscht wird, müssen auch die anderen Vektoren aus dieser Klasse gelöscht werden, damit die Löschung einen Effekt hat (vgl. Lemma 3.2). Betrachtet man diese Aussage aus einem anderen Blickwinkel, so kann man auch nur die Äquivalenzklassen betrachten und weist ihnen die Anzahl der zugehörigen Vektoren als Gewicht zu. Das Python-Skript fasst dazu im Preprocessing paarweise linear abhängige Vektoren zusammen, indem es alle bis auf den jeweils ersten Repräsentanten löscht und dabei die Gewichte bestimmt. In der zu minimierenden Summe werden diese Gewichte mit den binären Variablen für jede Klasse multipliziert. Folgende zwei Zeilen werden dafür abgeändert:

9: **minimize** $\sum_{j \in M} d_j$

15: **subject to** : $\sum_{j \in M} d_j \leq \%$

Dabei bezeichnet % wieder das beste derzeit bestimmte Minimum aus den vorhergehenden Teillösungen. Als Resultat der Änderung enthält das Skript stattdessen:

9: **minimize** $\sum_{j \in M} w_j \cdot d_j$

15: **subject to** : $\sum_{j \in M} w_j \cdot d_j \leq \%$

Zusätzlich werden die Variablen w_j angelegt und im Daten-Teil des MIP mit den im Preprocessing errechneten Werten ausgestattet.

Zeitlimitierung Um dem Benutzer möglichst zeitig einen Blick auf suboptimale Lösungen zu geben, wird GLPSOL auf den n MIPs mit einem bestimmten Zeitlimit gestartet. Wurden bei mehreren MIPs nur suboptimale Lösungen bestimmt, werden diese in einem nächsten Anlauf nochmals berechnet, diesmal jedoch mit einem doppelt so großem Zeitfenster. Dies wird wiederholt, bis alle MIPs gelöst wurden. Bleibt nur ein MIP übrig, so setzt das Python-Skript selbstverständlich keine Zeitschranke mehr ein. Die benötigte Zeit ist mit dieser Technik nur höchstens dreimal so hoch wie zuvor, jedoch werden die schnell lösbaren Probleme früher berechnet.

Die Zeitfenster-Technik hat noch weitere Vorteile, die sich aus der Kombination mit den anderen Verbesserungen ergeben. So kann das kleinste erreichbare Minimum früher gefunden werden, als wenn alle Teilprobleme vollständig nacheinander durchgerechnet werden. Auch werden auf diese Weise (suboptimale) Teillösungen schneller berechnet und können als Schranken für andere und sogar dasselbe Teilproblem verwendet werden.

3.3 Heuristik

Da MATRIXROBUSTHEIT (für endliche Körper) ein NP -vollständiges Problem ist, könnte eine schnelle, aber das Ergebnis nur näherungsweise bestimmende heuristische Vorgehensweise von Nutzen sein. Eine einfache Möglichkeit einen solchen Algorithmus zu gewinnen ist es, das in Abschnitt 3.2 vorgestellte Python-Skript so abzuändern, dass der Benutzer eine Zeit angibt, bei der das Programm abbricht. Daraufhin können die bis dahin gewonnenen suboptimalen Lösungen als Näherung an die exakte Lösung des Problems abgelesen werden.

Eine weitere Möglichkeit für eine Heuristik besteht in einer grundlegenden Abänderung des MIP. Der Unterschied ist, dass nicht nach einem optimalen Normalenvektor gesucht

Input: $m \times n$ matrix $M = (y_{j,i})$

Output: number of deleted rows \triangleright nach dem Löschen dieser Zeilen fällt M im Rang

```

1: function MATRIXROBUSTNESSHEURISTIC( $M$ )
2:    $l \leftarrow 0$   $\triangleright$  Zähler für gelöschte Zeilen
3:   for  $k \leftarrow 1$  to  $n$  do
4:     solve the  $k$ -th LP  $\triangleright$  gleich dem MIP 3.3 bis auf die Definition von  $d_j$ :  
 $\triangleright$  define variable  $\forall j \in M: d_j$ 
5:   end for
6:   search for global best solution and its normal vector  $x$ 
7:   if the solution is not zero then  $\triangleright x$  steht nicht senkrecht zu allen Vektoren
8:     determine the vector  $y$  with the smallest scalar product with  $x$ 
9:     delete  $y$ 
10:     $l \leftarrow l + 1$ 
11:    goto line 3
12:  else
13:    return  $l$ 
14:  end if
15: end function

```

Algorithmus 3.4: Heuristik basierend auf LPs zur Lösung von MATRIXROBUSTHEIT mit polynomialer Laufzeit

wird, sondern nach einem „Durchschnittsnormalenvektor“. Dazu werden die Variablen d_i nun zu reellen Variablen, die die Abweichung eines Vektors zu dem Normalenvektor messen, wobei positive und negative Abweichungen gleichwertig behandelt werden. Über die Abweichungen wird minimiert, also genau wie im MIP über die Summe der d_i . Ein Vektor x , der diese Summe minimiert, wird als „Durchschnittsnormalenvektor“ verwendet. Da es nun keine auf die ganzen Zahlen eingeschränkte Variablen mehr gibt, liegt ein LP vor, welches in polynomialer Zeit gelöst werden kann.

Auch diese Heuristik enthält die nicht-konvexe Bedingung $\|x\| > 0$ und wird daher n -mal gelöst, wobei jeweils eine Komponente von x fest gewählt wird. Sind alle n Teilprobleme gelöst, wird das beste Ergebnis herausgefiltert und der Vektor mit der größten Abweichung zur Hyperebene (also mit der kleinsten Abweichung zum Normalenvektor x) gesucht und gelöscht. Dieses Vorgehen wird solange wiederholt, bis die Summe der d_i gleich Null ist, mit anderen Worten bis ein Normalenvektor gefunden wurde, zu dem alle verbliebenen Zeilenvektoren senkrecht stehen. Algorithmus 3.4 fasst das hier motivierte Vorgehen zusammen.

Von den Verbesserungen des MIPs lässt sich die Zusammenfassung der Klassen paarweiser linear abhängiger Vektoren übernehmen. So kann durch ein Preprocessing die Dimension der Problem Instanz verkleinert werden.

Bemerkung 3.6

Die hier vorgestellte Heuristik wäre ein exakter Algorithmus, wenn der Vektor mit dem größtem Skalarprodukt zu dem Normalenvektor stets nicht in der optimalen Hyperebene liegt. Dies gilt allerdings nicht. Betrachte dazu eine Sensitivitätsmatrix M mit der Spaltenanzahl $n = 3$, die k paarweise linear abhängige Vektoren, dazu einen einzelnen senkrechten Vektor und weitere k paarweise linear abhängige Vektoren, die nur einen kleinen Winkel zu den ersten k Vektoren besitzen, aber in der durch die ersten $k + 1$ Vektoren aufgespannten

Ebene liegen. Zusätzlich enthalte M k paarweise abhängige Vektoren, die senkrecht auf der Hyperebene stehen. Abbildung 3.1 verdeutlicht diese Konstellation.

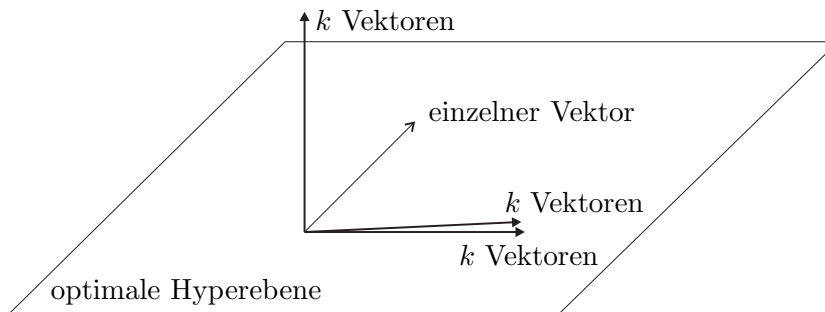


Abbildung 3.1: Beispiel zur Heuristik

Da $3 \cdot k$ Vektoren mehr Einfluss in der zu minimierenden Summe besitzen als ein einzelner Vektor, wird bei der Bildung eines „Durchschnittsnormalenvektors“ der einzelne Vektor vernachlässigt und die korrespondierende Hyperebene liegt in der Nähe der $3 \cdot k$ Vektoren. Somit wird der einzelne Vektor im Schritt 9 von Algorithmus 3.4 gelöscht, also ein Vektor aus der optimalen Hyperebene.

3.4 Experimentelle Ergebnisse vom MIP-Algorithmus und der Heuristik

Von der Firma DIGSILENT GmbH wurden aus vier Beispiel-Stromnetzen die entsprechenden Sensitivitätsmatrizen zum Testen des MIP-Algorithmus und der LP-Heuristik generiert. Diese Stromnetze werden im Folgenden zusammen mit den Ergebnissen vorgestellt. Zum Ablauf der Sensitivitätsmatrizen-Generierung sei erneut auf [AP05] verwiesen. In allen Beispielen werden jeweils die Wirk- und die Blindleistung gemessen und sollen an den sogenannten States überwacht werden. Die Matrizen sind bis auf den Datensatz für das namibische Stromnetz auf ganze Zahlen im Bereich $-9, \dots, 0, \dots, 9$ diskretisiert, während die Sensitivitätsmatrix des namibischen Stromnetzes auf den Bereich $-999, \dots, 0, \dots, 999$ diskretisiert wurde. Damit werden numerische Fehler entfernt, die bei der Berechnung der Matrizen entstehen, wobei die Wahl des Diskretisierungsbereiches auf Erfahrungswerten beruht.

Als Testrechner wurde ein AMD Athlon 64 3400+ mit 2,4GHz, 512KB L2-Cache und 1GB Hauptspeicher eingesetzt. Debian GNU/Linux 3.1 diente als Betriebssystem.

NineBusSystem Dieses kleine Energieversorgungsnetz (EVN) besitzt neun Knoten (Sammelschienen) und eine Masche (Kreis, der die Busse 4, 7, 8 und 9 umfasst). Es wurde zu Testzwecken entworfen. Abbildung 3.2 zeigt die Topologie des Netzes.

Insgesamt umfasst das Netzwerk 40 Messgeräte und 12 States, weshalb eine 40×12 -Sensitivitätsmatrix vorliegt. Diese Matrix enthält keine Null-Zeilen und besteht zu 56,8% aus Null-Einträgen. Aufgrund von paarweisen linearen Abhängigkeiten können 12 Zeilen ignoriert werden.

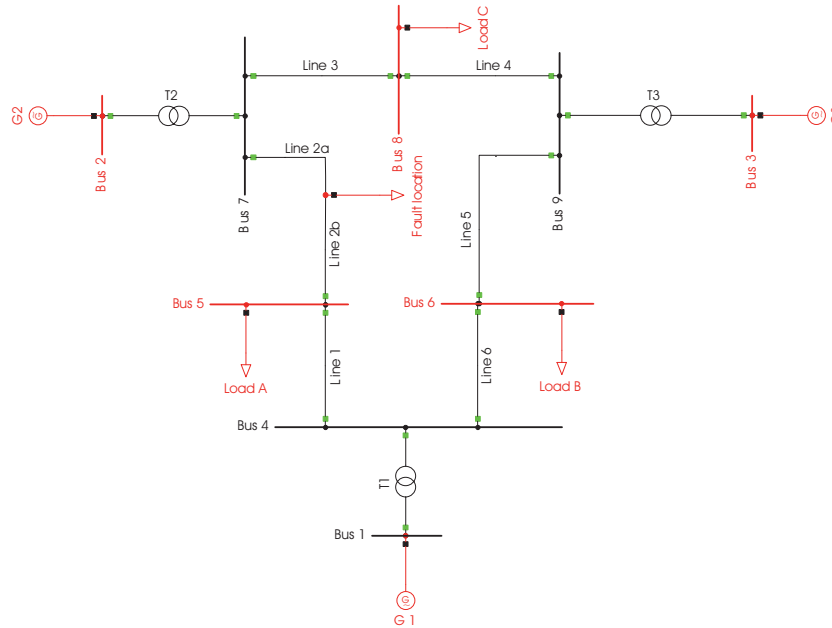


Abbildung 3.2: Netzplan für das Beispiel „Nine Bus System“. Die kleinen Quadrate sind mögliche Positionen für Messgeräte, dienen aber auch als Schalter für die darunterliegende Verbindung (gefüllte Kästchen stehen für geschlossene Schalter). Grüne Farbe kennzeichnet die Präsenz von Messgeräten, während rote Sammelschienen (Busse) mit rotem Abgang für die States stehen.

Treelike System Auch hier liegt ein EVN vor, wobei jedoch durch die Abschaltung von Line 4-5 die Masche aufgetrennt wird und eine baumartige Struktur vorliegt. Abbildung 3.3 zeigt dieses Netz.

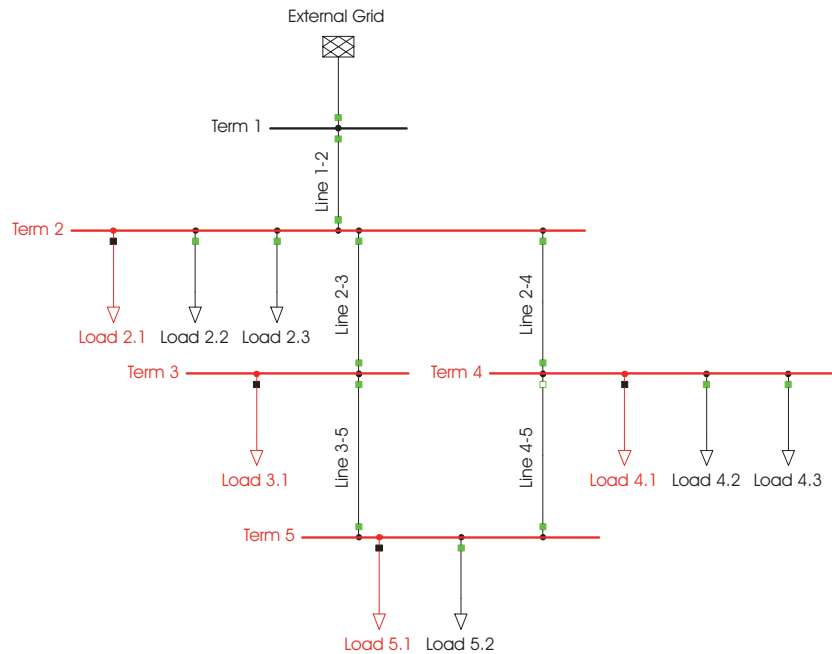


Abbildung 3.3: Netzplan für das Beispiel „Treelike System“. Die kleinen Quadrate sind mögliche Positionen für Messgeräte, dienen aber auch als Schalter für die darunterliegende Verbindung (gefüllte Kästchen stehen für geschlossene Schalter). Grüne Farbe kennzeichnet die Präsenz von Messgeräten, während rote Sammelschienen (Busse) mit rotem Abgang für die States stehen.

Gemessen werden 40 Werte, die zur Bestimmung von acht States genügen sollen, gegeben ist also eine 40×8 -Matrix. Die diskretisierte Sensitivitätsmatrix enthält 14 Null-Zeilen und besitzt nach dem Löschen dieser Zeilen einen Spärlichkeitsfaktor von 79,2%. Bei genauerem Hinsehen lässt sich jedoch der einfache Aufbau der Matrix erkennen, denn die Zeilenvektoren können in 12 Mengen aufgeteilt werden, wobei innerhalb einer Menge alle Vektoren paarweise linear abhängig sind. Außerdem enthält jeder Vektor entweder nur an Komponenten mit geraden oder ungeraden Indizes Einträge ungleich Null, es findet also zusätzlich eine Zweiteilung der Matrix statt. Diese Einteilung ist das Resultat der separaten Messung von Wirk- und Blindleistung, sowie fehlender Impedanzen (keine Widerstände, also keine Veränderung des Stroms) auf den Verbindungen. Schließlich gibt es nur die Einträge ± 9 und 0, wobei ein Vektor stets entweder keine positiven oder keine negativen Einträge enthält.

MV/LV Distribution Im Gegensatz zu den bisherigen Beispielen liegt hier ein Verteilungsnetz für die Mittelspannungs- (MV) und die Niederspannungsebene (LV) vor, welches in Abbildung 3.4 dargestellt ist. Trotz der geringen Ausmaße kommt es einem realen Netz recht nahe.

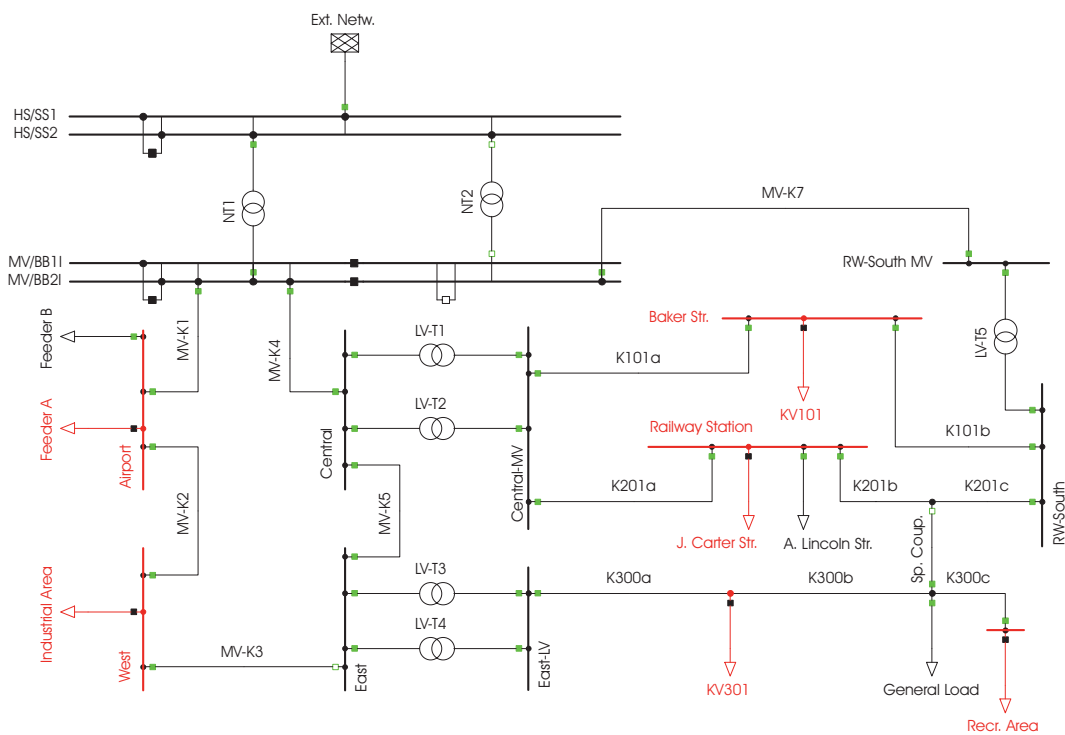


Abbildung 3.4: Netzplan für das Beispiel „MV/LV Distribution“. Die kleinen Quadrate sind mögliche Positionen für Messgeräte, dienen aber auch als Schalter für die darunterliegende Verbindung (gefüllte Kästchen stehen für geschlossene Schalter). Grüne Farbe kennzeichnet die Präsenz von Messgeräten, während rote Sammelschienen (Busse) mit rotem Abgang für die States stehen.

Mit 108 Messgeräten und 16 States ist die Sensitivitätsmatrix (108×16) dieses Netzes größer als die beiden vorhergehenden. Trotz der relativ wenigen (22) Null-Zeilen besteht die verbleibende Matrix zu 81,9% aus Null-Einträgen. Zudem gibt es 40 Klassen paarweise abhängiger Vektoren.

Netzwerk	Dimension	Dim. ohne Null-Zeilen	Spärlichkeitsfaktor	lin. Abhängigkeitsklassen	globales Minimum
NineBusSystem	54×12	54×12	64,7%	36	4
Treelike System	40×8	26×8	79,2%	12	2
MV/LV Distrib.	108×16	86×16	81,9%	40	2
Namibia	573×163	359×163	94,7%	247	1

Tabelle 3.1: Eigenschaften der Beispielprobleme. Dargestellt ist für jedes Beispiel-Stromnetz die Dimension der generierten Sensitivitätsmatrix und ihre Dimension nach dem Löschen aller Null-Zeilen. Nach dieser Löschung wurde der Spärlichkeitsfaktor, d. h. wie viele Einträge der Matrix gleich Null sind, und die Anzahl der Klassen paarweiser linear abhängiger Vektoren bestimmt. Die letzte Spalte gibt an, wie viele Einträge der Sensitivitätsmatrix gelöscht werden müssen, damit sich der Rang dieser verringert.

Namibia Als Real-World-Datensatz wurde von DIGSILENT aus dem namibischen Energieversorgungsnetz eine diskretisierte 573×164 -Sensitivitätsmatrix generiert, die also 573 Messgeräte und 164 States modelliert. Nach dem Löschen der 214 Null-Zeilen sind 94,7% der Einträge gleich Null.

Tabelle 3.1 fasst die Eigenschaften der Probleme zusammen.

3.4.1 Testergebnisse für Berechnungen von Stromnetzwerken

Der in Abschnitt 3.2 vorgestellte Algorithmus wurde mit den verschiedenen Beschleunigungstechniken aus Abschnitt 3.2.3 auf die oben beschriebenen Beispiele angewandt. Für alle Probleme wurde stets die Zeitfenstertechnik eingesetzt, um einen besseren Vergleich der einzelnen Beschleunigungen zu ermöglichen.

NineBusSystem Die beschriebene Komplexität des Beispiels (geringer Spärlichkeitsfaktor, relativ viele Äquivalenzklassen von paarweise abhängigen Vektoren) könnte der Grund dafür sein, dass über 16 Minuten Rechenzeit benötigt wurde, um das Problem ohne Beschleunigungen zu lösen.

Setzt man die Klassifizierung von paarweise abhängigen Vektoren ein, konnten alle Teilprobleme in weniger als einer Minute gelöst werden. Die Weiterverwendung von Teillösungen erwies sich als noch effektiver, mit beiden Techniken zusammen wurde die Lösung in 18 Sekunden berechnet.

Treelike System Aufgrund der erläuterten Einfachheit des Problems wurden alle Teilprobleme ohne Beschleunigungen im ersten Zeitfenster (2 sek) gelöst, wobei stets 2 als lokales Optimum gefunden wurde und damit auch das globale Minimum darstellt. Die Gesamtlaufzeit betrug weniger als eine Sekunde.

MV/LV Distribution Die Berechnung ohne Beschleunigungstechniken nimmt etwa 1,5 Minuten in Anspruch. Als lokale Optima treten die Werte 2, 4, 6, 8, 9, 12 und 49 auf. Das globale Optimum 2 wird im sechsten Teilproblem gefunden.

Bei diesem Problem konnten wie beim „NineBusSystem“-Beispiel gute Verbesserungen durch die Beschleunigungstechniken erzielt werden. Sowohl die Weiterverwendung von bereits gewonnenen Lösungen, als auch die Bildung von Äquivalenzklassen führen zu einer Laufzeit von wesentlich weniger als einer Sekunde.

Namibia Trotz des Umfangs des Problems beendet das Skript nach acht Sekunden die Berechnung, da im vierten Teilproblem festgestellt wird, dass nur eine einzelne Zeile gelöscht werden muss. Auch das zweite und dritte Teilproblem wird im Zeitfenster von zwei Sekunden gelöst und ergibt die Werte 5 und 3, während beim ersten Teilproblem in zwei Sekunden keine zulässige Lösung gefunden wird.

Versuche ohne diese spezielle Abbruchbedingung ergaben, dass nach dem zweiten Zeitfenster über die Hälfte der Teilproblem gelöst wurde, wobei fast immer das kleinste erreichbare Minimum selbst bzw. 2 oder 3 als Ergebnis vorliegen. Bei vielen der anderen Teilprobleme deutet sich an, dass ihre Berechnung wesentlich mehr Zeit in Anspruch nimmt. So konnte das Problem ohne Beschleunigungen nicht innerhalb eines Tages gelöst werden.

Um die anderen Beschleunigungstechniken zu testen, wurde der Abbruch bei Erreichen des kleinstmöglichen Minimums abgeschaltet. Während die Elimination von paarweisen linearen Abhängigkeiten, die das Problem auf eine 247×163 -Matrix reduziert, die Rechenzeit nicht auf unter einen Tag verkürzte, konnte mit der Weiterverwendung von Teillösungen die Laufzeit extrem reduziert werden. Der Grund hierfür liegt allerdings wiederum bei dem kleinem Minimum, das sehr früh gefunden wird. Beide Techniken zusammen ergaben schließlich eine Laufzeit von etwa 8 Minuten.

Tabelle 3.2 zeigt zusammenfassend die Ergebnisse bzgl. des MIP-Algorithmus.

Netzwerk	Ergebnis	Rechenzeit in Sekunden			
		einfach	ÄK	OS	beides
NineBusSystem	4	1011,26	33,36	25,97	17,41
Treelike System	2	0,07	0,07	0,06	0,04
MV/LV Distribution	2	1,26	0,41	0,18	0,14
Namibia	1	–	–	687,28	476,14

Tabelle 3.2: Experimentelle Ergebnisse mit dem MIP. Dargestellt sind die Laufzeiten mit nur der Zeitfenstertechnik („einfach“), die Zeitfenstertechnik in Kombination mit der Zusammenfassung von Äquivalenzklassen paarweiser linear abhängiger Zeilenvektoren (ÄK) und in Kombination mit der Weiterverwendung von Teillösungen als obere Schranken für folgende Teilprobleme (OS), sowie mit allen drei Techniken gleichzeitig.

Ergebnisse mit der Heuristik Auch zum Testen der Heuristik wurden die oben vorgestellten Instanzen verwendet. Jedes Beispiel wurde sowohl mit Zusammenfassung paarweise linear abhängiger Vektoren als auch ohne diese Optimierung berechnet. In zwei Beispielen lieferte der Algorithmus das exakte Ergebnisse, während er für die zwei anderen Beispiele recht stark davon abweicht, wie in Tabelle 3.3 dargestellt ist.

Zu beachten ist, dass die Laufzeit der Heuristik bei kleineren Datensätzen nicht zwingend geringer ist als für den exakt rechnenden Algorithmus, wie der folgende Abschnitt u. a. zeigt. Ein weiterer Grund für eine höhere Laufzeit ist die Größe der Lösung. Ist diese klein, so werden die MIPs des exakten Algorithmus’ durch die Weiterverwendung von Teillösungen stark eingeschränkt und ihre Lösung damit erheblich beschleunigt.

3.4.2 Testergebnisse für synthetische Daten

Um einen besseren Vergleich zwischen exakt rechnendem Algorithmus und Heuristik ziehen zu können, werden im Folgenden die Ergebnisse von Rechnungen auf synthetische Daten

Netzwerk	einfach		Äquivalenzklassen	
	Ergebnis	Rechenzeit in Sekunden	Ergebnis	Rechenzeit in Sekunden
NineBusSystem	12	1,71	12	1,17
Treelike System	2	0,10	2	0,04
MV/LV Distribution	2	0,62	2	0,14
Namibia	7	2071,77	7	1455,94

Tabelle 3.3: Experimentelle Ergebnisse mit der Heuristik. Dargestellt sind die Ergebnisse und Laufzeiten sowohl ohne („einfach“), als auch mit Zusammenfassung der Äquivalenzklassen paarweise linear abhängiger Vektoren.

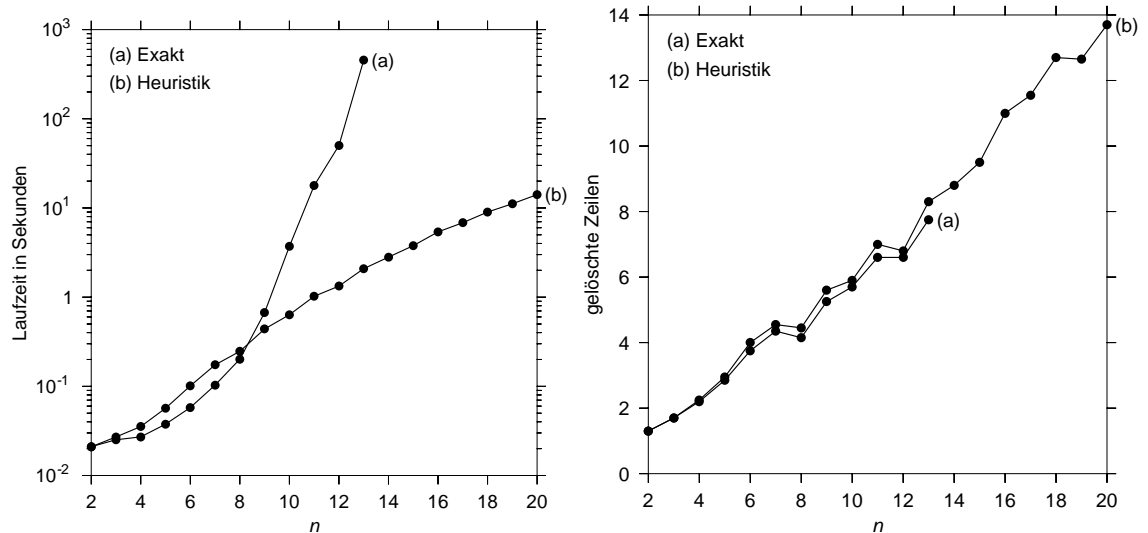


Abbildung 3.5: Laufzeit und Lösungsqualität für Zufallsmatrizen. Diese sind von der Größe $5n \times n$ mit Einträgen im Bereich von $\{-9, \dots, 9\}$ und einem Spärlichkeitsfaktor von 80%. Jeder Punkt steht für einen Durchschnitt über 20 Zufallsinstanzen.

vorgestellt. Mit den synthetischen Daten konnten eine große Menge an Beispielen gerechnet werden, was es erlaubt Durchschnittswerte u. ä. zu bilden. Um die Daten zu generieren, wurden Matrizen mit zufälligen Einträgen im Bereich von $\{-9, \dots, 9\}$ generiert, die einen Spärlichkeitsfaktor von etwa 80% besitzen. Als Dimensionen der Matrizen wurde $5 \cdot n \times n$ gewählt, um in etwa dem Verhältnis der realen Beispiele zu entsprechen. Abbildung 3.5 zeigt vergleichend die Ergebnisse zur Laufzeit und Lösungsqualität beider Algorithmen.

Es ist deutlich zu sehen, dass die Laufzeit des exakten Algorithmus im Durchschnitt exponentiell ist, während die Heuristik mit polynomialer Laufzeit rechnet und so größere Instanzen innerhalb weniger Minuten lösen kann. Außerdem zeigt sich, dass das Ergebnis der Heuristik im Durchschnitt sehr nahe am exakten liegt. Zudem sind die Laufzeiten der Heuristik besser vorherzusagen, da z. B. für $n = 13$ die Standardabweichung nur 21% beträgt im Gegensatz zum exakten Algorithmus mit 130%.

Aufgrund dessen könnte man schließen, dass es günstig wäre, stets die Heuristik zu verwenden. Allerdings betragen die Abweichungen der Heuristik vom exakten Ergebnis bis zu 50% und es lässt sich nicht vorhersagen, wie gut die Heuristik ist. Daher scheint es sinnvoll zunächst mit dem exakten Algorithmus zu rechnen, bevor man auf die Heuristik aufgrund von zu hohen Laufzeiten zurückfällt.

Die vorgestellten Algorithmen erscheinen prinzipiell brauchbar für Instanzen aus der Praxis, bei denen die Ausfallsicherheiten (der Parameter k) relativ gering sind und somit die Beschleunigungstechniken eine gute Wirkung erzielen. Zusätzlich lassen sich sowohl der exakte Algorithmus, als auch die Heuristik so erweitern, dass die für die Lösung zu löschenden Vektoren ausgegeben werden. Für den exakten Algorithmus kann dies über eine genauere Auswertung der Summe der d_i erfolgen, also für welche Indexe i die Variable d_i gleich 1 ist. Die Heuristik dagegen bestimmt die zu löschenden Vektoren sukzessive, welche somit direkt ausgegeben werden können.

3.5 Weitere Ansätze

Diskrete Matrix mit endlich vielen Werten Setzt man voraus, dass die Elemente der Vektoren nur aus einer endlichen Menge mit Kardinalität l stammen, so gibt es l^n viele Möglichkeiten für diese Vektoren. Damit ist die Anzahl der möglichen Hyperebenen aus der Problemstellung UMFASSENDSTE HYPEREBENE beschränkt durch

$$f(n) := \binom{\frac{l^n}{2}}{n-1}.$$

Diese Anzahl ergibt sich, da gegenüberliegende Vektoren sich nur um den Faktor -1 unterscheiden und somit zusammengefasst werden können. Aus den resultierenden $l^n/2$ vielen Vektoren müssen $n-1$ ausgewählt werden, die die Hyperebene definieren. Mit dieser Überlegung kann ein Algorithmus abgeleitet werden, der eine Laufzeit von $O(f(n) \cdot (nm)^{O(1)})$ besitzt, indem alle möglichen Kombinationen getestet werden.

Dieses Ergebnis ist offensichtlich nur theoretischer Natur und praktisch nicht relevant, da n eine der beiden Eingabegrößen und in der Praxis meist nicht auf kleine Werte beschränkt ist.

Kapitel 4

Verwandte Probleme

In Abschnitt 1.5 und Kapitel 2 wurden eine Reihe von Problemen vorgestellt, die äquivalent zu MATRIXROBUSTHEIT sind. Ausgehend davon werden in diesem Kapitel Beziehungen weiterer Problemstellungen zu MATRIXROBUSTHEIT betrachtet. Zunächst wird in Abschnitt 4.1 gezeigt, wie die Problemstellung des KANTENZUSAMMENHANGS auf Graphen auf MATRIXROBUSTHEIT reduziert werden kann. Weiterhin lässt sich MATRIXROBUSTHEIT auf DOMINATING SET reduzieren, wie im darauf folgenden Abschnitt 4.2 gezeigt wird.

In Abschnitt 4.3 wird zunächst die Reduktion von MINIMUM CIRCUIT auf NULLRAUM erläutert. Anschließend wird ein Fehler in der Reduktion von CLIQUE auf MINIMUM CIRCUIT aus [McC83] aufgezeigt. Diese Reduktion diente in [McC83] als Beweis der NP -Härte von NULLRAUM und hätte als NP -Härtebeweis für MATRIXROBUSTHEIT auf unendlichen Körpern verwendet werden können.

4.1 Kantenzusammenhang

Um die NP -Härte des Problems der MATRIXROBUSTHEIT zu zeigen, könnte der folgende Ansatz verwendet werden: Ein Graph wird als Matrix modelliert, wobei gelten soll, dass die Position einer Zeile in der Matrix vernachlässigbar ist, da diese Positionen bei Rangberechnungen nicht beachtet werden. Anschließend wird ein NP -hartes Graphenproblem, passend zu der Graph-Matrix-Abbildung gesucht, um die NP -Härte von MATRIXROBUSTHEIT zu beweisen. Jedoch konnte im Zuge dieser Arbeit lediglich die Reduktion des in Polynomialzeit berechenbaren [CLRS01] Problems KANTENZUSAMMENHANG auf MATRIXROBUSTHEIT gezeigt werden.

Um das Problem KANTENZUSAMMENHANG zu definieren, werden zunächst einige Begriffe aus der Graphentheorie eingeführt.

Definition 4.1 *Ein (ungerichteter) Graph $G = (V, E)$ besteht aus einer Menge V von Knoten und einer Menge E von Kanten, welche 2-elementige Teilmengen von V sind. Eine Kante $\{x, y\}$ verbindet die Knoten x und y . Ein Weg in einem solchen Graphen ist eine Folge von Knoten, wobei stets zwei aufeinander folgende Knoten durch eine Kante verbunden sind. Ein Graph heißt verbunden, wenn es für jedes Paar von Knoten einen Weg gibt, der an dem einen Knoten beginnt und an dem anderen endet.*

Eine Teilmenge U von E eines Graphen $G = (V, E)$ heißt trennend, wenn der Graph nicht mehr verbunden ist, nachdem die Kanten aus U gelöscht wurden.

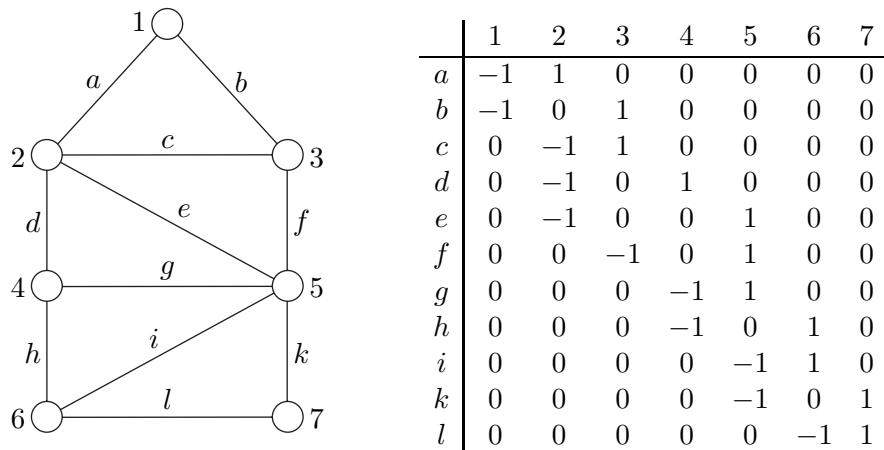


Abbildung 4.1: Beispiel einer Kantenmatrix eines Graphen

Das Problem des Kantenzusammenhangs ist nun wie folgt definiert. Zu einem gegebenen zusammenhängenden Graphen $G = (V, E)$ wird die Kantenzusammenhangszahl $\kappa'(G)$ gesucht:

$$\kappa'(G) := \min\{|F| \mid F \subseteq E \text{ ist trennende Kantenmenge in } G\}$$

Das zugehörige Entscheidungsproblem lautet:

Problem: KANTENZUSAMMENHANG

Instanz: Gegeben ist ein zusammenhängender Graph $G = (V, E)$ und eine natürliche Zahl k .

Frage: Gibt es eine trennende Kantenmenge $F \subseteq E$ in G mit $|F| \leq k$?

Zunächst wird in diesem Abschnitt die Abbildung zwischen Graph und Matrix definiert und gezeigt, dass sie eine Many-One-Reduktion von KANTENZUSAMMENHANG auf ELIMINATION VON LINEARKOMBINATIONEN ist.

Definition 4.2

Es sei ein Graph $G = (V, E)$ gegeben. Eine $|E| \times |V|$ -Matrix M wird Kantenmatrix genannt, wenn sie auf folgende Art und Weise gebildet wird. Jedem Knoten wird eine eindeutige Zahl aus $\{1, \dots, n\}$ zugeordnet, und für jede Kante $e \in E$ existiert genau eine Zeile in M , die -1 an der Position des Endknotens mit der kleineren Nummer, 1 an der Position des Endknotens mit der größeren Nummer und 0 an allen anderen Stellen enthält. Die hier definierte Abbildung kann offensichtlich in $O(|E| \cdot |V|)$ bei geeigneter Graph-Repräsentation berechnet werden.

Das Beispiel in Abbildung 4.1 veranschaulicht diese Definition.

Um einen Zusammenhang zwischen Linearkombinationen von Vektoren und des Zusammenhangs in Graphen zu gewinnen, werden sogenannte Flüsse in Graphen betrachtet.

Definition 4.3 Für einen Fluss $f : V \times V \rightarrow \mathbb{R}^+$ zwischen zwei Knoten x und y in einem Graphen $G = (V, E)$ gilt das Flusserhaltungsgesetz

$$\forall w \in V \setminus \{x, y\} : \sum_{v \neq w} f(v, w) = 0 \quad (4.1)$$

und die Eigenschaft

$$\forall p, q \in V : f(p, q) = -f(q, p). \quad (4.2)$$

Außerdem muss ein Fluss zulässig sein, d. h. für eine gegebene Funktion $c : E \rightarrow \mathbb{R} \cup \{\infty\}$ die Bedingung

$$\forall p, q \in V : 0 \leq |f(p, q)| \leq c(\{p, q\})$$

erfüllen. Der Wert $c(e)$ einer Kante e wird auch Kapazität von e genannt. Zu einem solchen Fluss wird der sogenannte Flusswert $|f|$ definiert

$$|f| := \sum_{v \neq x} f(x, v),$$

wobei aufgrund der obigen beiden Eigenschaften gilt:

$$|f| = \sum_{v \neq x} f(x, v) = \sum_{v \neq y} f(v, y) \quad (4.3)$$

Abbildung 4.2 enthält u. a. einen solchen Fluss in einem Graphen. Das folgende Lemma zeigt nun, dass bestimmte Linearkombinationen der Zeilenvektoren einer Kantenmatrix einem Fluss im zugehörigen Graphen entsprechen und umgekehrt.

Lemma 4.4

Es sei ein Graph $G = (V, E)$ gegeben. Eine Linearkombination von Zeilenvektoren aus der zugehörigen Kantenmatrix M , die wiederum einen Zeilenvektor aus M ergibt (mit -1 an Position x und 1 an Position y), repräsentiert einen Fluss f von Knoten x zu Knoten y mit dem Wert $|f| = 1$ in dem Graphen und umgekehrt. Jede Kante des Graphen habe dabei die Kapazität ∞ , Kanten des Co-Graphen, also Kanten die nicht im Graphen enthalten sind, die Kapazität 0 .

Beweis:

Betrachtet wird zunächst die Konstruktion einer Linearkombination aus einem gegebenen Fluss. Es sei mit e_{xy} der Zeilenvektor der Matrix M bezeichnet, der der Kante zwischen x und y entspricht. Aus dem Fluss f kann nun durch die Summe

$$z = \sum_{\{v,w\} \in E \wedge v < w} f(v, w) \cdot e_{vw}$$

eine Linearkombination abgeleitet werden. Für eine Komponente z^w mit $w \in V$ kann wegen des Aufbaus eines Zeilenvektors der Kantenmatrix und der festgelegten Kapazitäten die nachstehende Gleichung festgehalten werden:

$$z^w = -1 \cdot \sum_{w < v} f(w, v) + 1 \cdot \sum_{v < w} f(v, w)$$

Aufgrund der Eigenschaft (4.2) eines Flusses gilt:

$$z^w = -1 \cdot \sum_{w < v} -f(v, w) + 1 \cdot \sum_{v < w} f(v, w) = \sum_{v \neq w} f(v, w)$$

Daraus folgt $z = e_{xy}$, da gilt:

$$z^x = \sum_{v \neq x} f(v, x) = - \sum_{v \neq x} f(x, v) = -|f| = -1$$

$$z^y = \sum_{v \neq y} f(v, y) = |f| = 1$$

$$\forall w \in V - \{x, y\} : z^w = \sum_{v \neq w} f(v, w) = 0$$

Somit konnte aus dem gegebenen Fluss f zwischen x und y eine Linearkombination für e_{xy} abgeleitet werden. Um einen Fluss aus einer Linearkombination eines Zeilenvektors von M zu konstruieren, sei die Kombination durch

$$e_{xy} = \sum_{\{v,w\} \in E \wedge v < w} k_{e_{vw}} \cdot e_{vw},$$

also

$$e_{xy}^w = -1 \cdot \sum_{\{v,w\} \in E \wedge v < w} k_{e_{vw}} + 1 \cdot \sum_{\{w,v\} \in E \wedge w < v} k_{e_{wv}}$$

gegeben. Mit der Zuweisung

$$f(v, w) := \begin{cases} k_{e_{vw}} & \text{falls } \{v, w\} \in E \wedge v < w \\ -k_{e_{vw}} & \text{falls } \{v, w\} \in E \wedge v > w \\ 0 & \text{sonst} \end{cases}$$

wird ein Fluss definiert, da wegen

$$\sum_{v \neq w} f(v, w) = \sum_{\{v,w\} \in E \wedge v < w} k_{e_{vw}} - \sum_{\{v,w\} \in E \wedge v > w} k_{e_{vw}} = -e_{xy}^w$$

offensichtlich die Flusseigenschaften (4.1) und (4.3) gelten. □

Die Abbildung 4.2 verdeutlicht den Zusammenhang zwischen einem Fluss in einem Graphen und einer Linearkombination von Vektoren der zugehörigen Kantenmatrix.

Mithilfe dieser Erkenntnis, kann nun die Many-One-Reduktion von KANTENZUSAMMENHANG auf MATRIXROBUSTHEIT gezeigt werden, was mittels der zu dem letzteren Problem äquivalenten Formulierung ELIMINATION VON LINEARKOMBINATIONEN (siehe Abschnitt 1.5) getan wird.

Problem: ELIMINATION VON LINEARKOMBINATIONEN

Instanz: Eine $m \times n$ -Sensitivitätsmatrix M über einem beliebigen Körper K mit Rang n , $m \geq n$ und eine positive ganze Zahl k .

Frage: Gibt es eine Zeile, die nach dem Löschen von $k - 1$ anderen Zeilen der Sensitivitätsmatrix nicht mehr als Linearkombination der verbliebenen Vektoren dargestellt werden kann?

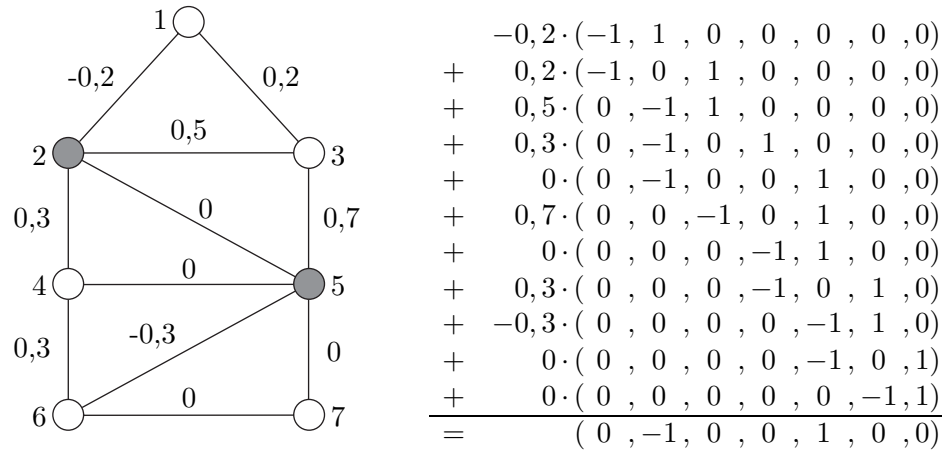


Abbildung 4.2: Beispiel zur Abbildung zwischen Fluss und Linearkombination. Auf der linken Seite ist ein Graph mit einem Fluss zwischen den Knoten 2 und 5 dargestellt, wobei immer der Flusswert $f(x, y)$ für $x < y$ an den Kanten abgebildet ist. Die rechte Seite zeigt die Linearkombination aller Vektoren e_{xy} auf Basis des Flusses, die $e_{2,5}$ ergibt.

Theorem 4.5

Das Problem KANTENZUSAMMENHANG kann in Polynomialzeit auf ELIMINATION VON LINEARKOMBINATIONEN reduziert werden.

Beweis:

O. B. d. A. werden nur Graphen ohne Schleifen betrachtet, da diese als kleine Kreise mit je einem zusätzlichen Knoten modelliert werden können.

Gibt es eine trennende Kantenmenge F in G , so betrachte eine Kante $e \in F$, deren beiden Endpunkte nach dem Löschen der Kanten aus F nicht mehr verbunden sind. Der Zeilenvektor in der Kantenmatrix M , der e repräsentiert, kann nicht als Linearkombination der anderen Zeilenvektoren aus M ohne die mittels F gelöschten Vektoren dargestellt werden. Gäbe es eine solche Linearkombination, würde aufgrund von Lemma 4.4 ein Fluss zwischen den beiden Endpunkten der Kante existieren und der Graph wäre trotz des Löschens der Kanten aus F noch verbunden. Ein Widerspruch.

Gibt es einen Vektor unter den gelöschten, der sich nun nicht mehr als Linearkombination darstellen lässt, so existiert nach Lemma 4.4 auch kein Fluss zwischen den beiden Endpunkten der entsprechenden Kanten, weshalb der Graph nicht mehr verbunden ist. \square

Aufgrund der Tatsache, dass KANTENZUSAMMENHANG in Polynomialzeit berechenbar ist, taugt es nicht, um zu beweisen, dass MATRIXROBUSTHEIT NP -hart ist. Außerdem lässt die Einfachheit der Kantenmatrix vermuten, dass eine umgekehrte Reduktion von MATRIXROBUSTHEIT auf KANTENZUSAMMENHANG nicht möglich ist und somit auch nicht gezeigt werden kann, dass MATRIXROBUSTHEIT in Polynomialzeit berechenbar ist. Trotzdem könnten die hier vorgestellten Ideen Ausgangspunkt weiterer Untersuchungen sein, mit der die Komplexität von MATRIXROBUSTHEIT für unendliche Körper aufgezeigt werden kann, in dem z. B. eine ähnliche Reduktion von einem NP -hartem Graphenproblem auf MATRIXROBUSTHEIT gefunden wird.

4.2 Dominating Set auf Split-Graphen

Ein weiterer Ansatz, die Komplexität von MATRIXROBUSTHEIT für unendliche Körper zu bestimmen, bestünde in dem Beweis, dass MATRIXROBUSTHEIT auf ein in Polynomialzeit berechenbares Problem reduzierbar ist. Auch dies wurde für diese Arbeit versucht, jedoch konnte das zu MATRIXROBUSTHEIT äquivalente Problem ELIMINATION VON LINEARKOMBINATIONEN (siehe Abschnitt 1.5) nur auf das *NP*-vollständige Problem DOMINATING SET auf sogenannten Split-Graphen reduziert werden.

Problem: DOMINATING SET

Instanz: Gegeben ist ein Graph $G = (V, E)$ und eine natürliche Zahl k .

Frage: Gibt es eine Menge von Knoten $D \subseteq V$ mit $|D| \leq k$, so dass jeder Knoten von G in D enthalten, oder er mit einem Knoten aus D verbunden ist, formal:

$$\forall v \in V : (v \in D \vee \exists d \in D : \{d, v\} \in E)?$$

Die gefundene (i. A. nicht in Polynomialzeit berechenbare) Reduktion wird im Folgenden vorgestellt. Dazu muss u. a. eine Teilmatrix von einer gegebenen Matrix definiert werden, wofür eine kompakte Schreibweise eingeführt wird.

Definition 4.6

Es sei eine $m \times n$ -Matrix gegeben. Ist C eine Menge von Zeilenindizes ($C \subseteq \{1, \dots, n\}$) oder eine Menge von Zeilen ($C \subseteq \text{ZR}(M)$ mit $\text{ZR}(M)$ als Zeilenmenge von M), sowie D eine Menge von Spaltenindizes ($D \subseteq \{1, \dots, m\}$) oder eine Menge von Spalten ($D \subseteq \text{SR}(M)$ mit $\text{SR}(M)$ als Spaltenmenge von M), so ist $M^{C,D}$ die Matrix, die nur die Zeilen aus M definiert durch C und die Spalten definiert durch D enthält. Das Zeichen \bullet kann hierbei als die Menge aller Zeilen bzw. aller Spalten verwendet werden, wodurch $M^{C,\bullet}$ die Matrix definiert, in der nur die Zeilen definiert durch C , aber alle Spalten enthält. Um einzelne Zeilen bzw. Spalten zu löschen, ist die Schreibweise C/E hilfreich, wobei E eine Menge von Indizes oder Zeilen bzw. Spalten ist. Die Matrix $M^{C/E,\bullet}$ enthält also nur die Zeilen definiert durch C außer der in E . Statt einer Menge von Zeilen bzw. Spalten kann auch direkt eine einzelne Zeile bzw. Spalte angegeben werden.

Zur Berechnung von ELIMINATION VON LINEARKOMBINATIONEN wird im Folgenden ein Algorithmus entwickelt, der sukzessive jeder Zeile der Sensitivitätsmatrix betrachtet. Zu jede Zeile analysiert er die möglichen Linearkombinationen, die diesen Vektor ergeben und eliminiert diese, in dem er aus jeder Linearkombination einen beteiligten Vektor löscht. Um die Anzahl der Linearkombinationen — wovon es potenziell unendlich viele geben kann — auf endlich viele zu beschränken, wird die nachstehende Definition verwendet.

Definition 4.7

Zwei Lösungen $x' = (c'_1, \dots, c'_n)$ und $x'' = (c''_1, \dots, c''_n)$ eines Gleichungssystems $y = M \cdot x$ heißen strukturell verschieden, wenn $\{i : c'_i \neq 0\} \neq \{i : c''_i \neq 0\}$, also ein $i \in \{1, \dots, m\}$ existiert mit $x'_i \cdot x''_i = 0$, aber $x'_i \neq 0$ und $x''_i \neq 0$.

Um ELIMINATION VON LINEARKOMBINATIONEN zu lösen, wird also für jede Zeile l der Sensitivitätsmatrix getestet, ob $k-1$ andere Zeilenvektoren aus der Matrix gelöscht werden

Input: $m \times n$ sensitivity matrix M , integer $k > 0$
Output: true or false ▷ M ist robust bzgl. k bzw. ist es nicht

```

1: function MATRIXROBUSTNESS( $M, k$ )
2:   for  $l \leftarrow 1$  to  $m$  do ▷  $y_1, \dots, y_{m-1}$  seien die Zeilen der Matrix  $M^{\bullet/l, \bullet}$ 
3:     solve  $M^{l, \bullet} = x \cdot M^{\bullet/l, \bullet}$ 
▷  $\{x_1, \dots, x_p\}$  sei die Menge der strukturell unterschiedlichen Lösungen
▷ eine Lösung sei von der Form  $x_i = \sum_{j=1, \dots, m-1} \alpha_{i,j} \cdot y_j$ 
4:     create graph  $G_l$  having  $x_1, \dots, x_p$  and  $y_1, \dots, y_{m-1}$  as its nodes
5:     for  $(j, k) \in \{1, \dots, m-1\}^2$  do
6:       connect nodes  $y_j$  and  $y_k$  in  $G_l$  ▷ alle  $y_j$  miteinander verbinden
7:     end for
8:     for  $(i, j) \in \{1, \dots, p\} \times \{1, \dots, m-1\}$  do
9:       if  $\alpha_{i,j} \neq 0$  then ▷ wenn  $y_j$  an Lösung  $x_i$  beteiligt,
10:        connect nodes  $x_i$  and  $y_j$  in  $G_l$  ▷ dann  $x_i$  mit  $y_j$  verbinden
11:      end if
12:    end for
13:    if DOMINATINGSET( $G_l, k-1$ ) return false
14:  end for
15:  return true
16: end function

```

Algorithmus 4.1: Lösung von MATRIXROBUSTHEIT mittels DOMINATING SET

können, damit l nicht als Linearkombination der verbleibenden Zeilenvektoren dargestellt werden kann. Dazu werden für jede Zeile l zunächst alle strukturell unterschiedlichen Lösungen x_1, \dots, x_p von

$$M^{l, \bullet} = x \cdot M^{\bullet/l, \bullet}$$

bestimmt, wobei die Lösungen die Gestalt

$$x_i = \sum_{j=1, \dots, m-1} \alpha_{i,j} \cdot y_j$$

besitzen. Danach wird ein Graph generiert, der die Knoten x_1, \dots, x_p und y_1, \dots, y_{m-1} besitzt. Dabei werden alle Knoten y_j miteinander verbunden und zudem die Paare (x_i, y_j) , bei denen y_j an der Lösung x_i beteiligt ist, also $\alpha_{i,j} \neq 0$. Auf diesem Graphen wird anschließend DOMINATING SET mit dem Parameter $k-1$ gelöst. Gibt es ein solches Dominating Set, so ist die Matrix nicht robust bzgl. k . Gibt es dagegen zu keiner Zeile l ein solches Dominating Set, so ist die Sensitivitätsmatrix robust bzgl. k . Dieses Vorgehen wird in Algorithmus 4.1 zusammengefasst.

Die Korrektheit von Algorithmus 4.1 beweist das folgende Theorem, da MATRIXROBUSTHEIT äquivalent zu ELIMINATION VON LINEARKOMBINATIONEN ist.

Theorem 4.8

Bricht Algorithmus 4.1 mit einer Fehlermeldung bei Betrachtung der l -ten Zeile ab, so gibt es k Zeilen (inklusive der l -ten Zeile), so dass nach deren Löschen die Sensitivitätsmatrix nicht mehr den vollen Rang besitzt. Bricht der Algorithmus nicht ab, gibt es keine k Zeilen, so dass dieser Fall eintritt. Algorithmus 4.1 löst also ELIMINATION VON LINEARKOMBINATIONEN. Somit ist MATRIXROBUSTHEIT many-one-reduzierbar auf DOMINATING SET.

Beweis:

Angenommen der Algorithmus bricht bei der Betrachtung der l -ten Zeile mit „false“ ab, d. h. es wurde ein Dominating Set D für den Graphen G_l gefunden. Enthält D einen Knoten x_i , so kann stattdessen ein benachbarter Knoten y_j gewählt werden, was das Dominating Set nicht vergrößert. Dies wird gesichert, da die Knoten y_j alle untereinander verbunden sind, während die Knoten x_i nicht miteinander verbunden sind. Daher kann statt D stets ein Connected Dominating Set betrachtet werden, also ein Dominating Set C , so dass alle Knoten von C miteinander verbunden sind.

Dieses Connected Dominating Set enthält somit für jeden Knoten x_i einen mit x_i verbundenen Knoten y_j , also für jede Gleichungssystem-Lösung x_i eine beteiligte Zeile y_j . Werden die Zeilen y_j aus der Sensitivitätsmatrix gelöscht, werden sozusagen alle Lösungen zerstört, d. h. die l -te Zeile lässt sich nicht mehr als Linearkombination der restlichen Zeilen darstellen. Wird nun noch die l -te Zeile gelöscht (insgesamt also k Zeilen), hat die Sensitivitätsmatrix nicht mehr den vollen Rang.

Angenommen es eine l -te Zeile des Sensitivitätsmatrix und $k - 1$ weitere zu löschende Zeilen, so dass sich die l -te Zeile nicht als Linearkombination der übrig gebliebenen Zeilen darstellen lässt. Dann ist an jeder Lösung des Gleichungssystems $M^{l,\bullet} = x \cdot M^{\bullet/l,\bullet}$ mindestens eine gelöschte Zeile beteiligt. Daher bildet die Menge mit den gelöschten Zeilen ein Connected Dominating Set für den Graphen des Gleichungssystems. Somit kann der Algorithmus nur erfolgreich enden, wenn das Löschen von k Zeilen den Rang der Sensitivitätsmatrix nicht verringert. \square

Die generierten Graphen fallen aufgrund der Konstruktion stets in die Klasse der Split-Graphen, die aus einem Independent Set¹ und einer Clique² bestehen. Allerdings werden nicht alle Split-Graphen erzeugt, da die Anzahl der Gleichungssystemlösungen (Independent Set) durch die Anzahl der beteiligten Lösungen (Clique) beschränkt ist. Diese Reduktion kann also nicht umgedreht werden.

Weiterhin kann laut des Beweises von Theorem 4.8 stets ein Connected Dominating Set als Lösung angenommen werden, sofern es eine gibt. Das Problem des CONNECTED DOMINATING SET ist jedoch auf Split-Graphen NP -vollständig [Ber84, CP84].

Bei der hier vorgestellten recht naiven Definition von *strukturell verschieden* müssen exponentiell viele Lösungen erwartet werden. Dazu betrachte man eine Matrix, bei der alle Vektoren paarweise linear abhängig sind. Dann ist jede Kombination von Vektoren eine Lösung des Gleichungssystems und alle Lösungen sind strukturell verschieden. Aufgrund dieser möglichen 2^m vielen Lösungen ist auch der entstehende Split-Graph exponentiell groß. Somit ist die Reduktion i. A. nicht in Polynomialzeit berechenbar. Im Falle der paarweise linear unabhängigen Vektoren könnte man jedoch z. B. nur einzelne Vektoren als Lösung betrachten, da die Kombinationslösungen durch das Löschen der einzelnen Vektoren auch zerstört würden.

Trotz der NP -Vollständigkeit des Zielproblems könnte eine geschicktere Definition von *strukturell verschieden* lohnenswert sein, wenn sich dadurch eine brauchbare Parametrisierung finden ließe, die zu einem FPT-Algorithmus führt, d. h. einem Algorithmus, der zwar nicht-polynomiale Laufzeit besitzt, der nicht-polynomiale Anteil der Laufzeit aber nur von einem bestimmten (in der Praxis möglichst kleinem) Parameter der Instanz abhängt.

¹Ein *Independent Set* ist eine Menge von Knoten, zwischen denen es keine Kanten gibt.

²Eine *Clique* ist eine Menge von Knoten, die allesamt miteinander verbunden sind.

4.3 Das Nullraum-Problem

Bei der Suche nach einer Antwort auf die Komplexitätsfrage für MATRIXROBUSTHEIT auf unendlichen Körpern wurde die Arbeit [CP86] betrachtet. Darin fand sich die Aussage, dass das zu MATRIXROBUSTHEIT äquivalente (siehe Abschnitt 2.1) Problem MINIMUM CIRCUIT NP -hart ist. Für den Beweis wird jedoch auf [McC83] verwiesen. Bei genauerer Analyse der Arbeit von McCormick stellte sich allerdings heraus, dass der Beweis der NP -Härte einen Fehler beinhaltet. Dieser Fehler wird im Abschnitt 4.3.2 aufgezeigt.

In beiden Arbeiten wurde die vermeintliche NP -Vollständigkeit von MINIMUM CIRCUIT zum Beweis der NP -Härte des NULLRAUM-Problems benutzt. Die dazu verwendete Reduktion aus [CP86] wird im folgenden Abschnitt präsentiert.

4.3.1 Die Reduktion von Minimum Circuit auf das Nullraum-Problem

Zunächst werden der Begriff Nullraum und das NULLRAUM-Problem definiert. Anschließend wird ein wenig Vorarbeit geleistet, um schließlich die Reduktion von MINIMUM CIRCUIT auf NULLRAUM zu zeigen. Die hier präsentierten Ergebnisse wurden aus [CP86] entnommen.

Definition 4.9

Der Nullraum einer $m \times n$ -Matrix M ist definiert als

$$N := N(M) := \{x \in \mathbb{R}^n : M \cdot x = 0\}.$$

Ein Vektor $x \in N(M)$ wird auch als Nullraumvektor der Matrix M bezeichnet.

Die Bestimmung einer Basis für den Nullraum einer Matrix spielt eine große Rolle in der sogenannten Nullraummethode zur Lösung von linearen Gleichungssystemen [CH97]. Dabei wird eine Zielfunktion $f(x)$ minimiert, wobei die Randbedingung $Mx = b$ gewahrt werden soll. Die Nullraummethode arbeitet in zwei Phasen: In Phase 1 wird ein zulässiger Vektor y gesucht, also ein solcher, der die Nebenbedingung $Mx = b$ erfüllt. In der zweiten Phase wird ein Nullraumvektor z gesucht, so dass $f(y + z) < f(y)$ ist. Diese Korrektur wird in einer Schleife ausgeführt, bis keine Verbesserung mehr möglich oder gewünscht ist. Um effizient zu arbeiten, ist dazu eine Basis des Nullraums nötig, die möglichst spärlich besetzt ist, also viele Nulleinträge beinhaltet. Dies führt zu dem folgenden Problem:

Problem: NULLRAUM

Instanz: Eine Matrix M über einem beliebigen Körper K .

Ziel: Finde eine am spärlichsten besetzte Matrix N , deren Spalten den Nullraum von M aufspannen.

Um eine Beziehung zwischen MINIMUM CIRCUIT — beschäftigt sich mit linearen Abhängigkeiten — und NULLRAUM zu erhalten, werden die Vektoren des Nullraums betrachtet, die eine lineare Abhängigkeit repräsentieren.

Definition 4.10

Ein Nullraumvektor $x = (x^1, \dots, x^n)$ der Matrix M mit den Spaltenvektoren s_1, \dots, s_n entspricht einer Linearkombination dieser Spaltenvektoren auf folgende Art und Weise:

$$x \text{ ist Nullraumvektor} \Leftrightarrow M \cdot x = 0 \Leftrightarrow s_1 \cdot x^1 + \dots + s_n \cdot x^n = 0$$

Gilt $x \neq 0$, so kann aus dieser Linearkombination eine Menge linear abhängiger Vektoren bestimmt werden und umgekehrt, was im Folgenden als korrespondierend bezeichnet wird. Ist diese Menge minimal (jede echte Teilmenge ist nicht linear abhängig), so wird diese in Anlehnung an die Matroidtheorie, siehe Abschnitt 2.1, Kreis von M und der korrespondierende Nullraumvektor Kreis-Nullraumvektor genannt.

Als Vorbereitung für den Beweis der NP-Härte des NULLRAUM-Problems wird im Folgenden das Konzept der konformen Dekomposition eingeführt. Dies ist eine Möglichkeit, einen Nullraumvektor durch Kreis-Nullraumvektoren darzustellen.

Definition 4.11

Zu einem Vektor x ist der Träger definiert als $S(x) := \{j \mid x^j \neq 0\}$.

Definition 4.12

Ein Vektor x ist zu einem Vektor y genau dann konform, wenn gilt:

$$x^j \neq 0 \Rightarrow \operatorname{sgn}(x^j) = \operatorname{sgn}(y^j) \quad \text{mit } \operatorname{sgn}(x^j) := \begin{cases} +1 & \text{falls } x^j > 0 \\ -1 & \text{falls } x^j < 0 \\ 0 & \text{falls } x^j = 0 \end{cases}$$

Betrachtet man beispielsweise zwei Vektoren x, y mit $\operatorname{sgn}(x) = (+0-0+0)$ und $\operatorname{sgn}(y) = (++-0+-)$, so ist x konform zu dem Vektor y , aber nicht umgekehrt. Außerdem gilt generell, dass wenn ein Vektor x konform zu einem Vektor y ist, auch $S(x) \subseteq S(y)$ gilt.

Lemma 4.13

Für jeden Nullraumvektor x einer Matrix M gibt es ein Kreis-Nullraumvektor c von M , der zu diesem konform ist.

Beweis: ([CP86, Lemma 2.3])

Angenommen, die Aussage sei falsch. Dann kann ein Nullraumvektor x mit $|S(x)|$ minimal gewählt werden, so dass es kein Kreis-Nullraumvektor von M gibt, der zu diesem konform ist. Es sei c ein Kreis-Nullraumvektor mit $S(c) \subseteq S(x)$ und

$$J = \{j \mid c^j \neq 0 \wedge \operatorname{sgn}(c^j) \neq \operatorname{sgn}(x^j)\}.$$

Die Menge J ist offensichtlich nicht leer, da ansonsten c konform zu x sein würde. Es sei weiterhin $a := \min_{j \in J} -x^j/c^j$. Damit ist $a \geq 0$ und für $z := x + a \cdot c$ gelten die Aussagen der folgenden Übersicht:

x	0	0	0	+	+	+	-	-	-
c	0	+	-	0	+	-	0	+	-
z	0	$0^{(1)}$	$0^{(1)}$	+	+	$0/+^{(2)}$	-	$0/-^{(3)}$	-

(1) $a = 0$ (2) $z < 0 \stackrel{c \leq 0}{\Rightarrow} -\frac{x}{c} < a$ Widerspruch. (3) $z < 0 \Rightarrow -\frac{x}{c} < a$ Widerspruch.

Somit ist z konform zu x und es gilt $S(z) \subset S(x)$, da für mindestens ein j gilt $z^j = 0$, wobei $x^j > 0$ oder $x^j < 0$. Aufgrund der Wahl von x gibt es einen Kreis-Nullraumvektor d , der konform zu z ist. Aber dann ist auch d konform zu x , ein Widerspruch. \square

Lemma 4.14

Jeder Nullraumvektor x einer Matrix M kann als Summe von verschiedenen Kreis-Nullraumvektoren

$$x = c_1 + \cdots + c_p$$

geschrieben werden, wobei jeder Kreis-Nullraumvektor c_i konform zu dem Vektor x ist. Diese Summe wird auch als konforme Dekomposition bezeichnet.

Beweis: ([CP86, Lemma 2.4])

Nach Lemma 4.13 gibt es einen Kreis-Nullraumvektor d_1 , der zu x konform ist. Wähle eine beliebige Position j mit $d_1^j \neq 0$ aus, anhand der dieser Vektor an x normiert wird:

$$c_1 = \frac{x^j}{d_1^j} \cdot d$$

Damit kann x nun geschrieben werden als $x = c_1 + (x - c_1)$, wobei $S(x - c_1) \subset S(x)$ ist. Dieses Verfahren wird nun induktiv fortgeführt, bis $S(x - c_1 - \cdots - c_p) = \emptyset$. \square

Das Konzept der konformen Dekomposition wird im folgenden Theorem benutzt, um die Zusammensetzung einer Nullraumbasis zu charakterisieren.

Theorem 4.15

Jeder Vektor einer am spärlichsten besetzten Matrix N , deren Spaltenvektoren eine Nullraumbasis bilden, ist ein Kreis-Nullraumvektor.

Beweis: ([CP86, Theorem 2.6])

Angenommen ein Vektor n dieser Basis sei kein Kreis-Nullraumvektor. Dann kann nach Lemma 4.14 dieser in eine Summe von Kreis-Nullraumvektoren $n = c_1 + \cdots + c_p$ zerlegt werden. Da n linear unabhängig von den anderen Vektoren aus der Basis ist, muss einer der Kreis-Nullraumvektoren c_i aus der Summe es ebenso sein. Da c_i ein Kreis-Nullraumvektor ist, gilt $S(c_i) \subset S(n)$. Somit ist c_i spärlicher als n und kann n in der Basis ersetzen. Die entstehende Basis wäre spärlicher als die ursprüngliche, ein Widerspruch. \square

Diese Charakterisierung sollte die Grundlage für den Beweis der NP -Härte des NULLRAUM-Problems liefern. Aufgrund der Tatsache, dass der dafür notwendige Beweis aus [McC83] der NP -Härte von MINIMUM CIRCUIT nicht korrekt ist, kann nicht gezeigt werden, dass NULLRAUM auch für unendliche Räume NP -Hart ist. Jedoch kann die folgende Reduktion vom NULLRAUM-Problem auf MINIMUM CIRCUIT verwendet werden, um die NP -Härte des NULLRAUM-Problems für endliche Körper zu zeigen.

Theorem 4.16

Das NULLRAUM-Problem für endliche Körper ist NP -hart.

Beweis: (in Anlehnung an [CP86, Theorem 3.2])

Nach Theorem 4.15 sind alle Vektoren in einer am spärlichsten besetzten Nullraumbasis für eine Matrix M Kreis-Nullraumvektoren. Damit ist auch einer der am spärlichsten besetzten Kreis-Nullraumvektoren in dieser Basis enthalten. Der korrespondierende Kreis

Input: $m \times n$ matrix M having rank r

Output: $m \times n - r$ matrix N , such that the rows of N span the nullspace of M

```
1: function NULLSPACE( $M$ )
2:    $N \leftarrow \emptyset$ 
3:   for  $i \leftarrow 1$  to  $n - r$  do
4:     find a sparsest nullvector  $n_i$ , such that  $\text{RANK}(n_1, \dots, n_i) = i$ 
5:      $N \leftarrow N \cup \{n_i\}$ 
6:   end for
7: end function
```

Algorithmus 4.2: Greedy-Algorithmus für das NULLRAUM-Problem

braucht lediglich auf die Anzahl seiner Nicht-Null-Einträge geprüft und mit k verglichen werden, um das Problem MINIMUM CIRCUIT mit der Matrix M und dem Parameter k als Eingabe zu lösen. Da MINIMUM CIRCUIT für endliche Körper NP -vollständig ist, muss demzufolge das NULLRAUM-Problem für endliche Körper NP -hart sein. \square

Bemerkung 4.17

In [CP86] wird weiterhin gezeigt, dass der Greedy-Algorithmus 4.2 das NULLRAUM-Problem löst.

Allerdings zeigt der Beweis von Theorem 4.16, dass das Finden eines am spärlichsten besetzten Nullraumvektors (Schritt 4 im Algorithmus) dem Lösen von MINIMUM CIRCUIT gleichkommt. Damit besitzt Algorithmus 4.2 die Laufzeit $O((n-r) \cdot f(n, m))$, wobei $f(n, m)$ die Laufzeit von MINIMUM CIRCUIT ist.

4.3.2 Fehlerhafte Reduktion von Clique auf Minimum Circuit

Um die NP -Härte des NULLRAUM-Problems zu zeigen, berufen sich sowohl [CP86] als auch [McC83] auf die NP -Härte von MINIMUM CIRCUIT, die in [McC83] durch eine Reduktion von CLIQUE bewiesen werden soll.

Problem: CLIQUE

Instanz: Ein ungerichteter Graph G und eine positive ganze Zahl m .

Frage: Besitzt G eine Clique der Größe m ?

Der in [McC83] zu dieser Reduktion vorgestellte Beweis lautet wie folgt, wobei mit MCP das Problem MINIMUM CIRCUIT und mit NSP das NULLRAUM-Problem bezeichnet wird.

Behauptung 4.18 (Theorem 3.3.6 aus [McC83])

MCP is NP-Complete, and thus NSP is NP-hard.

Beweis:

Given a graph G with v vertices and e edges, construct a $(v + \binom{m}{2} - m - 1) \times e$ sparsity pattern $A(G)$ as follows. [Anm.: Ein Sparsity Pattern ist eine Matrix mit 0 und \times als Einträge, wobei das Zeichen \times für einen beliebigen Nicht-Null-Eintrag steht.] Index the first v rows of $A(G)$ with the vertices of G , and its columns with the edges of G . Make

$a_{i,\{j,k\}} \neq 0$ when $i = j$ or $i = k$, so that the first v rows of $A(G)$ have the same sparsity pattern as the vertex-edge incidence matrix of G . For row $i > v$, set $a_{ie} \neq 0$ for all e .

Suppose that $A(G)$ has a circuit C with $|C| = c < \binom{m}{2}$. Let $d = \binom{m}{2} - c$, so that $d > 0$. Since C is a circuit, $A_{\bullet C}$ [Anm.: in dieser Arbeit als $A^{\bullet, C}$ geschrieben] has $c - 1$ non-zero rows. Because the $\binom{m}{2} - m - 1$ non-vertex rows are among these, $A_{\bullet C}$ has $(c - 1) - (\binom{m}{2} - m - 1) = m - d$ non-zero vertex rows. Denote the set of such rows by R , and denote by f the number of edges in the subgraph of G induced by the vertices in R . Since $|R| = m - d$ vertices can induce at most $\binom{m-d}{2}$ edges, $f \leq \binom{m-d}{2}$. But surely all the edges in C are among those induced by R , so that $f \geq c = \binom{m}{2} - d$. Putting these inequalities together yields $\binom{m}{2} - d \leq f \leq \binom{m-d}{2}$. [Anm.: Die Variable e muss in dieser Formel durch f ersetzt werden.] However, it is easy to show that when $m > 3$ (which can be assumed without loss of generality), $\binom{m}{2} - d > \binom{m-d}{2}$.

Thus every circuit C of $A(G)$ must satisfy $|C| \geq \binom{m}{2}$. Suppose that G has an m -clique, say on the vertices in the set R (so that $|R| = m$), and with the $\binom{m}{2}$ edges in the set C . Since $|R| + (\text{number of non-vertex rows}) = \binom{m}{2} - 1 = |C| - 1$, it is easy to verify that C is a circuit.

Conversely, suppose that $A(G)$ has a circuit of size $\binom{m}{2}$, and let R be the m non-zero vertex rows in $A_{\bullet C}$. Let f again be the number of edges in the subgraph of G induced by R . As above, it must be true that $f \geq |C| = \binom{m}{2}$. But m vertices can induce at most $\binom{m}{2}$ edges, implying that $f \leq \binom{m}{2}$. Hence $f = \binom{m}{2}$, and the vertices in R are an m -clique.

Thus G has an m -clique if and only if the girth of $A(G)$ is $\binom{m}{2}$. If there were a polynomial algorithm for MCP it could be used to determine the girth of $A(G)$, and thereby determine whether G has an m -clique. But the m -Clique Problem is NP -Complete, and so MCP must also be NP -Complete. Therefore solving NSP is NP -Hard, since (as shown above), solving NSP also solves MCP. \square

Beispiel 4.19 (Gegenbeispiel zu Behauptung 4.18)

Betrachte den Graphen G mit $v = 4$ Knoten und $e = 6$ Kanten in Abbildung 4.3. Der Parameter m sei auf 4 gesetzt.

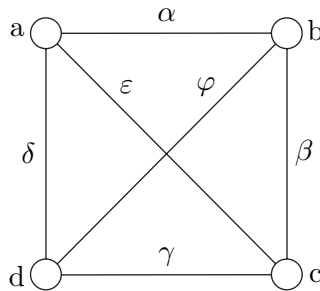


Abbildung 4.3: Graph für das Gegenbeispiel zu McCormicks NP -Härtebeweis für MINIMUM CIRCUIT

Daraus wird die im Beweis von Behauptung 4.18 beschriebene Konstruktion eines Spärlichkeitsmusters durchgeführt. In Abbildung 4.4 ist sowohl das Ergebnis, als auch (eine mögliche) Ausprägung als Matrix gegeben.

	α	β	γ	δ	ε	φ
a	\times	0	0	\times	\times	0
b	\times	\times	0	0	0	\times
c	0	\times	\times	0	\times	0
d	0	0	\times	\times	0	\times
	\times	\times	\times	\times	\times	\times

Spärlichkeitsmuster

	α	β	γ	δ	ε	φ
a	-1	0	0	-1	-1	0
b	1	-1	0	0	0	-1
c	0	1	-1	0	1	0
d	0	0	1	1	0	1
	1	-1	1	1	1	-1

Matrix

Abbildung 4.4: Spärlichkeitsmuster und Matrix für das Gegenbeispiel zu McCormicks NP -Härtebeweis für MINIMUM CIRCUIT

Im zweiten Absatz des Beweises wird ein Kreis C mit $|C| < \binom{m}{2}$ angenommen und diese Annahme widerlegt. Das obige Beispiel liefert jedoch den Kreis $C := \{\alpha, \beta, \gamma, \delta\}$ mit $c := |C| = 4 < \binom{m}{2} = 6$. Der Beweis führt die Variable $d = \binom{m}{2} - c$ ein. Hier gilt also $d = 6 - 4 = 2 > 0$. Da C ein Kreis ist, wird geschlossen, dass die Matrix

$$A^{\bullet, C} = \begin{pmatrix} -1 & 0 & 0 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 1 & 1 \end{pmatrix}$$

$c - 1 = 3$ Nicht-Nullzeilen besitzt. Dies gilt, da jede $(c - 1)$ -elementige Teilmenge von C linear unabhängig ist. Damit ist der Rang der durch diese Menge induzierten Matrix gleich $c - 1$. Um dies zu sichern, müssen auch $c - 1$ Nicht-Nullzeilen in der induzierten Matrix vorhanden sein und somit auch in $A^{\bullet, C}$. Anschließend müssen jedoch die zusätzlichen $\binom{m}{2} - m - 1 = 1$ Zeile(n) abgezogen werden, da diese in $A^{\bullet, C}$ vorhanden sind. Somit kann gefolgert werden, dass es mindestens $(c - 1) - (\binom{m}{2} - m - 1) = m - d = 2$ Nicht-Nullzeilen gibt, die Knoten repräsentieren. Die Menge dieser Zeilen wird mit R bezeichnet, wobei hier $R \subseteq \{a, b, c, d\}$ mit $|R| = 2$ gewählt werden kann. Die Variable f soll die Anzahl der Kanten in dem durch die Knoten aus R induzierten Graph bezeichnen. Also ist im Beispiel $f = 1$, da es nur zwei Knoten in R gibt, die alle miteinander verbunden sind. Auch die nächste Schlussfolgerung ist korrekt, da $|R| = m - d$ gilt, können höchstens $\binom{m-d}{2}$ Kanten induziert werden, in diesem Fall $f \leq 1$.

Jedoch sind nicht alle Kanten aus C unter denen, die durch R induziert werden, was klar im Beispiel zu erkennen ist. Dies kann allgemein nicht geschlossen werden, da in R i. A. weniger Knoten enthalten sind, als durch die Kanten aus C induziert wird. Zusammenfassend lässt sich also sagen, dass nicht angenommen werden kann, dass jeder Kreis mindestens $\binom{m}{2}$ Vektoren umfasst. Der Beweis kann daher so nicht fortgeführt werden.

Wegen des Fehlers in dem Beweis von Behauptung 4.18 ist die Komplexität des NULLRAUM-Problems für unendliche Körper weiterhin offen. Lediglich für endliche Körper kann die NP -Vollständigkeit von MINIMUM DISTANCE (siehe Abschnitt 2.3) aufgrund der Äquivalenz zu MINIMUM CIRCUIT (siehe Abschnitt 2.2) übertragen werden.

Allerdings liefert die Reduktion von NULLRAUM auf MINIMUM CIRCUIT einen weiteren Grund, die Komplexität von MATRIXROBUSTHEIT für unendliche Körper zu untersuchen, da für NULLRAUM auch hauptsächlich unendliche Körper interessant sind.

Kapitel 5

Zusammenfassung und Ausblick

Zusammenfassung Untersucht wurde die Komplexität des Problems MATRIXROBUSTHEIT. Dieses wurde unter Verwendung der Sensitivitätsmatrix, wie sie in [AP05] vorgestellt wurde, aus einer elektrotechnischen Problemstellung abgeleitet. Dabei soll geprüft werden, ob eine zu einem gewissen Grad ausfallsichere Verteilung von Messgeräten in einem gegebenen Stromnetz vorliegt.

MATRIXROBUSTHEIT wurde mit einer Reihe von Problemen in Beziehung gesetzt, wie in Abbildung 1.3 zu sehen ist. Komplexitätstheoretisch am interessantesten sind dabei die Äquivalenzen zu MINIMUM CIRCUIT und MINIMUM DISTANCE, wobei MINIMUM DISTANCE im Sinne von der Verwendung unendlicher Körper etwas allgemeiner aufgefasst wurde, als dies i. A. getan wird. Das ist auch der Grund, warum die Ergebnisse zu Komplexitätsbetrachtungen von MINIMUM DISTANCE zunächst nur für MATRIXROBUSTHEIT mit endlichen Körpern übertragbar sind. Die Frage nach der Komplexität von MATRIXROBUSTHEIT (und den äquivalenten Problemen) für unendliche Körper bleibt offen.

Jedoch motivieren die hier vorgestellten Zusammenhänge eine erweiterte Betrachtung von MINIMUM DISTANCE oder der äquivalenten Problemstellungen, da sowohl die Komplexität von MATRIXROBUSTHEIT, als auch von dem NULLRAUM-Problem für unendliche Körper bestimmt werden kann. Die Vielfalt der hier vorgestellten äquivalenten Probleme bieten viele Ansatzpunkte für weitere Untersuchungen.

In einer Arbeit von McCormick [McC83] wurde ein vermeintlicher *NP*-Vollständigkeitsbeweis für MINIMUM CIRCUIT gefunden, der die Komplexität von MATRIXROBUSTHEIT auch für den Fall den unendlichen Körper hätte bestimmen können. Das dort vorgestellte Ergebnis konnte jedoch nicht bestätigt werden, vielmehr wurde ein Fehler gefunden, der in Abschnitt 4.3.2 vorgestellt wurde.

Neben den bisher genannten Problemstellungen konnte MATRIXROBUSTHEIT weiterhin mit den Graphentheorie-Problemen KANTENZUSAMMENHANG, sowie DOMINATING SET in Beziehung gesetzt werden, woraus evtl. weitere Ansatzpunkte zur Untersuchung der Komplexität von MATRIXROBUSTHEIT gewonnen werden können.

Aufgrund der auch für unendliche Körper zu vermutenden *NP*-Vollständigkeit von MATRIXROBUSTHEIT wurde — basierend auf einer weiteren Problemformulierung — ein Mixed Integer Linear Program geschrieben, welches MATRIXROBUSTHEIT exakt löst. Zusätzlich wurde eine Polynomialzeit-Heuristik präsentiert, die allerdings keine exakte Lösung bestimmt. In die Algorithmen wurden diverse Verbesserungen implementiert, welche dazu

fürten, dass die von der Firma DIGSILENT GmbH bereitgestellten Beispiele in annehmbarer Zeit gelöst werden konnten.

Ausblick Wie bereits erwähnt, blieb eine Antwort auf die Komplexitätsfrage zur MATRIXROBUSTHEIT auf unendlichen Körpern aus. Aufgrund der Äquivalenzen zu verschiedenen wichtigen Problemen erscheint eine weitere, tiefergehende Untersuchung jedoch lohnenswert. Neben einem möglichen Beweis der NP -Härte von MATRIXROBUSTHEIT könnte die $W[1]$ -Härte dessen gezeigt werden. Ein Problem ist $W[1]$ -hart, wenn es keinen FPT-Algorithmus (fixed parameter tractable) gibt. Ein solcher Algorithmus besitzt als Teil der Eingabe Parameter, wobei die Laufzeit des Algorithmus prinzipiell nicht polynomial sein muss. Der nicht polynomiale Anteil daran darf jedoch ausschließlich von den Parametern abhängen. Wenn in der Praxis die entsprechenden Parameter stets klein sind oder nur langsam mit der Größe der Probleminstanz anwachsen, kann ein entsprechender FPT-Algorithmus effizient zur Lösung des Problems eingesetzt werden.

Betrachtet man speziell MATRIXROBUSTHEIT und die zugrunde liegende Sensitivitätsmatrix, so könnte die Analyse und Ausnutzung der Eigenschaften dieser Matrix evtl. bessere als die in dieser Arbeit vorgestellten Algorithmen liefern. Ein Ansatzpunkt hierbei wäre eine Blockstruktur der Sensitivitätsmatrix der folgenden Art, welche in den aus elektrischen Netzwerken gewonnenen Beispielen oft in Ansätzen erkennbar war.

$$\begin{pmatrix} S_1 & & & 0 \\ & S_2 & & \\ & & \ddots & \\ 0 & & & S_l \end{pmatrix}$$

Sind S_1, \dots, S_l wiederum Matrizen, so könnte der in der vorliegende Arbeit vorgestellte MIP-Algorithmus auf diese angewendet und das Minimum über alle Lösungen als endgültiges Ergebnis bestimmt werden. Dies wäre möglich, da es keine blockübergreifenden linearen Abhängigkeiten gibt. Damit wäre ein FPT-Algorithmus mit der Blockgröße dieser Matrizen als Parameter denkbar. Weiterhin könnte untersucht werden, inwieweit die Verarbeitung von blockübergreifenden Zeilen möglich wäre, um den Algorithmus auch auf allgemeinere Probleminstanzen anzuwenden.

Die ursprüngliche Idee hinter dieser Arbeit bestand in der Untersuchung des Problems der optimalen Messgeräteverteilung in einem Netz, welches auch als Optimal Meter Placement (OMP) bekannt ist und in dieser Arbeit in Bezug auf die Sensitivitätsmatrix als das Problem MINIMALE ROBUSTE SUBMATRIX definiert wurde. In der in Abschnitt 1.3 referenzierten Literatur gibt es dazu bereits mehrere Ansätze, wobei einige auch einfache Messgeräteausfall-Szenarien betrachten. Fortführende Arbeiten könnten bekannte Algorithmen für andere Observierbarkeitsmodelle als Basis zur Lösung von MINIMALE ROBUSTE SUBMATRIX verwenden, neue entwickeln oder Komplexitätsbetrachtungen dazu durchführen. Außerdem wäre OMP mit Absicherung gegenüber komplexeren Ausfallszenarien ein interessantes Forschungsfeld.

Bei der Betrachtung von OMP oder der Observability Analysis sind weiterhin Ansätze jenseits der Sensitivitätsmatrix möglich, wie auch die Literatur zeigt (siehe Abschnitt 1.3). Ein evtl. möglicher Ansatz wäre eine graphentheoretische Modellierung der Überwachbarkeit, welche genau wie das Sensitivitätsmatrix-Modell nicht auf wenige Messgerätetypen beschränkt ist.

Literaturverzeichnis

- [AM01] Ali Abur und Fernando H. Magnago. Optimal meter placement against contingencies. In *Power Engineering Society Summer Meeting*, Band 1, Seiten 424–428. IEEE, Juli 2001. 3
- [Ant04] Howard Anton. *Lineare Algebra*. Spektrum, 3. Auflage, 2004. 10, 18, 20
- [AP05] Jochen Alber und Markus Pöller. Observability of power systems based on fast pseudorank calculation of sparse sensitivity matrices. IEEE PES Transmission and Distribution Conference, New Orleans, Oktober 2005. 3, 6, 35, 44, 65
- [ATF01] Alessandra B. Antonio, José R. A. Torreão und Milton Brown Do Coutto Filho. Meter placement for power system state estimation using simulated annealing. In *Porto Power Tech Conference Proceedings*, Band 3, Seite 5 pp., Porto, Portugal, September 2001. IEEE. 3
- [Ber84] Alan A. Bertossi. Dominating sets for split and bipartite graphs. *Information Processing Letters*, 19(1):37–40, Juli 1984. 58
- [BL98] Newton Geraldo Bretas und João Bosco Augusto London, Jr. Network observability: The critical measurement identification using the symbolic Jacobian matrix. In *International Conference on Power System Technology, POWERCON '98*, Band 2, Seiten 1222–1226, August 1998. 6
- [BLAB05] Newton Geraldo Bretas, João Bosco Augusto London, Jr., Luís F. C. Alberto und Arturo S. Bretas. A topological approach to the identification of critical measurements in power-system state estimation. *IEEE Transactions on Circuits and Systems*, 52(1):139–147, Januar 2005. 5
- [BMvT78] Elwyn R. Berlekamp, Robert J. McEliece und Henk C. A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, IT-24:384–386, 1978. 33
- [Bre96] Newton Geraldo Bretas. Network observability: Theory and algorithms based on triangular factorisation and path graph concepts. *IEE Proceedings of Generation, Transmission and Distribution*, 143(1):123–128, Januar 1996. 5
- [BZZG95] Mesut E. Baran, Jinxiang Zhu, Hongbo Zhu und Kenneth E. Garren. A meter placement method for state estimation. *IEEE Transactions on Power Systems*, 10(3):1704–1710, August 1995. 3

- [CH97] Anthony J. Cox und Nicholas J. Higham. Accuracy and stability of the null space method for solving the equality constrained least squares problem. Numerical Analysis Report 306, Manchester Centre for Computational Mathematics, August 1997. 59
- [Che90] Rong-Liang Chen. A fast integer algorithm for observability analysis using network topology. *IEEE Transactions on Power Systems*, 5(3):1001–1009, August 1990. 6
- [Cle90] Kevin A. Clements. Observability methods and optimal meter placement. *Electrical Power & Energy Systems*, 12(2):88–93, April 1990. 2
- [CLRS01] Thomas H. Cormen, Charles Leiserson, Ronald L. Rivest und Clifford Stein. *Introduction to Algorithms*. MIT Press, 2. Auflage, 2001. 38, 51
- [CP84] Derek G. Corneil und Yehoshua Perl. Clustering and domination in perfect graphs. *Discrete Applied Mathematics*, 9(1):27–39, September 1984. 58
- [CP86] Thomas F. Coleman und Alex Pothén. The null space problem I. Complexity. *Journal of Algebraic Discrete Methods*, 7(4):527–536, Oktober 1986. 4, 59, 60, 61, 62
- [DA02] Qifeng Ding und Ali Abur. An improved measurement placement method against loss of multiple measurements and branches. In *Power Engineering Society Winter Meeting*, Band 1, Seiten 234–237. IEEE, Januar 2002. 3
- [Din04] Qifeng Ding. *Optimal Meter Placement and Transaction-based Loss Allocation in Deregulated Power System Operation*. Doktorarbeit, Texas A&M University, Dezember 2004. 3
- [DMS03] Ilya Dumer, Daniele Micciancio und Madhu Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Transactions on Information Theory*, 49(1):22–37, 2003. 33
- [EZ04] Amany El-Zonkoly. A novel algorithm for optimal meter placement to maintain network observability. In *Power Systems Conference and Exposition*, Band 1, Seiten 93–99. IEEE PES, Oktober 2004. 3
- [Fou05] Robert Fourer. Software survey: Linear programming. *Operation Research/Management Science Today*, (3), Juni 2005. 38
- [Gou05] Bei Gou. Observability analysis by measurement Jacobian matrix for state estimation. In *Proceedings of the International Symposium on Circuits and Systems*, Band 4, Seiten 3881–3884. IEEE, Mai 2005. 5
- [HI01] Ibrahim O. Habiballah und Malcolm R. Irving. Observability analysis for state estimation using linear programming. *IEE Proceedings of Generation, Transmission and Distribution*, 148:142–145, März 2001. 5
- [HLA03] Garng M. Huang, Jiansheng Lei und Ali Abur. A heuristic approach for power system measurement placement design. In *Proceedings of the International Symposium on Circuits and Systems*, Band 5, Seiten 407–410. IEEE, Mai 2003. 3

-
- [Hod67] Joseph Lawson Hodges Jr. Efficiency in normal samples and tolerance of extreme values for some estimates of location. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1:163–168, 1967. 2
- [JBT⁺05] Amit Jain, Ravi Balasubramanian, S. C. Tripathy, Brij N. Singh und Yoshiyuki Kawazoe. Power system topological observability analysis using artificial neural networks. In *Power Engineering Society General Meeting*, Seiten 2687–2692. IEEE, Juni 2005. 5
- [JKBT03] Amit Jain, Yoshiyuki Kawazoe, Ravi Balasubramanian und S. C. Tripathy. Network observability: A solution technique using neural networks. In *Conference on Convergent Technologies for Asia-Pacific Region, TENCON*, Band 3, Seiten 1007–1011, Oktober 2003. 5
- [Kar84] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–396, 1984. 38
- [Kha79] Leonid G. Khachiyan. A polynomial algorithm in linear programming. In *Doklady Akedamii Nauk SSSR*, 244, Seiten 1093–1096, 1979. in Russisch. 38
- [KK03] George N. Korres und Peter J. Katsikas. A hybrid method for observability analysis using a reduced network graph theory. *IEEE Transactions on Power Systems*, 18(1):295–304, Februar 2003. 5
- [KKCD03] George N. Korres, Peter J. Katsikas, Kevin A. Clements und Paul W. Davis. Numerical observability analysis based on network graph theory. *IEEE Transactions on Power Systems*, 18(3):1035–1044, August 2003. 5
- [LX04] San Ling und Chaoping Xing. *Coding Theory: A First Course*. Cambridge University Press, Cambridge, 2004. 24, 27, 28, 29, 31, 32
- [MA00] Fernando H. Magnago und Ali Abur. A unified approach to robust meter placement against loss of measurements and branch outages. *IEEE Transactions on Power Systems*, 15(3):945–949, August 2000. 3
- [Mak05] Andrew Makhorin. *GNU Linear Programming Kit: Modelling Language GNU MathProg*. Free Software Foundation, 4.8. Auflage, Januar 2005. Entwurf. 40
- [McC83] S. Thomas McCormick. *A Combinatorial Approach to Some Sparse Matrix Problems*. Dissertation, Stanford University, Stanford, California, Juni 1983. Erschienen in Systems Optimization Laboratory 83-5. 4, 51, 59, 61, 62, 65
- [MCR94] Lamine Mili, Michael G. Cheniae und Peter J. Rousseeuw. Robust state estimation of electric power systems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 41(5):349–358, Mai 1994. 2
- [MT91] Hiroyuki Mori und Senji Tsuzuki. A fast method for topological observability analysis using a minimum spanning tree technique. *IEEE Transactions on Power Systems*, 6(2):491–500, Mai 1991. 5
- [Oxl04] James G. Oxley. *Matroid Theory*. Oxford University Press Inc., New York, 2004. 10, 12, 13, 14, 15, 16, 17, 18, 22

- [PMCK88] Young Moon Park, Young Hyun Moon, Jin Boo Choo und Tae Won Kwon. Design of reliable measurement system for state estimation. *IEEE Transactions on Power Systems*, 3(3):830–836, August 1988. 3
- [Sha05] Yakov Shafranovich. *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. Internet Engineering Task Force, Oktober 2005. RFC 4180. 40
- [Var97] Alexander Vardy. The intractability of computing the minimum distance of a code. *IEEE Transactions on Information Theory*, 43(6):1757–1766, November 1997. 33
- [vR05] Guido van Rossum. *Python Tutorial*. Python Software Foundation, 2.4.1. Auflage, März 2005. 40
- [Wel76] Dominic J. A. Welsh. *Matroid Theory*. Academic Press Inc. (London) Ltd., 1976. 10, 16
- [Wik05] Wikipedia. Online, Oktober 2005. <http://de.wikipedia.org/>. 1

Danksagung

Mein Dank geht an Prof. Rolf Niedermeier, Dr. Jochen Alber und Falk Hüffner für die Bereitstellung dieses eher gewagten, interessanten Themas und der guten Betreuung während der Anfertigung der Diplomarbeit. Jochen und allen, die mich dabei organisatorisch und finanziell unterstützt haben, danke ich im Speziellen für den interessanten Besuch in Gomaringen.

Geoff Whittle von der Victoria University of Wellington danke ich für den entscheidenden Hinweis für Kapitel 2.

Das Rückgrat meines gesamten Studiums bildete meine Familie, die mich immer unterstützt und so ein sorgenfreies Studieren ermöglichte, und die mir einen Rückzugs-, aber auch einen wunderschönen Ferienort bietet.

Wenn mein Geist mal streikte, konnte ich immer auf meine Freunde zurückkommen, die neben des Studiums für reichhaltige Abwechslung sorgten. Der Ma00-Truppe aus Ilmenau rund um Stephan verdanke ich viel im Grundstudium, das ohne sie wohl nicht ganz so angenehm gewesen wäre. Nach meinem Wechsel nach Jena fand ich auch dort gute Freunde, die mir nicht zuletzt gezeigt haben, dass Volleyball doch der einzig wahre Sport ist. Hier möchte ich vor allem bei Martin, Hannes, Tom und Katharina für die schöne Zeit bedanken und hoffe, dass auch die Zukunft noch einige gemeinsame Erlebnisse für uns bereit hält.

Doch der größte Dank geht an meine liebe Freundin Berit, mit der ich so viele wunderschöne Stunden verbracht habe, die alle meine Schwächen akzeptiert und die den größten Platz in meinem Herzen einnimmt.

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Jena, den 16. Februar 2006