

# Erweiterungsplanung in elektrischen Netzen mittels dynamischer Programmierung

Bachelorarbeit  
von

**Robert Mumper**

An der Fakultät für Informatik  
Institut für Theoretische Informatik

Erstgutachter:	Prof. Dr. Dorothea Wagner
Zweitgutachter:	Prof. Dr. Peter Sanders
Betreuende Mitarbeiter:	Franziska Wegner Matthias Wolf

Bearbeitungszeit: 26. November 2018 – 25. März 2019



### **Eigenständigkeitserklärung**

Ich versichere hiermit, die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt zu haben. Die verwendeten Hilfsmittel und Quellen sind im Literaturverzeichnis vollständig aufgeführt.

Karlsruhe, 25. März 2019



## **Abstract**

Durch die anhaltende Energiewende und den ständig steigenden Strombedarf der Verbraucher sind viele Stromnetze nicht mehr zeitgemäß oder werden in absehbarer Zukunft den Anforderungen nicht mehr entsprechen. Transmission Network Expansion Planning (TNEP) befasst sich unter anderem mit der Ermittlung optimaler Erweiterungen der aktuellen Stromnetze. In dieser Arbeit wird ein dynamisches Programm zur Lösung dieses Optimierungsproblems entwickelt. Da diese Arbeit eine theoretische Betrachtung zur Motivierung des Problems ist, wird lediglich die Klasse der serien-parallelen Graphen betrachtet. Als Referenz wird ein bekanntes Optimierungsprogramm herangezogen, mit dem die Ergebnisse und die Performance verglichen werden. Für alle betrachteten Eingaben liefert der vorgestellte Algorithmus optimale Ergebnisse. Die Laufzeiten des vorgestellten Algorithmus sind außer bei sehr kleinen Stromnetzen schlechter als die des Referenzprogramms.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Stand der Forschung</b>	<b>3</b>
<b>3</b>	<b>Grundlagen</b>	<b>5</b>
3.1	Einführung in Graphen . . . . .	5
3.2	Einführung in Stromnetze . . . . .	6
<b>4</b>	<b>Problemdefinition</b>	<b>9</b>
<b>5</b>	<b>Maximum Transmission Network Expansion Planning (MTNEP)</b>	<b>13</b>
5.1	Idee . . . . .	13
5.2	Algorithmus . . . . .	14
5.2.1	Beispiel . . . . .	16
5.3	Abschätzung . . . . .	18
<b>6</b>	<b>Evaluation</b>	<b>23</b>
6.1	Maximal möglicher Flusswert . . . . .	23
6.1.1	Serien-paralleler Graph . . . . .	23
6.1.2	Einfacher Graph . . . . .	24
6.2	Laufzeit . . . . .	24
<b>7</b>	<b>Zusammenfassung</b>	<b>29</b>
	<b>Literaturverzeichnis</b>	<b>31</b>
	<b>Anhang</b>	<b>35</b>





# 1. Einleitung

Stromnetze sind Netzwerke, welche aus elektrischen Stromleitungen wie Freileitungen und Erdkabeln und den dazugehörigen sowie angeschlossenen Einrichtungen bestehen. Diese Einrichtungen sind zum Beispiel Schalt-, Umspannwerke, Kraftwerke und Verbraucher. Durch den steigenden Strombedarf der Verbraucher ist die optimale Erweiterung der Stromnetze in den letzten Jahren zu einem wichtigen Thema geworden. Des Weiteren werden durch die Energiewende alte Kohle- und Atomkraftwerke abgeschaltet und der Fokus liegt auf erneuerbaren Energien [ALH01]. Oft sind Anlagen zur Gewinnung von erneuerbarer Energie an bestimmte Standorte gebunden, da gewisse Umweltfaktoren gegeben sein müssen. So macht ein Windpark, eine räumliche Ansammlung von Windenergieanlagen, nur dort Sinn, wo es regelmäßig Wind gibt. In Deutschland sind die Bedingungen für Windparks zum Beispiel hauptsächlich in Mittel- und Norddeutschland gegeben. Um die Einspeisung der Windstromerzeugungen durch neue Windparks zu ermöglichen, ist meist ebenfalls eine Erweiterung des Stromnetzes erforderlich. In vielen Arbeiten wurde das Optimierungsproblem Transmission Network Expansion Planning (TNEP) betrachtet, um die vorhandenen Energienetze so zu erweitern, dass sie zukünftige Anforderungen für die Energieversorgung erfüllen können. Bei TNEP geht es darum, die anfallenden Kosten für die Erweiterung zu minimieren und dabei Einschränkungen, die zum Beispiel durch die Eigenschaften der Kraftwerke entstehen, zu erfüllen.

In dieser Arbeit liegt der Fokus jedoch erst einmal darauf, Erweiterungen zu bestimmen, welche einen maximal möglichen Flusswert für ein vorliegendes Stromnetz liefern. Des Weiteren ist diese Arbeit eine theoretische Betrachtung zur Motivierung des Problems. Daher wird lediglich die einfache Klasse der serien-parallelen Graphen mit einer Quelle und einer Senke betrachtet. Außerdem wird das DC-Leistungsfluss-Modell zur Modellierung der Stromnetze verwendet, da das *feasibility problem* (das Machbarkeitsproblem eines zusätzlichen Leistungsfluss) im AC-Modell NP-schwer ist [BV15]. Viele Arbeiten mit Schwerpunkt TNEP verwenden Varianten der linearen Programmierung (LP [Gar70], MILP [SA85, AMC03] oder MINLP [RM94]) zur Lösung des Problems. Da bei TNEP viele gleichartige Teilprobleme vorkommen, ist dynamische Programmierung eine weitere Möglichkeit das Optimierungsproblem zu lösen. Es gibt jedoch nur wenige Arbeiten die, genauso wie diese Arbeit, dynamische Programmierung verwenden [DE73].

In Kapitel 2 wird der aktuelle Stand der Forschung zu TNEP und auf welche Art damit verbundene Optimierungsprobleme gelöst werden können dargelegt. Benötigte Grundlagen zu Graphen und Stromnetzen sind in Kapitel 3 beschrieben. In Kapitel 4 wird die Problemdefinition erläutert und mehrere Optimierungsprobleme vorgestellt. Eines dieser

Optimierungsprobleme wird in Kapitel 5 genauer betrachtet und ein dynamisches Programm vorgestellt, mit welchem das Problem gelöst werden kann. Dieses dynamische Programm wird in Kapitel 6 evaluiert. Am Ende, in Kapitel 7, wird die Arbeit zusammengefasst und ein Ausblick auf mögliche weiterführende Arbeiten gegeben.

## 2. Stand der Forschung

In [MAAR18] werden die relevantesten Publikationen mit Bezug zu Transmission Network Expansion Planning (TNEP) aufgrund ihrer vorgeschlagenen Modelle, Fallstudien, Lösungsmethoden und Innovationen klassifiziert. Dabei werden drei verschiedene Typen von Optimierungen vorgestellt. Die mathematische, die heuristische und die metaheuristische Optimierung.

Bei den mathematischen Optimierungen werden meistens klassische Optimierungstechniken, wie das Maximumprinzip, lineare Programmierung [Gar70], dynamische Programmierung [DE73], gemischt-ganzzahlige lineare Programmierung (MILP) [SA85, AMC03] oder gemischt-ganzzahlige nichtlineare Programmierung (MINLP) [RM94], verwendet. Auch werden Optimierungstechniken wie die Benders-Zerlegung verwendet, um das Problem mit „branch and bound“-Algorithmen zu lösen [HMG<sup>+</sup>00].

Für Heuristiken werden Techniken verwendet, die anstatt eines klassischen Optimierungsansatzes, Schritt für Schritt Erweiterungsoptionen generieren, bewerten und auswählen [RRMS05, OCB95]. Diese Techniken bloß Annäherungen an eine optimale Lösung und somit ungenauer als die klassischen Optimierungstechniken. Ihr Vorteil gegenüber klassischen Optimierungstechniken ist, dass sie nicht nur schneller sind, sondern auch weniger Rechenaufwand benötigen.

Bei der Metaheuristik werden heuristische Techniken iterativ angewandt, um qualitativ hochwertige Lösungen zu finden, wobei bestimmte Kriterien im Optimierungsprozess verwendet werden. Beispiele für Metaheuristiken sind Genetische Algorithmen (GA) [SGA00, dJSRR<sup>+</sup>05] und Greedy Randomized Adaptive Search Procedure (GRASP) [BdOdA01]. Bei GAs wird iterativ eine geeignete Lösung generiert, dabei werden von der Natur inspirierte Verfahren wie Mutation, Kreuzung oder Selektion verwendet [Hol92]. GRASP ist eine iterative Technik mit zwei Phasen. Bei der ersten Phase wird eine praktikable Lösung für das Problem gefunden. In der zweiten Phase wird mit Hilfe einer lokalen Suche nach Verbesserungen der Lösung der ersten Phase gesucht. Die beste Lösung für alle GRASP-Iterationen wird als Ergebnis ausgewählt. Metaheuristiken liefern wie auch Heuristiken lediglich eine näherungsweise Lösung für Optimierungsprobleme. Ihr Vorteil liegt darin, dass sie eine abstrakte Folge von Schritten definieren, die auf beliebige Problemstellungen angewandt werden können. Die einzelnen Schritte müssen jedoch wie Heuristiken auch problemspezifisch sein.

In [HHK13] und [LCAV03] wird ebenfalls eine Übersicht zu verschiedenen mathematischen als auch metaheuristischen Optimierungsmethoden betrachtet.

Aufgrund ihrer Berechnungskomplexität werden TNEP-Probleme traditionell mit Approximationen des Netzes untersucht [RMGH02]. Dabei wird der komplexe Wechselstrom zu einem Gleichstrom vereinfacht. Solche Annäherungen sind interessant, da sie Mixed Integer Linear Programs (MILPs) liefern, welche mit jahrzehntelanger Forschung im Bereich Netzwerkdesignoptimierung und vorhandenen kommerziellen Werkzeugen gelöst werden können.

Viele TNEP-Studien verwenden das weit verbreitete DC-Leistungsfluss-Modell. Die Anwendbarkeit des DC-Leistungsfluss-Modells für Leistungsfluss-Anwendungen wird jedoch diskutiert [HHK13]. Einige Arbeiten sind optimistisch eingestellt [OCS04, PMDB05], während andere, meist neuere Arbeiten, dem Modell gegenüber skeptisch eingestellt sind [SJA09]. Das DC-Modell hat bei TNEP-Anwendungen einige Nachteile, wie in [BTB12] festgestellt wurde. So sind die Anpassungen des DC-Modells bei Erzeugung von erneuerbarer Energie zum Beispiel nicht mehr einfach. Aufgrund dieser potenziellen Einschränkungen wurde jüngst begonnen, das TNEP-Problem mit den AC-Leistungsfluss-Gleichungen und dem damit verbundenen komplexen AC-Modell zu betrachten [BTB12, Jab13, RRCR10, RGR07]. In [BCGH14] wird zum Beispiel ein LPAC-Leistungsfluss-Modell verwendet, welches eine lineare Programmierung des Leistungsflusses ist, die die Blindleistung und die Spannungsgrößen erfasst.

In [MPS10] wird die Erweiterungsplanung unter dem Aspekt, dass das Netzwerk neu gestaltet werden kann, betrachtet. Dabei können zur Optimierung des Stromnetzes Leitungen sowohl hinzugefügt als auch entfernt werden. Dieses Modell beruht auf der Annahme, dass ein existierendes Übertragungsnetz, das in der Vergangenheit gestaltet wurde, möglicherweise nicht mehr optimal ist und in der Zukunft den Gegebenheiten nicht mehr entspricht.

In [DE73] wird eine diskrete dynamische Optimierung (DDO) angewandt. Die grundlegende Idee von DDO ist, den deterministischen Suchvorgang eines dynamischen Programms mit einer diskreten Optimierung, einer probabilistischen Suche mit heuristischem Abbruchkriterium, zu kombinieren. In dieser Arbeit wird ebenfalls ein dynamisches Programm verwendet, um die Problemstellung zu lösen. Es wird aber kein heuristisches Abbruchkriterium verwendet und somit nicht nur eine näherungsweise Lösung gefunden, sondern eine optimale Lösung berechnet.

## 3. Grundlagen

In diesem Kapitel werden Grundlagen dargelegt, die benötigt werden, um Konzepte und Denkansätze in dieser Arbeit zu verstehen.

### 3.1 Einführung in Graphen

Ein *Graph* ist ein Tupel  $(V, E)$ , wobei  $V$  eine Menge von Knoten und  $E$  eine Menge von Kanten bezeichnet. Eine Kante ist wiederum ein Tupel  $(i, j)$  von zwei Knoten  $i, j \in V$ . Jeder ungerichtete Graph kann in einen gerichteten Graphen überführt werden, indem eine ungerichtete Kante durch zwei gerichtete Kanten in beide Richtungen ersetzt wird. Diese Art von Graphen nennt man *bidirektionalen Graphen*. Zur Vereinfachung wird  $\underline{E}$  zur Bezeichnung der zugrunde liegenden Menge von ungerichteten Kanten verwendet und für  $e \in E$  bezeichnet  $\underline{e} \in \underline{E}$  die zugrunde liegende ungerichtete Kante. So gilt zum Beispiel  $\underline{(i, j)} = \underline{(j, i)}$ .

**Definition 3.1** (Pfad). *Ein nicht-leerer Graph  $G$ , mit der Knotenmenge  $\{x_1, x_2, \dots, x_n\}$  und der Kantenmenge  $\{\{x_1, x_2\}, \{x_2, x_3\}, \dots, \{x_{n-1}, x_n\}\}$ , heißt Pfad, wenn die Knoten  $x_i$   $i \in 1, \dots, n$  paarweise verschieden sind. Ein Pfad von Knoten  $x_1$  nach Knoten  $x_n$  wird mit  $\pi(1, n)$  angegeben.*

**Definition 3.2** (Serien-paralleler Graph). *Ein Graph wird als serien-parallel bezeichnet, wenn er*

- aus genau zwei Knoten (Quelle  $s$ , Senke  $t$ ) sowie der Kante  $(s, t)$  besteht oder
- aus zwei serien-parallelen Graphen  $G_1, G_2$ , mit Quellen  $s_1, s_2$  und Senken  $t_1, t_2$ , durch eine der folgenden Kombinationen hervorgeht
  - *Serielle Komposition (s. Abb. 3.1 (a))*: Die Senke  $t_1$  und Quelle  $s_2$  werden identifiziert und bilden einen neuen Knoten. Knoten  $s_1$  wird zur neuen Quelle und die neue Senke ist  $t_2$ .
  - *Parallele Komposition (s. Abb. 3.1 (b))*: Die Quellen  $s_1$  und  $s_2$  sowie  $t_1$  und  $t_2$  werden jeweils identifiziert und beschreiben die neue Quelle bzw. Senke.

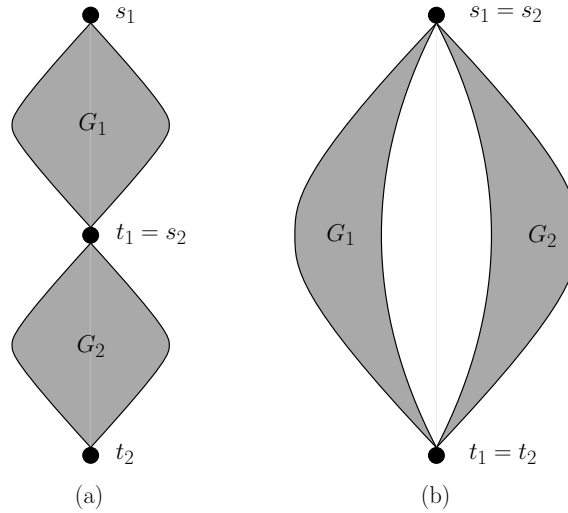


Abbildung 3.1: (a) Eine serielle Komposition der serien-parallelen Graphen  $G_1$  und  $G_2$ . (b) Eine parallele Komposition der serien-parallelen Graphen  $G_1$  und  $G_2$ .

### 3.2 Einführung in Stromnetze

Ein *Stromnetz* besteht aus einem Graphen  $G$ , einer Menge von *Erzeugern*  $V_g \subseteq V$  und *Verbrauchern*  $V_l \subseteq V$  sowie den Funktionen  $b$ ,  $\text{tlim}$ ,  $\bar{d}$ ,  $\underline{d}$ ,  $\bar{x}$  und  $\underline{x}$ . Dabei gibt  $b: E \rightarrow \mathbb{R}^+$  die *Suszeptanz* und  $\text{tlim}: E \rightarrow \mathbb{R}^+$  das *thermische Limit* (Kapazität) der Kanten an. Der *maximal mögliche Bedarf* eines Knoten wird durch die Funktion  $\bar{d}: V_l \rightarrow \mathbb{R}^+$  beschrieben und die *maximal mögliche Erzeugung* durch  $\bar{x}: V_g \rightarrow \mathbb{R}^+$ . Der *minimale Bedarf* und die *minimale Erzeugung* werden durch  $\underline{d}$  und  $\underline{x}$  festgelegt. Ist ein maximal möglicher Bedarf  $\bar{d}$  gegeben, handelt es sich um ein *beschränktes Netz*  $N_b = (G, V_g, V_l, b, \text{tlim}, \underline{d}, \bar{d}, \underline{x}, \bar{x})$ , ansonsten liegt ein *unbeschränktes Netz*  $N = (G, V_g, V_l, b, \text{tlim}, \underline{d})$  vor.

Der folgende Abschnitt und die darin angewandten Vereinfachungen basieren auf den Arbeiten von R. Zimmermann et al [ZMT09, ZMS11]. Da das *feasibility problem* (das Machbarkeitsproblem eines zusätzlichen Leistungsflusses) im AC-Modell NP-schwer ist [BV15], wenden wir im Folgenden Annäherungen an, um auf das linearisierte AC-Modell zu kommen. Dieses wird auch DC genannt, wobei man es nicht mit dem eigentlichen DC-Modell verwechseln darf. In dieser Arbeit ist mit DC-Modell stets das linearisierte AC-Modell gemeint.

$$P_i = \sum_{j=1}^N |V_i||V_j| (G_{ij} \cdot \cos(\theta_i - \theta_j) + B_{ij} \cdot \sin(\theta_i - \theta_j)) \quad (3.1)$$

$$Q_i = \sum_{j=1}^N |V_i||V_j| (G_{ij} \cdot \sin(\theta_i - \theta_j) - B_{ij} \cdot \cos(\theta_i - \theta_j)) \quad (3.2)$$

Im AC-Modell setzt sich die *Scheinleistung*  $S$  aus der *Wirkleistung*  $P$  und der *Blindleistung*  $Q$  zusammen. Für die Scheinleistung an Knoten  $i$  gilt also  $S_i = P_i + jQ_i$  wobei  $j$  eine imaginäre Einheit ist. Die Wirkleistung und Blindleistung an Knoten  $i$  sind wie in den Gleichungen 3.1 und 3.2 gegeben. Dabei entspricht  $N$  der Anzahl der Nachbarknoten von Knoten  $i$ . Im DC-Modell wird die Scheinleistung lediglich durch die tatsächliche Wirkleistung bestimmt, womit die Blindleistung und Gleichung 3.2 vernachlässigt werden können.

Der *komplexe Wechselstromwiderstand*  $Z$  wird auch *Scheinwiderstand* (engl. Impedance) genannt und ist durch  $Z = R - jX$  definiert, wobei  $R$  die *resistive Komponente* (*Realteil*) und

$X$  die *reaktive Komponente (imaginärer/komplexer Teil)* ist. Der Realteil ist üblicherweise signifikant kleiner als der Imaginärteil, somit ergibt sich für die *Admittanz*  $Y$

$$\begin{aligned} Y &= \frac{1}{Z} = \frac{1}{R - jX} = \frac{1}{R - jX} \cdot \frac{R + jX}{R + jX} = \frac{R + jX}{R^2 + X^2} \\ &= \frac{R}{R^2 + X^2} + \frac{jX}{R^2 + X^2} = G + jB. \end{aligned} \quad (3.3)$$

Aus Gleichung 3.3 erkennt man

$$G = \frac{R}{R^2 + X^2} \quad (3.4)$$

$$B = \frac{X}{R^2 + X^2}. \quad (3.5)$$

Wenn  $R$  verglichen mit  $X$  sehr klein ist, ist folglich der *Wirkleitwert*  $G$  verglichen mit dem *Blindleitwert* bzw. *Suszeptanz*  $B$  sehr klein und es ist möglich, die Gleichungen 3.4 und 3.5 folgendermaßen zu vereinfachen

$$G = 0 \quad (3.6)$$

$$B = \frac{1}{X}. \quad (3.7)$$

Der *Wirkleitwert*  $G$  wird dann 0. Die *Suszeptanz*  $B$  entspricht dem Kehrwert des Imaginärteils des komplexen Wechselstromwiderstands  $Z$ . Die Spannungsgrößen  $|V_i|$  und  $|V_j|$  sind im Bereich der Nennspannung, die für die Geräte geeignete Spannung, da sonst Geräte wie zum Beispiel Transformatoren beschädigt werden können. Deswegen sind die Spannungsgrößen  $|V_i|$  und  $|V_j|$  sehr nah an 1,0. Treten die beiden Größen als Multiplikationsfaktoren auf, kann man die Vereinfachung  $|V_i| = |V_j| = 1$  verwenden. Die Annahme, dass die Spannungswinkel  $\theta$  sehr klein sind, erlaubt es den Sinus durch sein Argument zu ersetzen. Daraus ergeben sich folgende Annäherungen

$$\sin(\theta_i - \theta_j) \approx (\theta_i - \theta_j) \quad (3.8)$$

$$\cos(\theta_i - \theta_j) \approx 1. \quad (3.9)$$

Wendet man diese Vereinfachungen auf die Gleichung 3.1 an, ergibt sich für die Wirkleistung an Knoten  $i$

$$P_i = \sum_{j=1}^N (B_{ij} \cdot (\theta_i - \theta_j)) \quad (3.10)$$

und für den Wirklastfluss über die Leitung von Knoten  $i$  nach Knoten  $j$

$$P_{ij} = B_{ij} \cdot (\theta_i - \theta_j). \quad (3.11)$$

Das DC-Modell wird durch das Ohmsche Gesetz  $U = R \cdot I$  beschrieben. Dabei entspricht die Spannung  $U$  der *Spannungswinkeldifferenz*  $\Delta\theta: \pi(e_1, e_n) \rightarrow \mathbb{R}^+$ . Stellt man die Gleichung 3.11 nach der Spannungswinkeldifferenz um, erkennt man die Analogie zum Ohmschen Gesetz. Die Suszeptanz lässt sich somit analog zum Widerstand in der Elektrizitätslehre berechnen. Die Suszeptanz von parallelen Leitungen ist der Kehrwert der Summe aller Kehrwerte der einzelnen Suszeptanzen auf dem Pfad

$$b_{\text{ges}}(s, t) = \frac{1}{\sum_{e \in \pi(s, t)} \frac{1}{b(e)}}. \quad (3.12)$$

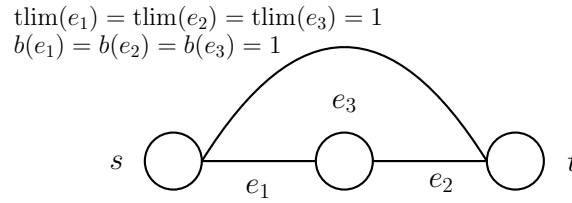


Abbildung 3.2: Beispiel Graph mit Kanten  $e_1$ ,  $e_2$  und  $e_3$ , die jeweils eine Suszeptanz und ein thermisches Limit von 1 haben. Die Gesamtsuszeptanz ist  $b_{\text{ges}}(\pi(s, t)) = \frac{3}{2}$  und der maximal mögliche Fluss  $\text{cap}_{\text{ges}}(s, t) = \frac{3}{2}$ .

Betrachtet man parallele Leitungen, so summiert man die Suszeptanzen der einzelnen Leitungen auf

$$b_{\text{ges}}(s, t) = \sum_{e \in \pi(s, t)} b(e). \quad (3.13)$$

Der Strom  $I$  entspricht dem maximal möglichen Fluss  $\text{cap}_{\text{ges}}$ . Der maximal mögliche Fluss aller Pfade von  $s \in V$  nach  $t \in V$  wird durch  $\text{cap}_{\text{ges}}(s, t)$  bestimmt. Durch die Analogie zur Stromberechnung in der Elektrizitätslehre  $I = \frac{U}{R}$  ergibt sich für den maximal möglichen Fluss von parallelen Leitungen

$$\text{cap}_{\text{ges}}(s, t) = \sum_{\pi((s, t)^i)} b((s, t)^i) \cdot \min_{(s, t)^i}(\Delta\theta_i(s, t)). \quad (3.14)$$

Bei einer Reihenschaltung entspricht der Gesamtfluss dem Fluss der einzelnen Leitungen. Insbesondere wird der Fluss, analog zum Strom, durch die Kante mit der kleinsten Kapazität auf dem Pfad beschränkt

$$\text{cap}_{\text{ges}}(s, t) = \min \{ \text{tlim}(e_1), \text{tlim}(e_2), \dots, \text{tlim}(e_n) \} \quad e_1, e_2, \dots, e_n \in \pi(s, t). \quad (3.15)$$

Die Spannungswinkeldifferenz ist, analog zum Ohmschen Gesetz, als  $\Delta\theta(s, t) = \frac{\text{cap}_{\text{ges}}(s, t)}{b_{\text{ges}}(s, t)}$  definiert.

In Abbildung 3.2 sind die Kanten  $e_1$  und  $e_2$  in Reihe geschaltet und die Kante  $e_3$  parallel zu den beiden Kanten. Daher ergibt sich eine Gesamtsuszeptanz  $b_{\text{ges}} = \frac{1}{\frac{1}{b(e_1)} + \frac{1}{b(e_2)}} + b(e_3) =$

$$\frac{1}{\frac{1}{1} + \frac{1}{1}} + 1 = \frac{3}{2} \text{ und ein maximal möglicher Fluss } \text{cap}_{\text{ges}} = b_{\text{ges}} \cdot \min \left\{ \left( \frac{\min\{\text{tlim}(e_1), \text{tlim}(e_2)\}}{b_{\text{ges}} - b(e_3)} \right), \frac{\text{tlim}(e_3)}{b(e_3)} \right\} =$$

$$\frac{3}{2} \cdot \min \left\{ \left( \frac{\min\{1, 1\}}{\frac{3}{2} - 1} \right), \frac{1}{1} \right\} = \frac{3}{2}.$$



## 4. Problemdefinition

Betrachtet man das *aktuelle elektrische Netzwerk*  $N^0$  und möchte dieses erweitern, kann man ein *Kandidatennetz*  $N^1$  verwenden, welches alle Möglichkeiten zur Erweiterung beinhaltet. Aus der Sicht der kombinatorischen Optimierung sind diese Netze jeweils ein ungerichteter Graph  $\underline{G}^0 = (V, \underline{E}^0)$  respektive  $\underline{G}^1 = (V, \underline{E}^1)$ . Der Graph des aktuellen Netzes  $\underline{G}^0$  und der des Kandidatennetzes  $\underline{G}^1$  ergeben einen vollständigen Graphen  $\underline{G}_{\text{vollst}}$  wie in 4.1. Die beiden Graphen können als bidirektionaler Graph  $G = (V, E)$  repräsentiert werden. Hierbei werden die Knoten  $i \in V$  auch Leiterschienen (engl. *buses*) und die Kanten  $e \in E$  auch Leitungen (engl. *branches*) genannt. Die Menge der Knoten besteht aus erzeugenden Knoten  $V_G \subseteq V$ , im Folgenden *Erzeuger* genannt, verbrauchenden Knoten  $V_C \subseteq V$ , im Folgenden *Verbraucher* genannt und Zwischenknoten  $V \setminus (V_G \cup V_C)$ , sodass  $V_G \cap V_C = \emptyset$ . Die Menge der Kanten wird in die Menge  $E^0 \subseteq E$  der bereits existierenden Kanten und die Menge  $E^1 \subseteq E$  der möglichen Kandidaten für neue Kanten unterteilt. Dabei gilt  $E^0 \cap E^1 = \emptyset$  und  $E^0 \cup E^1 = E$ . Der *Fluss* dieser Kanten wird durch die Funktion  $f: E \rightarrow \mathbb{R}^+$  angegeben. Der *maximal mögliche Fluss* (auch Kapazität der Kante genannt) einer Kante wird durch die Funktion  $t_{\text{lim}}: E \rightarrow \mathbb{R}^+$  beschrieben. Der Nettofluss eines Knoten  $i \in V$  ist durch

$$f_{\text{net}}(i) := \sum_{\{i,j\} \in \underline{E}} f(i,j) \quad (4.1)$$

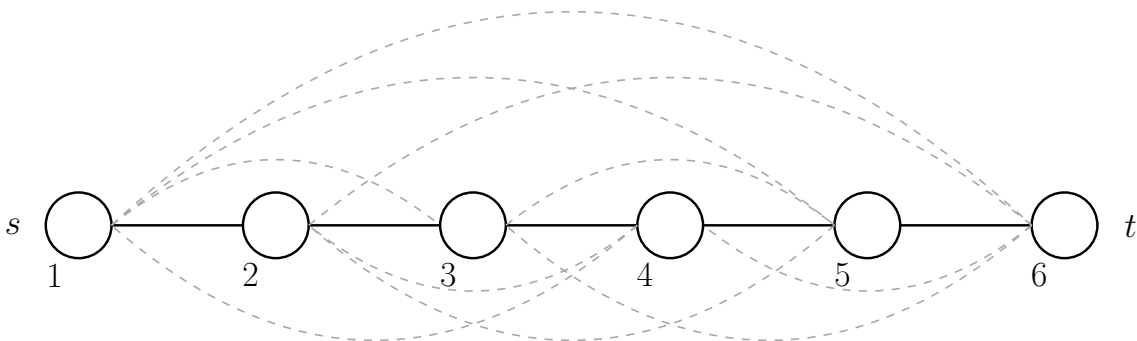


Abbildung 4.1: Vollständiger Graph  $\underline{G}_{\text{vollst}}$  mit Knotenmenge  $\{1,2,3,4,5,6\}$ . Schwarze Kanten entsprechen den Kanten aus dem Graphen des aktuellen Netzes  $\underline{G}^0$ . Graue Kanten entsprechen den Kanten aus dem Graphen des Kandidatennetzes  $\underline{G}^1$ .

gegeben. Die *maximale Erzeugung* eines Knotens wird mit  $\bar{x}: V \rightarrow \mathbb{R}^+$  dargestellt, der *minimale Bedarf* durch  $\underline{d}: V \rightarrow \mathbb{R}^+$ . Die *Suszeptanz* einer Kante wird durch  $b: E \rightarrow \mathbb{R}^+$  beschrieben. Des Weiteren gibt  $\sigma: E \rightarrow \mathbb{R}^+$  die *Kosten* der Kanten an. Der *Spannungswinkel* wird durch  $\theta: V \rightarrow \mathbb{R}$  beschrieben. Ob eine Kante ausgebaut wird, wird mit der Entscheidungsvariable  $y: E^1 \rightarrow \{0, 1\}$  festgelegt.

Des Weiteren gelten folgende Bedingungen:

$$f_{\text{net}}(i) = 0 \quad \forall i \in V \setminus \{V_g \cup V_l\} \quad (4.2)$$

$$-\infty \leq f_{\text{net}}(i) \leq -\underline{d} \quad \forall i \in V_l \quad (4.3)$$

$$0 \leq f_{\text{net}}(i) \leq \bar{x}(i) \quad \forall i \in V_g \quad (4.4)$$

$$f(i, j) - b(i, j) \cdot (\theta(i) - \theta(j)) = 0 \quad \forall (i, j) \in E^0 \quad (4.5)$$

$$f(i, j) - b(i, j) \cdot y(i, j) \cdot (\theta(i) - \theta(j)) = 0 \quad \forall (i, j) \in E^1 \quad (4.6)$$

$$|f(e)| \leq \text{tlim}(e) \quad \forall e \in E. \quad (4.7)$$

Die Bedingungen 4.2 bis 4.4 stellen die Flusserhaltung in jedem Knoten dar (auch Kirchhoffs Knotenregel genannt). Die Maschenregel (engl. Kirchhoff's voltage law, kurz KVL) besagt, dass alle Teilspannungen einer Masche (Zyklus im Graph) sich zu Null addieren und wird von den Bedingungen 4.5 und 4.6 modelliert. Durch die letzte Bedingung wird der Flusswert beschränkt.

$$f(i, j) - b(i, j) \cdot (\theta(i) - \theta(j)) - (1 - y(i, j))M \leq 0 \quad \forall (i, j) \in E^1 \quad (4.8)$$

$$f(i, j) - b(i, j) \cdot (\theta(i) - \theta(j)) + (1 - y(i, j))M \geq 0 \quad \forall (i, j) \in E^1 \quad (4.9)$$

Die Bedingungen (Gleichungen 4.8 und 4.9) linearisieren die quadratische Bedingung 4.6 und werden big-M-Bedingungen genannt. Der Faktor M ist eine geeignet große Konstante, um keine implizite Spannungswinkeldifferenzgrenze an der Kante festzulegen.

Im Folgenden werden drei mögliche Optimierungsprobleme vorgestellt.

Das erste dieser Optimierungsprobleme wird *Transmission Network Expansion Planning* (kurz *TNEP*) genannt. Um dieses Optimierungsproblem von den weiteren abzugrenzen, wird es im Folgenden *Minimum Cost TNEP* genannt. Bei Minimum Cost TNEP wird eine Funktion, welche die Kosten für einen Ausbau eines beschränkten Netzwerkes  $N_b = (G, V_g, V_l, b, \text{tlim}, \underline{d}, \bar{x}, \underline{x})$  darstellt, minimiert. Die Kosten für eine Kante werden, wie bereits erwähnt, durch die Funktion  $\sigma$  beschrieben. Nun werden für jede Kante  $e \in E^1$  diese Kosten mit  $y(e)$  multipliziert. Anschließend werden die Ergebnisse der Multiplikationen über die Menge der Kanten  $E^1$  aufsummiert. Die Zielfunktion sieht also folgendermaßen aus:

$$\min \sum_{e \in E^1} \sigma(e) \cdot y(e). \quad (4.10)$$

Ein weiteres Optimierungsproblem ist *Maximum Transmission Network Expansion Planning* (kurz *MTNEP*). Dafür wird die Funktion  $f_{\text{net}}$  verwendet. Bei MTNEP soll der *Fluss*  $F := \sum_{v \in V_g} f_{\text{net}}(v)$  eines unbeschränkten Netzwerkes  $N = (G, V_g, V_l, b, \text{tlim}, \underline{d})$  maximiert werden

$$\max \sum_{v \in V_g} f_{\text{net}}(v). \quad (4.11)$$

*Optimal Transmission Network Expansion Planning* (kurz *OTNEP*) ist das dritte Optimierungsproblem. Hierbei betrachtet man ein beschränktes Netzwerk  $N_b$  und berechnet den Nettofluss der Erzeuger und multipliziert diesen mit einer Funktion  $\rho: V \rightarrow \mathbb{R}^+$ , welche

---

die Kosten für die Erzeugung einer Einheit beschreibt. Diese Erzeugungskosten sollen wie folgt minimiert werden:

$$\min \sum_{v \in V_g} \rho(v) \cdot f_{\text{net}}(v). \quad (4.12)$$

Das zuletzt vorgestellte Optimierungsproblem entspricht einer Erweiterung des *Optimal Power Flows* (OPF), der den Leistungsfluss so optimiert, dass die Erzeugungskosten minimiert werden.



## 5. Maximum Transmission Network Expansion Planning (MTNEP)

Gegeben sei ein einfacher Pfad  $G_0 = (V, E_0)$  zwischen den Knoten  $i \in V$  und  $j \in V$  und ein vollständiger Graph  $G_1 = (V, E_1 \setminus E_0)$ . Da in dieser Arbeit die Topologie des zugrunde liegenden Graphen  $G$  des Stromnetzes  $N$  stets *serien-parallel* sein muss, ist eine Kantenmenge  $E_{neu} \subseteq E_1$  gesucht, sodass der Leistungsfluss des Graphen  $G_k = (V, E_0 \cup E_{neu})$  maximal und  $G_k$  serien-parallel ist.

### 5.1 Idee

Um den Algorithmus zu motivieren, verwenden wir ein Beispiel, in dem  $G_0 = (V, E_0)$  ein ungerichteter Graph mit der Knotenmenge  $V = \{1, 2, 3, 4, 5, 6\}$  und Kantenmenge  $E = \{(1, 2), (2, 3), (3, 4), (4, 5), (5, 6)\}$ , wie in Abb. 5.1 dargestellt, und  $G_1 = (V, E_1 \setminus E_0)$  der dazugehörige vollständige Graph sei. Des Weiteren sei die Suszeptanz  $b(e) = 1$  und das thermische Limit  $\text{tlim}(e) = 1$  für alle  $e \in E_0 \cup E_1$  und  $\theta(s)$  wird mit 0 festgelegt. Nun kann man einen *maximal möglichen Leistungsflusswert*  $\text{OPT}_{\text{MPF}}(N + E_{neu})$  durch hinzufügen von Kanten in  $E_{neu}$  erhöhen. Dabei muss der resultierende Graph  $G_k$  serien-parallel sein. So kann man also eine weitere Kante  $e_1 = (1, 6)$ , in Abb. 5.2 blau gefärbt, mit  $b(e_1) = 1$  und  $\text{tlim}(e_1) = 1$  in  $E_{neu}$  einfügen und  $\text{OPT}_{\text{MPF}}(N + E_{neu})$  von 1 auf 1,2 erhöhen, da die Suszeptanz auf 1,2 steigt (siehe Gleichung 5.2) und der Spannungswinkel an der Senke durch den Pfad mit der kleinsten Spannungswinkeldifferenz bestimmt wird (siehe Gleichung 5.3). Da keine weiteren Kanten zwischen den Knoten  $s$  und  $t$  in  $E_1$  vorhanden sind, muss man den Graphen in serielle Komponenten aufteilen, um weitere Kanten hinzuzufügen. Teilt man den Graphen an dem Knoten 4 auf, können die in Abb. 5.2 grün gefärbten Kanten  $e_2 = (1, 4)$  und  $e_3 = (4, 6)$  mit  $b(e_2) = b(e_3) = 1$  und  $\text{tlim}(e_2) = \text{tlim}(e_3) = 1$  in  $E_{neu}$  eingefügt werden. Dadurch steigt  $\text{OPT}_{\text{MPF}}(N + E_{neu})$  von 1,2 auf  $\frac{12}{7}$ , denn die Suszeptanz

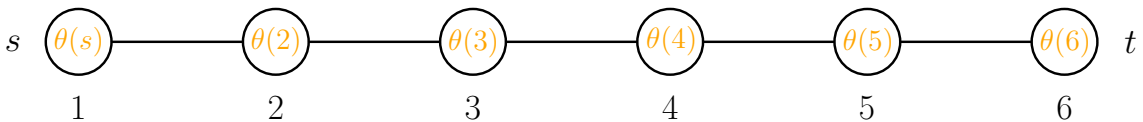


Abbildung 5.1: Ung gerichteter Graph  $G_0 = (V, E_0)$  mit Knotenmenge  $V = \{1, 2, 3, 4, 5, 6\}$ , Kantenmengen  $E_0 = \{(1, 2), (2, 3), (3, 4), (4, 5), (5, 6)\}$  mit  $b(e) = 1$  und  $\text{tlim}(e) = 1 \forall e \in E_0$  und Spannungswinkeln  $\theta$ .

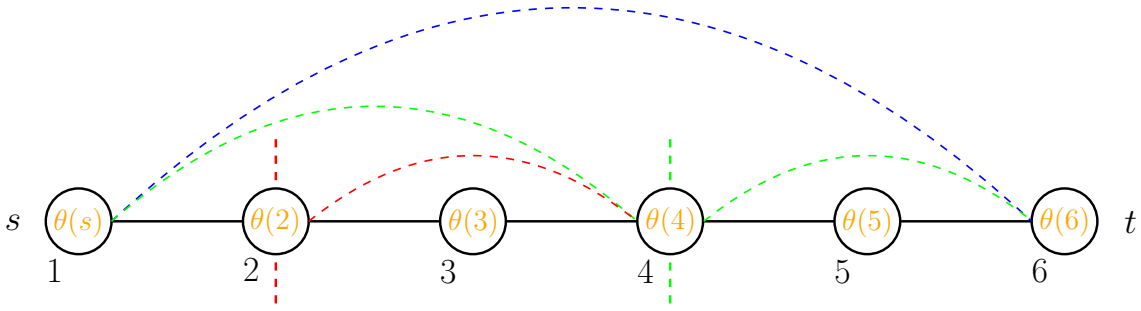


Abbildung 5.2: Ungerichteter Graph  $G_k$  mit Knotenmenge  $V = \{1, 2, 3, 4, 5, 6\}$ , Kantenmengen  $E_0 \cup E_{\text{neu}} = \{e_1 = (1, 6), e_2 = (1, 4), e_3 = (4, 6), e_4 = (2, 4)\}$  mit  $b(e) = 1$  und  $\text{tlim}(e) = 1 \forall e \in E_0 \cup E_{\text{neu}}$  und Spannungswinkeln  $\theta$ .

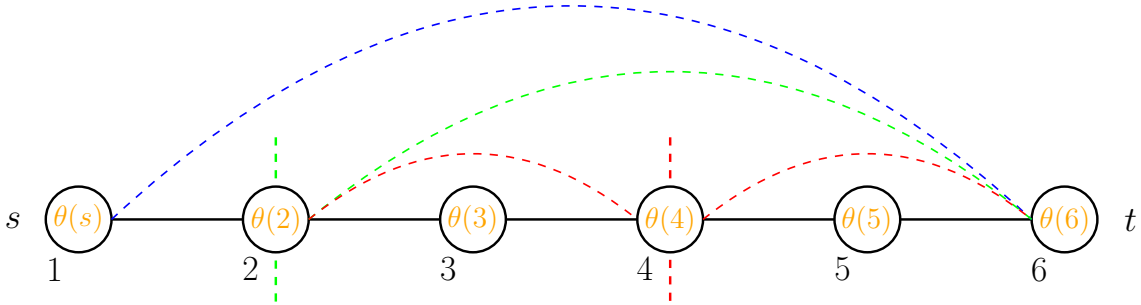


Abbildung 5.3: Ungerichteter Graph  $G_k$  mit Knotenmenge  $V = \{1, 2, 3, 4, 5, 6\}$ , Kantenmengen  $E_0 \cup E_{\text{neu}} = \{e_1 = (1, 6), e_2 = (2, 6), e_3 = (2, 4), e_4 = (4, 6)\}$  mit  $b(e) = 1$  und  $\text{tlim}(e) = 1 \forall e \in E_0 \cup E_{\text{neu}}$  und Spannungswinkeln  $\theta$ .

steigt auf  $\frac{12}{7}$  (siehe Gleichung 5.2) und der Spannungswinkel an der Senke wird weiterhin durch den Pfad über die Kante  $e_1 = (1, 6)$  bestimmt. Am Knoten 2 kann der Graph ein weiteres Mal aufgeteilt werden und dadurch, die in Abb. 5.2 rot gefärbte Kante  $e_4 = (2, 4)$ , mit  $b(e_4) = 1$  und  $\text{tlim}(e_4) = 1$  in  $E_{\text{neu}}$  eingefügt werden, womit  $\text{OPT}_{\text{MPF}}(N + E_{\text{neu}})$  auf  $\frac{16}{9}$  steigt. Auch hier wird der Spannungswinkel an der Senke weiterhin durch den Pfad über die Kante  $e_1 = (1, 6)$  bestimmt. Der neue Wert für  $\text{OPT}_{\text{MPF}}(N + E_{\text{neu}})$  ergibt sich durch die neue Suszeptanz von  $\frac{16}{9}$ .

Teilt man den Graphen  $G$  zuerst am Knoten 2, dann am Knoten 4 auf und fügt jeweils die entsprechenden Kanten hinzu, erhält man den Graphen aus Abb. 5.3. Dieser hat einen maximal möglichen Leistungsflusswert  $\text{OPT}_{\text{MPF}}(N + E_{\text{neu}}) = \frac{13}{8}$ . Es ist also zu erkennen, dass die Reihenfolge, in welcher man einen Graphen aufteilt und welche Kanten man zu  $E_{\text{neu}}$  hinzufügt, zu unterschiedlichen Ergebnissen für  $\text{OPT}_{\text{MPF}}(N + E_{\text{neu}})$  führen können. Daher sind mehrere Kombinationen zu berechnen, um diejenige mit dem größten  $\text{OPT}_{\text{MPF}}(N + E_{\text{neu}})$  zu finden. Dies lässt sich mit einem dynamischen Programm realisieren. Dabei wird das Problem in Teilprobleme aufgeteilt und die Zwischenresultate gespeichert. Mit diesen Zwischenresultaten wird eine Tabelle wie in Tabelle 5.1 aufgebaut.

## 5.2 Algorithmus

Wenn man die Gesamtsuszeptanz  $b_{\text{ges}}$  und die Gesamtspannungswinkeldifferenz  $\Delta\theta_{\text{ges}}$  des Graphen  $G_k$  kennt, kann man einen optimalen Wert  $\text{OPT}_{\text{MPF}}(N + E_{\text{neu}})$  des Netzwerkes berechnen. Diese zwei Werte werden in den Tabelleneinträgen mit Label gespeichert.

**Definition 5.1** (Label). Ein Label  $l$  besteht aus zwei Komponenten – einem Wert für die Suszeptanz  $b$  und einem Wert für die Differenz der Spannungswinkel  $\Delta\theta$ .

$$l_i = (b_i, \Delta\theta_i) \quad (5.1)$$

Tabelle 5.1: Mögliche Aufteilungen für den Graphen  $G$ .

$T_{[a,b]}$	b	1	2	3	4	5	6
a							
1	-	$T_{[1,2]}$	$T_{[1,3]}$	$T_{[1,4]}$	$T_{[1,5]}$	$T_{[1,6]}$	
2	-	-	$T_{[2,3]}$	$T_{[2,4]}$	$T_{[2,5]}$	$T_{[2,6]}$	
3	-	-	-	$T_{[3,4]}$	$T_{[3,5]}$	$T_{[3,6]}$	
4	-	-	-	-	$T_{[4,5]}$	$T_{[4,6]}$	
5	-	-	-	-	-	$T_{[5,6]}$	
6	-	-	-	-	-	-	-

Ein Tabelleneintrag  $T_{[i,j]}$  besteht aus einer Menge von Label, die Werte für die Gesamtsuszeptanz und Gesamtspannungswinkeldifferenz verschiedener resultierender Graphen  $G_k = (\{i, \dots, j\}, E_0 \cup E_{\text{neu}})$ , die durch unterschiedliche Erweiterungen entstanden sind, angeben. Das Label eines Tabelleneintrags  $T_{[i,j]}$ , für den Fall dass  $|i - j| = 1$  gilt, entspricht  $(b(i, j), \Delta\theta(i, j))$  also den Werten der Kante selbst. Für alle andere Fälle berechnen sich die Label wie in Gleichung 5.4 angegeben. Es werden also jeweils die Funktionen  $\beta$  und  $\gamma$ , aus den Gleichungen 5.2 und 5.3, auf die Label  $l_1 \in T_{[i,j-n]}$  und  $l_2 \in T_{[j-n,j]}$  für alle  $n \in \{1, \dots, j - i - 1\}$  angewendet, um die Gesamtsuszeptanz und Gesamtspannungswinkeldifferenz aller möglichen Erweiterungen, bei denen der resultierende Graph  $G_k$  serien-parallel ist, zu erhalten. Es entsteht ein rekursiver Aufruf der Labelberechnungen, wobei der Basisfall, welcher die Rekursion abbricht, der Fall  $|i - j| = 1$  ist. Um  $E_{\text{neu}}$  zu bestimmen, enthält jedes neue Label  $l$  in einem Tabelleneintrag  $T_{[i,j]}$  zwei Pointer auf die Label  $l_1 \in T_{[i,j-n]}$  und  $l_2 \in T_{[j-n,j]}$ , aus welchen das neue Label  $l$  berechnet wurde und eine Information, ob die Kante hinzugefügt wurde ( $y = 1$ ) oder nicht ( $y = 0$ ).

Der Eintrag  $T_{[1,4]}$  in Tabelle 5.1 besteht aus acht Label – zwei Kompositionen von  $T_{[1,2]}$  und  $T_{[2,4]}$  und zwei Kompositionen von  $T_{[1,3]}$  und  $T_{[3,4]}$ . Denn  $T_{[2,4]}$  und  $T_{[1,3]}$  enthalten zwei Label wie in dem nachfolgenden Beispiel in den Gleichungen 5.9 und 5.10 zu sehen ist, da jeweils die Möglichkeit besteht, die Kante (2, 4) bzw. (1, 3) hinzuzufügen. Bei diesen vier Kompositionen besteht wieder die Möglichkeit die Kante (1, 4) hinzuzufügen, wodurch sich die acht Label im dem Eintrag  $T_{[1,4]}$  ergeben.

Die Kompositionen der Suszeptanzen für zwei Tabelleneinträge lässt sich mit der Funktion in Gleichung 5.2 berechnen. Hierbei werden wie in Gleichung 3.12 die Kehrwerte der Suszeptanzen der beiden Tabelleneinträge addiert und davon der Kehrwert gebildet. Anschließend wird die Suszeptanz der neuen Kante  $b(i, j)$  addiert, falls diese hinzugefügt wird (s. Gleichung 3.13). Die Kompositionen der Spannungswinkeldifferenz  $\Delta\theta_{\text{ges}}$  wird mit der Funktion in Gleichung 5.3 berechnet. Hierbei wird das Minimum für  $\text{OPT}_{\text{MPF}}(N)$  der beiden Tabelleneinträge durch die Gesamtsuszeptanz  $b_{\text{ges}}$  geteilt. Dabei verwendet man die Gesamtsuszeptanz  $b_{\text{ges}}$ , wie sie in Gleichung 5.2 für  $y(i, j) = 0$  berechnet wird. Gilt  $y(i, j) = 1$ , die Kante  $(i, j)$  wird also hinzugefügt, ist die Spannungswinkeldifferenz  $\Delta\theta_{\text{ges}}$  das Minimum aus der zuvor berechneten Spannungswinkeldifferenz  $\Delta\theta_{\text{ges}}$ , ohne hinzugefügte Kante  $(i, j)$  und dem Quotienten des thermischen Limits  $t_{\text{lim}}$  und der Suszeptanz  $b$  der neuen Kante  $(i, j)$ .

$$\beta\left((b_1, \Delta\theta_1), (b_2, \Delta\theta_2), y, (i, j)\right) = \underbrace{\frac{1}{\frac{1}{b_1} + \frac{1}{b_2}}}_{\text{Serienschaltung exklusiv Kante}} + \underbrace{b(i, j) \cdot y(i, j)}_{\text{neue Kante}}. \quad (5.2)$$

Tabelle 5.2: Mögliche Aufteilungen für den Graphen  $G_1$  aus Abb. 5.4.

$T_{[a,b]}$	b	1	2	3	4
a					
1		-	$T_{[1,2]}$	$T_{[1,3]}$	$T_{[1,4]}$
2		-	-	$T_{[2,3]}$	$T_{[2,4]}$
3		-	-	-	$T_{[3,4]}$
4		-	-	-	-

$$\begin{aligned}
 & \gamma((b_1, \Delta\theta_1), (b_2, \Delta\theta_2), y, (i, j)) \\
 = & \left\{ \begin{array}{l} \min \left\{ \begin{array}{l} \text{Pfad mit der kleinsten Spannungswinkeldifferenz} \\ \text{bestimmt den Spannungswinkel an der Senke} \\ \left( \frac{\min\{b_1 \cdot \Delta\theta_1, b_2 \cdot \Delta\theta_2\}}{\beta((b_1, \Delta\theta_1), (b_2, \Delta\theta_2), y, (i, j)) - b(i, j)} \right), \frac{\text{tlim}(i, j)}{\underbrace{b(i, j)}_{\Delta\theta(i, j)}} \end{array} \right\}, \text{ falls } y(i, j) = 1 \\ \frac{\min\{b_1 \cdot \Delta\theta_1, b_2 \cdot \Delta\theta_2\}}{\beta((b_1, \Delta\theta_1), (b_2, \Delta\theta_2), y, (i, j))}, \text{ sonst.} \end{array} \right. \quad (5.3)
 \end{aligned}$$

Um die Performance des Algorithmus zu verbessern wird bei den Funktionen  $\beta$  und  $\gamma$  die Technik Memoization verwendet. Dabei werden die Ergebnisse zwischengespeichert und müssen bei einem wiederholten Aufruf mit identischen Argumenten nicht erneut berechnet werden.

$$T_{[i,j]} = \left\{ (\beta(l_1, l_2, y, (i, j)), \gamma(l_1, l_2, y, (i, j))) \mid \forall m \in \{i+1, \dots, j-1\}, l_1 \in T_{[i,m]}, l_2 \in T_{[m,j]} \right\} \quad (5.4)$$

Der maximal mögliche Fluss eines resultierenden Graphen  $G_k = (V, E_0 \cup E_{\text{neu}})$  lässt sich dann durch die Maximierung über die Multiplikationen der jeweiligen Paare eines Tabelleneintrages berechnen.

$$\text{cap}(i, j) = \max\{b(i, j) \cdot \Delta\theta(i, j)\} \quad \forall (b(i, j), \Delta\theta(i, j)) \in T_{[i,j]} \quad (5.5)$$

### 5.2.1 Beispiel

Um die Berechnung der Tabelleneinträge  $T_{[i,j]}$  zu verdeutlichen, sei  $G_1 = (V_1, E_1)$  ein ungerichteter Beispielgraph mit der Knotenmenge  $V_1 = \{1, 2, 3, 4\}$  und der Kantenmenge  $E_1 = \{(1, 2), (2, 3), (3, 4)\}$  wie in Abbildung 5.4 dargestellt. Es gelte  $b(e) = 1$  und  $\text{tlim}(e) = 1$  für alle  $e \in E_1$  und jede hinzugefügte Kante. Die Tabelleneinträge in Tabelle 5.2 lassen sich nun wie in den Gleichungen 5.6 bis 5.11 berechnen. Wobei der initiale Aufruf (Zeile 3 in Algorithmus 5.1) eigentlich die Berechnung des Eintrages  $T_{[1,4]}$  ist und die Berechnungen der weiteren Einträge rekursiv geschehen, wie in Algorithmus 5.2 zu



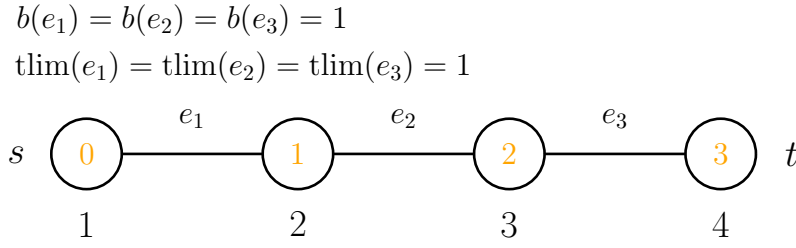


Abbildung 5.4: Ungerichteter Graph  $G_1$  mit Knotenmenge  $V_1 = \{1, 2, 3, 4\}$ , Kantenmenge  $E_1 = \{(1, 2), (2, 3), (3, 4)\}$ , Spannungswinkel  $\theta$  und  $b(e) = 1$  und  $\text{tlim}(e) = 1$  für alle  $e \in E_1$ .

sehen ist. Zur Vereinfachung des Beispiels wird dieser Teil hier weggelassen und lediglich die Aufstiege aus dem Rekursionsbaum gezeigt.

$$T_{[1,2]} = \{(1, 1)\} \quad (5.6)$$

$$T_{[2,3]} = \{(1, 1)\} \quad (5.7)$$

$$T_{[3,4]} = \{(1, 1)\} \quad (5.8)$$

$$\begin{aligned}
T_{[1,3]} &= \left\{ \left( \frac{1}{\frac{1}{1} + \frac{1}{1}} + 0, \frac{\min\{1, 1\}}{\frac{1}{2}} \right), \left( \frac{1}{\frac{1}{1} + \frac{1}{1}} + 1, \min \left\{ \left( \frac{\min\{1, 1\}}{\frac{3}{2} - 1} \right), \frac{1}{1} \right\} \right) \right\} \\
&= \left\{ \left( \frac{1}{2}, 2 \right), \left( \frac{3}{2}, 1 \right) \right\}
\end{aligned} \quad (5.9)$$

$$\begin{aligned}
T_{[2,4]} &= \left\{ \left( \frac{1}{\frac{1}{1} + \frac{1}{1}} + 0, \frac{\min\{1, 1\}}{\frac{1}{2}} \right), \left( \frac{1}{\frac{1}{1} + \frac{1}{1}} + 1, \min \left\{ \left( \frac{\min\{1, 1\}}{\frac{3}{2} - 1} \right), \frac{1}{1} \right\} \right) \right\} \\
&= \left\{ \left( \frac{1}{2}, 2 \right), \left( \frac{3}{2}, 1 \right) \right\}
\end{aligned} \quad (5.10)$$

Die Labelmengen  $T_{[1,3]}$  und  $T_{[2,4]}$  überschneiden sich also. Dadurch ergeben sich im nächsten Schritt doppelte Label, wovon jeweils nur ein Label in die Menge  $T_{[1,4]}$  aufgenommen wird.

$$\begin{aligned}
T_{[1,4]} &= \left\{ \left( \frac{1}{\frac{1}{2} + \frac{1}{1}} + 0, \frac{\min\{1, 1\}}{\frac{1}{3}} \right), \left( \frac{1}{\frac{1}{2} + \frac{1}{1}} + 1, \min \left\{ \left( \frac{\min\{1, 1\}}{\frac{4}{3} - 1} \right), \frac{1}{1} \right\} \right), \right. \\
&\quad \left( \frac{1}{\frac{1}{\frac{3}{2}} + \frac{1}{1}} + 0, \frac{\min\{\frac{3}{2}, 1\}}{\frac{3}{5}} \right), \left( \frac{1}{\frac{1}{\frac{3}{2}} + \frac{1}{1}} + 1, \min \left\{ \left( \frac{\min\{\frac{3}{2}, 1\}}{\frac{8}{5} - 1} \right), \frac{1}{1} \right\} \right), \\
&\quad \left( \frac{1}{\frac{1}{1} + \frac{1}{\frac{1}{2}}} + 0, \frac{\min\{1, 1\}}{\frac{1}{3}} \right), \left( \frac{1}{\frac{1}{1} + \frac{1}{\frac{1}{2}}} + 1, \min \left\{ \left( \frac{\min\{1, 1\}}{\frac{4}{3} - 1} \right), \frac{1}{1} \right\} \right), \\
&\quad \left. \left( \frac{1}{\frac{1}{1} + \frac{1}{\frac{3}{2}}} + 0, \frac{\min\{1, \frac{3}{2}\}}{\frac{3}{5}} \right), \left( \frac{1}{\frac{1}{1} + \frac{1}{\frac{3}{2}}} + 1, \min \left\{ \left( \frac{\min\{1, \frac{3}{2}\}}{\frac{8}{5} - 1} \right), \frac{1}{1} \right\} \right) \right\} \\
&= \left\{ \left( \frac{1}{3}, 3 \right), \left( \frac{4}{3}, 1 \right), \left( \frac{3}{5}, \frac{5}{3} \right), \left( \frac{8}{5}, 1 \right) \right\}
\end{aligned} \quad (5.11)$$

Um  $\text{OPT}_{\text{MPF}}(N)$  zu bestimmen, berechnet man den maximal möglichen Fluss jedes Tabelleneintrages und ermittelt aus diesen Werten das Maximum wie in Gleichung 5.5 angegeben. Für dieses Beispiel sieht das folgendermaßen aus:

Tabelle 5.3: Anzahl der Label, wenn identische Label in die Menge aufgenommen werden und somit Multimengen entstehen. Die gewählte Suszeptanz ist  $b(a, b) = 1$  für alle Kanten  $(a, b)$ .

$ T_{[a,b]} $ a	b	1	2	3	4	5	6	7	8	9	10
1	-	1	2	8	40	224	1344	8448	54912	366080	
2	-	-	1	2	8	40	224	1344	8448	54912	
3	-	-	-	1	2	8	40	224	1344	8448	
4	-	-	-	-	1	2	8	40	224	1344	
5	-	-	-	-	-	1	2	8	40	224	
6	-	-	-	-	-	-	1	2	8	40	
7	-	-	-	-	-	-	-	1	2	8	
8	-	-	-	-	-	-	-	-	1	2	
9	-	-	-	-	-	-	-	-	-	1	
10	-	-	-	-	-	-	-	-	-	-	

$$\begin{aligned}
 \text{OPT}_{\text{MPF}}(N) &= \max \left\{ \left( \frac{1}{3} \cdot 3 \right), \left( \frac{4}{3} \cdot 1 \right), \left( \frac{3}{5} \cdot \frac{5}{3} \right), \left( \frac{8}{5} \cdot 1 \right) \right\} \\
 &= \max \left\{ 1, \frac{4}{3}, 1, \frac{8}{5} \right\} = \frac{8}{5}
 \end{aligned} \tag{5.12}$$

Folgt man nun den Pointern aus dem Label  $(\frac{8}{5}, 1)$ , ist die Menge der hinzugefügten Kanten, abhängig davon welches der identischen Label hinzugefügt wurde, entweder  $E_{\text{neu}} = \{(2, 4), (1, 4)\}$  oder  $E_{\text{neu}} = \{(1, 3), (1, 4)\}$ .

### 5.3 Abschätzung

Wie in Tabelle 5.3 zu sehen, entspricht die Anzahl der Label, wenn identische Label in die Mengen aufgenommen werden und somit Multimengen entstehen, für  $n \leq 10$ , wobei  $n$  die Anzahl der Knoten angibt, der Zahlenfolge  $2^{n-1} * C(n-1)$  [OEI]. Hierbei entspricht  $C(n)$  den Catalan-Zahlen [Kos08]. Diese Beobachtung lässt vermuten, dass die Anzahl der Label in diesen Multimengen für alle  $n \in \mathbb{N}$  dieser Zahlenfolge entspricht. Ein Beweis hierfür wurde noch nicht geführt.

Wie in dem Beispiel 5.2.1 beschrieben, wird bei identischen Label nur eines in die Menge aufgenommen. Dadurch wird die Mächtigkeit der Mengen in einem Eintrag bedeutend reduziert, wie in Tabelle 5.4 zu erkennen.

Um die Menge der Label weiter zu reduzieren, wird ein Dominanzkriterium eingeführt. Angenommen es gibt zwei Label  $l_i = (b_i, \Delta\theta_i)$  und  $l_j = (b_j, \Delta\theta_j)$  mit  $b_i = b_j$ , die zu den Erweiterungen  $G_i$  und  $G_j$  gehören, in einem Tabelleneintrag. Betrachtet man Gleichung 5.2 ist es egal welches der beiden Label, in Kombination mit einem dritten Label  $l_3$ , zur Berechnung eines neuen Labels verwendet wird. Die Suszeptanz  $b$  des neuen Labels ist identisch. Gilt nun auch  $\Delta\theta_i \geq \Delta\theta_j$ , ist der Ausbau  $G_i$  immer mindestens so gut wie der von  $G_j$ . Denn betrachtet man Gleichung 5.4 und wählt  $l_i$  statt  $l_j$  als  $l_1$  gibt es drei Möglichkeiten. Entweder  $b_1 \cdot \Delta\theta_1$  war das Minimum von  $\min\{b_1 \cdot \Delta\theta_1, b_2 \cdot \Delta\theta_2\}$  und ist es mit einem größeren Wert weiterhin oder  $b_1 \cdot \Delta\theta_1$  wird größer als  $b_2 \cdot \Delta\theta_2$  und  $b_2 \cdot \Delta\theta_2$  ist das neue Minimum. In diesen beiden Fällen sowie dem Fall, dass  $b_2 \cdot \Delta\theta_2$  das Minimum war, wird der Zähler des Bruchs und somit auch der gesamte Ausdruck größer oder bleibt gleich. Die Spannungswinkeldifferenz  $\Delta\theta$  des neuen Labels ist, wenn der Ausbau  $G_i$  verwendet

Tabelle 5.4: Anzahl der Label. Die gewählte Suszeptanz ist  $b(a, b) = 1$  für alle Kanten  $(a, b)$ .

$ T_{[a,b]} $ a	b	1	2	3	4	5	6	7	8	9	10
1		-	1	2	4	10	24	68	187	554	1631
2		-	-	1	2	4	10	24	68	187	554
3		-	-	-	1	2	4	10	24	68	187
4		-	-	-	-	1	2	4	10	24	68
5		-	-	-	-	-	1	2	4	10	24
6		-	-	-	-	-	-	1	2	4	10
7		-	-	-	-	-	-	-	1	2	4
8		-	-	-	-	-	-	-	-	1	2
9		-	-	-	-	-	-	-	-	-	1
10		-	-	-	-	-	-	-	-	-	-

Tabelle 5.5: Anzahl der Label, wenn das Dominanzkriterium beachtet wird. Die gewählte Suszeptanz ist  $b(a, b) = 1$  für alle Kanten  $(a, b)$ .

$ T_{[a,b]} $ a	b	1	2	3	4	5	6	7	8	9	10
1		-	1	2	4	10	24	68	186	546	1607
2		-	-	1	2	4	10	24	68	186	546
3		-	-	-	1	2	4	10	24	68	186
4		-	-	-	-	1	2	4	10	24	68
5		-	-	-	-	-	1	2	4	10	24
6		-	-	-	-	-	-	1	2	4	10
7		-	-	-	-	-	-	-	1	2	4
8		-	-	-	-	-	-	-	-	1	2
9		-	-	-	-	-	-	-	-	-	1
10		-	-	-	-	-	-	-	-	-	-

wird, immer mindestens genauso groß wie wenn der Ausbau  $G_j$  verwendet wird. Da sich der maximal mögliche Fluss eines Ausbaus durch Multiplikation von Suszeptanz und Spannungswinkeldifferenz (s. Gleichung 5.5) berechnet, ist der Ausbau  $G_i$  immer mindestens so gut wie der von  $G_j$ . Tritt in einer Lösung  $G_j$  auf, kann  $G_j$  immer durch  $G_i$  ersetzt werden, ohne dass sich der maximal mögliche Fluss im gesamten Netzwerk verschlechtert. Deshalb werden bei identischer Suszeptanz  $b$  der Menge nur die Label hinzugefügt, welche die größere Spannungswinkeldifferenz  $\Delta\theta$  haben. Durch das Dominanzkriterium wird die Anzahl der Label in den Mengen auf die Werte in Tabelle 5.5 reduziert.

Wählt man für alle Kanten  $(i, j) \in E_1$ , welche neu hinzugefügt werden können, die Suszeptanz mit  $b(i, j) = \frac{1}{j-i}$ , erhält man für die Menge der Label die Werte aus Tabelle 5.6. Wird das Dominanzkriterium beachtet, entsteht die Tabelle 5.7.

Tabelle 5.6: Anzahl der Label mit der Suszeptanz  $b(a, b) = \frac{1}{(b-a)}$  für alle Kanten  $(a, b)$ .

$ T_{[a,b]} $ a	b	1	2	3	4	5	6	7	8	9	10
1		-	1	2	4	9	21	54	151	432	1274
2		-	-	1	2	4	9	21	54	151	432
3		-	-	-	1	2	4	9	21	54	151
4		-	-	-	-	1	2	4	9	21	54
5		-	-	-	-	-	1	2	4	9	21
6		-	-	-	-	-	-	1	2	4	9
7		-	-	-	-	-	-	-	1	2	4
8		-	-	-	-	-	-	-	-	1	2
9		-	-	-	-	-	-	-	-	-	1
10		-	-	-	-	-	-	-	-	-	-

Tabelle 5.7: Anzahl der Label, wenn das Dominanzkriterium beachtet wird. Die gewählte Suszeptanz ist  $b(a, b) = \frac{1}{(b-a)}$  für alle Kanten  $(a, b)$ .

$ T_{[a,b]} $ a	b	1	2	3	4	5	6	7	8	9	10
1		-	1	2	4	9	21	54	149	419	1228
2		-	-	1	2	4	9	21	54	149	419
3		-	-	-	1	2	4	9	21	54	149
4		-	-	-	-	1	2	4	9	21	54
5		-	-	-	-	-	1	2	4	9	21
6		-	-	-	-	-	-	1	2	4	9
7		-	-	-	-	-	-	-	1	2	4
8		-	-	-	-	-	-	-	-	1	2
9		-	-	-	-	-	-	-	-	-	1
10		-	-	-	-	-	-	-	-	-	-

---

**Algorithm 5.1:** OPTIMAL MAXIMUM FLOW

---

**Input:** Start node  $i$ , end node  $j$ , standard label  $l_s$

**Output:** List of added edges for the best label  $list$

**Data:** A network  $N = (G, V_c, V_g, tlim, b)$

// Initialization

1  $l_s = (1, 1)$

2  $T_{[i,j]} \leftarrow \emptyset$

//  $\forall i, j \in V$

3  $calc(i, j)$

// siehe Algorithmus 5.2

4  $l_{solution} \leftarrow \max(T_{[i,j]})$  // bestimmt Label mit höchstem  $cap(i, j)$  siehe Gl. 5.5 und gibt es zurück

5  $list \leftarrow trace(l_{solution})$  // folgt Pointern um hinzugefügte Kanten zu ermitteln und gibt diese zurück

6 return  $list$

---

---

**Algorithm 5.2:** FUNKTION CALC

---

**Input:** start node  $i$ , end node  $j$ **Data:** A network  $N = (G, V_c, V_g, tlim, b)$ , Set of label  $T_{[i,j]}$ 

```

1  $k \leftarrow 0$ 
2 if  $j - i = 1$  then
3    $T_{[i,j]}$ .INSERT( $l_s$ )
4 else
5   forall  $k < j - i$  do                                     //  $k \in \mathbb{N}$ 
6     if  $T_{[i,i+k]} = \emptyset$  then
7        $\text{calc}(i, i + k)$ 
8     if  $T_{[i+k,j]} = \emptyset$  then
9        $\text{calc}(i + k, j)$ 
10    forall  $l_1 \in T_{[i,i+k]}$  do
11      forall  $l_2 \in T_{[i+k,j]}$  do
12        forall  $y \in \{0, 1\}$  do
13           $b_{new} \leftarrow \beta(l_1, l_2, y, (i, j))$            // Def. von  $\beta$  siehe Gl. 5.2
14           $\Delta\theta_{new} \leftarrow \gamma(l_1, l_2, y, (i, j))$  // Def. von  $\gamma$  siehe Gl. 5.3
15           $l_{new} \leftarrow (b_{new}, \Delta\theta_{new})$ 
16          if exists  $(b, \Delta\theta) := l$  with  $b == b_{new}$  then //  $l \in T_{[i,j]}$ 
17            if  $\Delta\theta < \Delta\theta_{new}$  then // check Dominanzkriterium
18               $T_{[i,j]}$ .DELETE( $l$ )
19               $T_{[i,j]}$ .INSERT( $l_{new}$ )
20          else
21             $T_{[i,j]}$ .INSERT( $l_{new}$ )

```

---



## 6. Evaluation

Die Berechnungen, auf welchen die folgenden Ergebnisse beruhen, wurden auf einer Maschine mit einem Windows 10 Home 64-Bit-Betriebssystem ausgeführt, welche mit einer Intel Core i7-5500 CPU ausgestattet ist, die mit 2.40 GHz Taktfrequenz arbeitet. Des Weiteren ist in der Maschine 8 GB Arbeitsspeicher verbaut. Das dynamische Programm wurde in Java mittels der Eclipse IDE Version 4.10.0, welche die Java-Laufzeitumgebung Version 1.8.0 verwendet, implementiert. Um die Richtigkeit des dynamischen Programms zu verifizieren, wurde der Gurobi Optimizer Version 8.1.0 verwendet. Das dafür nötige Programm wurde in C++ mit Hilfe von Microsoft Visual Studio Community 2017 Version 15.9.8 implementiert. Der verwendete Compiler dabei ist Visual C++ 2017 Version 15.9 gewesen. Sowohl das dynamische Programm als auch der Gurobi Optimizer hatten ein Zeitlimit von zwei Stunden. Nach Ablauf der zwei Stunden wurde die bis dahin beste Lösung ausgegeben.

Als Eingabegröße  $n$  wurden alle  $n \in \mathbb{N}$  mit  $3 \leq n \leq 25$  verwendet. Dabei gibt  $n$  die Knotenanzahl an.

Sowohl das dynamische Programm als auch der Gurobi Optimizer liefern für alle  $n$  dieselben Ergebnisse für den maximal möglichen Flusswert  $\text{OPT}_{\text{MPF}}(N + E_{\text{neu}})$ . Jedoch liefern sie oftmals unterschiedliche Ergebnisse für  $E_{\text{neu}}$ . Der Grund hierfür ist die in dem Beispiel 5.2.1 gezeigte Tatsache, dass durch unterschiedliche Konfiguration von hinzugefügten Kanten identische Label entstehen können, welche einen identischen maximal möglichen Flusswert aufweisen.

### 6.1 Maximal möglicher Flusswert

Bei der Bestimmung des maximal möglichen Flusswertes  $\text{OPT}_{\text{MPF}}(N + E_{\text{neu}})$  wurden, wie auch bei der Bestimmung der Labelanzahl in Kapitel 5.3, zwei unterschiedliche Werte für die Suszeptanz  $b$  der Kanten gewählt. Der erste Wert ist  $b(i, j) = 1$  für alle  $(i, j) \in E_0 \cup E_1$  und der zweite Wert ist  $b(i, j) = \frac{1}{j-i}$  für alle  $(i, j) \in E_0 \cup E_1$ .

#### 6.1.1 Serien-paralleler Graph

In Abbildung 6.1 ist zu sehen, dass der Verlauf von  $\text{OPT}_{\text{MPF}}(N + E_{\text{neu}})$  sich für die unterschiedliche Suszeptanzen stark unterscheidet. So unterscheiden sich, bei  $b(i, j) = \frac{1}{j-i}$  für alle Kanten, zwei maximal mögliche Flusswerte wenn die Knotendifferenz 1 ist, um den Wert 0 oder 1. Gilt  $b(i, j) = 1$ , ist  $\text{OPT}_{\text{MPF}}(N + E_{\text{neu}})$  stets unterschiedlich. Hierbei nimmt die Differenz, wenn die Knotendifferenz 1 ist, mit zunehmender Knotenanzahl ab.

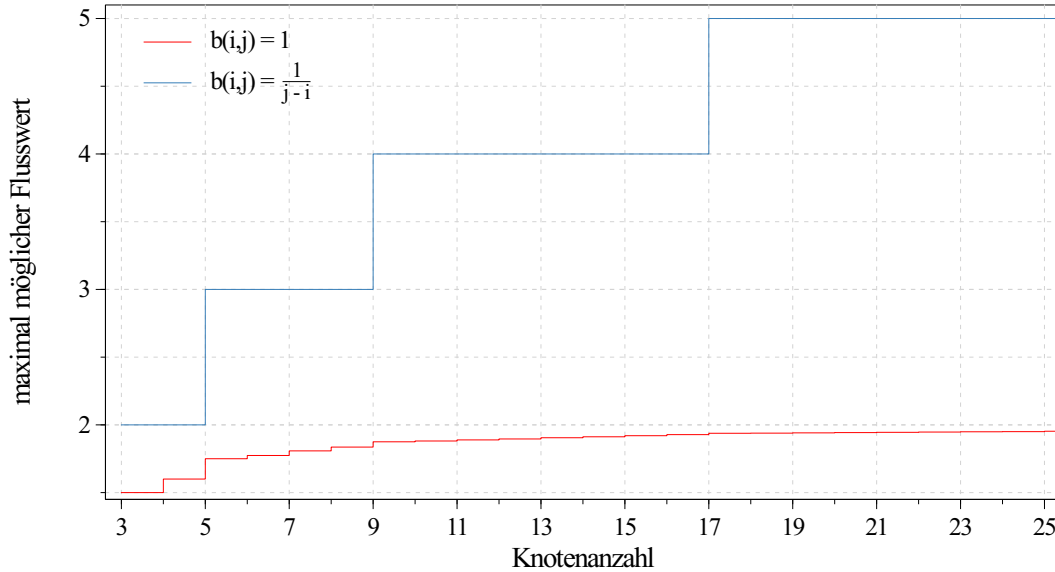


Abbildung 6.1: Maximal mögliche Flusswerte für serien-parallele Graphen mit einer Knotenanzahl  $\geq 3$  und  $\leq 25$ . Die rote Kurve zeigt die Flusswerte bei einer Suszeptanz  $b(i, j) = 1$  für alle Kanten. Wird für die Suszeptanz  $b(i, j) = \frac{1}{j-i}$  für alle Kanten gewählt, ergibt sich die blaue Kurve. Die Werte für eine Knotenanzahl  $> 20$  sind keine optimalen Werte, da bei der Berechnung das Zeitlimit von zwei Stunden überschritten wurde.

Wie in Abbildung 6.2 zu sehen ist, nähert sich  $\text{OPT}_{\text{MPF}}(N + E_{\text{neu}})$  mit zunehmender Knotenanzahl einem Wert zwischen 1,95 und 1,96 an. Eine genaue obere Schranke für  $\text{OPT}_{\text{MPF}}(N + E_{\text{neu}})$ , mit  $b(i, j) = 1$  für alle Kanten, konnte im Zuge dieser Arbeit nicht ermittelt werden, da für eine Knotenanzahl  $> 20$  das Zeitlimit von zwei Stunden überschritten wurde. Mit steigender Knotenanzahl wurde die Abweichung teils so groß, dass ein  $\text{OPT}_{\text{MPF}}(N + E_{\text{neu}})$  ausgegeben wurde, welcher unter den optimalen Werten bei einer Knotenanzahl  $\leq 20$  lag. Einzelne Werte sind dem Anhang A.1 zu entnehmen.

### 6.1.2 Einfacher Graph

Lässt man die in Kapitel 5 eingeführte Restriktion weg, dass der Graph  $G_k = (V, E_0 \cup E_{\text{neu}})$  serien-parallel sein muss, ergibt sich für  $\text{OPT}_{\text{MPF}}(N + E_{\text{neu}})$  ein anderer Wert. In Abbildung 6.3 ist zu sehen, dass sich der maximal mögliche Flusswert, ohne die Beschränkung, für  $b(i, j) = 1$  mit zunehmender Knotenanzahl konstant um 0,5 erhöht und sich für  $b(i, j) = \frac{1}{j-i}$  bei zunehmender Knotenanzahl immer weniger erhöht und scheinbar einer Schranke annähert. Wie in Abbildung 6.1 zu sehen ist, ist dieses Verhalten bei serien-parallel Graphen genau umgekehrt. Einzelne Werte sind in Anhang A.3 zu finden. Des Weiteren ist in Abbildung 6.4 ein Beispiel zu sehen, dass bei  $b(i, j) = 1$  lediglich eine Kante von dem Erzeuger zu jedem anderen Knoten und von jedem Zwischenknoten zur Senke hinzugefügt wird. Über die Kanten des anfangs gegebenen einfachen Pfads fließt, außer auf den Kanten zwischen der Quelle und dem ersten Zwischenknoten sowie dem letzten Zwischenknoten und der Senke, kein Fluss.

## 6.2 Laufzeit

Die Laufzeiten des dynamischen Programms geben, wie in Tabelle 6.1 zu sehen ist, nicht viele Informationen wieder. Für eine Knotenanzahl  $\leq 11$  liegen die Laufzeiten sowohl für  $b(i, j) = 1$  als auch für  $b(i, j) = \frac{1}{j-i}$  im niedrigen dreistelligen Millisekundenbereich. Danach



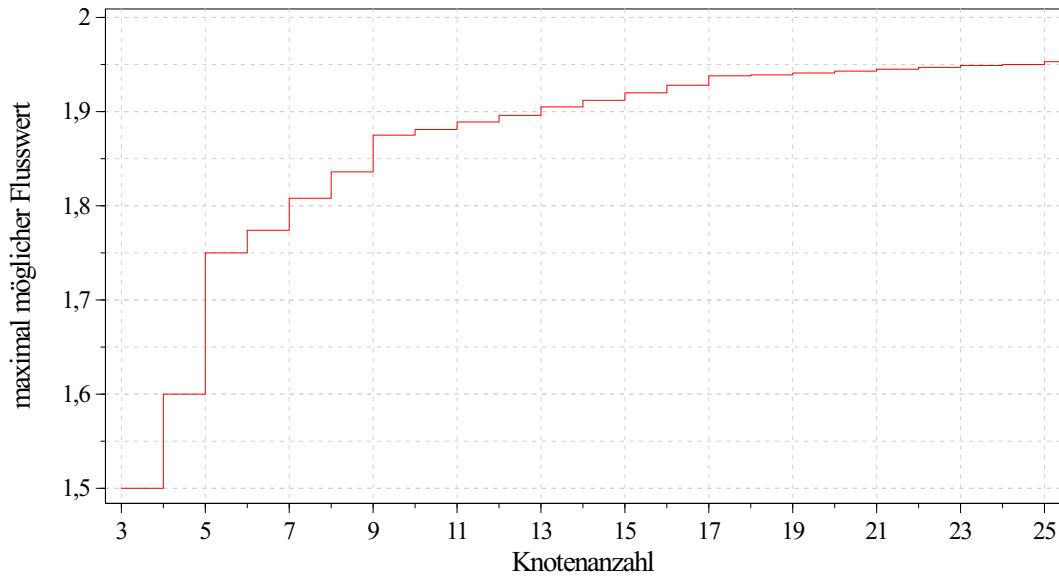


Abbildung 6.2: Maximal mögliche Flusswerte für serien-parallele Graphen mit einer Knotenanzahl  $\geq 3$  und  $\leq 25$ . Die gewählte Suszeptanz  $b$  für alle Kanten ist 1. Die Werte für eine Knotenanzahl  $> 20$  sind keine optimalen Werte, da bei der Berechnung das Zeitlimit von zwei Stunden überschritten wurde.

erhöhten sich die Laufzeiten jeweils stark und stellenweise um einen Faktor von 15 bis über 17. Ab einer Knotenanzahl von  $\geq 15$  waren im Zuge dieser Arbeit keine Laufzeiten mehr messbar, da das Zeitlimit von zwei Stunden überschritten wurde. Es ist jedoch zu erkennen, dass für eine Knotenanzahl  $\geq 12$  die Laufzeit für  $b(i, j) = \frac{1}{j-i}$  stets geringer war als für  $b(i, j) = 1$ .

Diese Beobachtung kann auch bei den Laufzeiten des Gurobi Optimizer gemacht werden. Bei  $b(i, j) = \frac{1}{j-i}$  betrug die Laufzeit zur Berechnung einer optimalen Lösung für einen Graphen mit 25 Knoten ungefähr 8,579 Sekunden (weitere Werte sind Anhang A.2 aufgelistet), wohingegen die Messung der Laufzeit bei  $b(i, j) = 1$  ab 21 Knoten nicht mehr möglich war, da das Zeitlimit überschritten wurde. Auch erhöhte sich die Laufzeit, bei Erhöhung der Knotenanzahl um 1, stellenweise um einen Faktor von 1 bis über 4,8 wie in Tabelle 6.2 zu sehen ist. Wurde für die Suszeptanz  $b(i, j) = \frac{1}{j-i}$  gewählt, erhöhte sich die Laufzeit, bei Erhöhung der Knotenanzahl um 1, fast ausschließlich um einen Faktor der kleiner als 2 war.

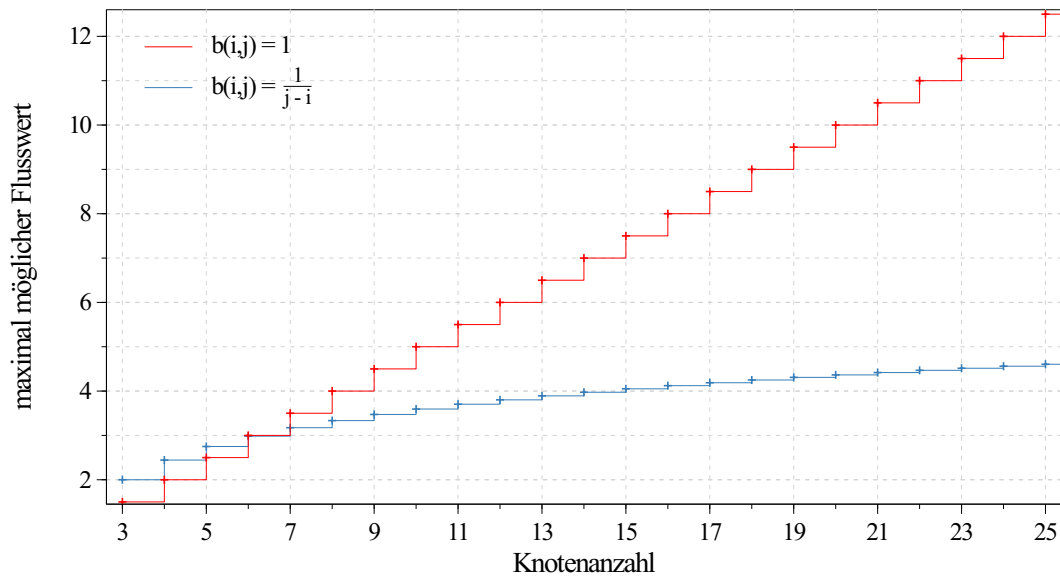


Abbildung 6.3: Maximal mögliche Flusswerte für einfache Graphen mit einer Knotenanzahl  $\geq 3$  und  $\leq 25$ . Die **rote** Kurve zeigt die Flusswerte bei einer Suszeptanz  $b(i, j) = 1$  für alle Kanten. Wird für die Suszeptanz  $b(i, j) = \frac{1}{j-i}$  für alle Kanten gewählt, ergibt sich die **blaue** Kurve.

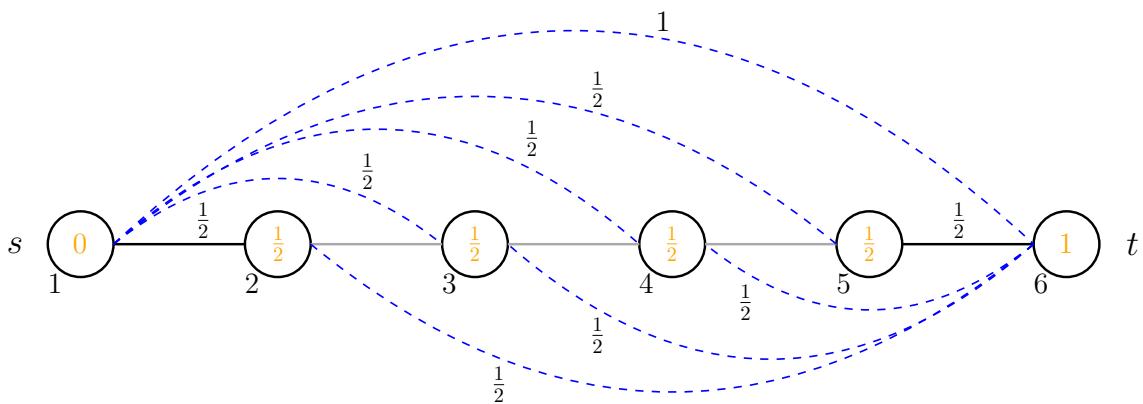


Abbildung 6.4: Einfacher Graph  $G_k$  mit Knotenmenge  $V = \{1, 2, 3, 4, 5, 6\}$ , Kantenmenge  $E_0 \cup E_{\text{neu}} = \{(1, 3), (1, 4), (1, 5), (1, 6), (2, 6), (3, 6), (4, 6), (5, 6)\}$  mit  $b(e) = 1$  und  $\text{tlim}(e) = 1 \forall e \in E_0 \cup E_{\text{neu}}$  und Spannungswinkel  $\theta$ . Auf den Kanten  $(2, 3)$ ,  $(3, 4)$  und  $(4, 5)$  fließt kein Fluss. Die Zahlen an den Kanten geben jeweils den Fluss auf der jeweiligen Kante an.

Tabelle 6.1: Laufzeiten des dynamischen Programms in Sekunden zur Berechnung einer optimalen Lösung des maximal möglichen Flusswertes für serien-parallele Graphen  $G_k$  mit einer Knotenanzahl von  $\geq 3$  und  $\leq 14$  und einer Suszeptanz  $b(i, j) = 1$  oder  $b(i, j) = \frac{1}{j-i} \forall (i, j) \in E_0 \cup E_{\text{neu}}$ .

Knotenanzahl	Laufzeit in Sekunden mit $b(i, j) = 1$	Laufzeit in Sekunden mit $b(i, j) = \frac{1}{j-i}$
3	0,121	0,14
4	0,107	0,126
5	0,114	0,127
6	0,122	0,123
7	0,127	0,139
8	0,133	0,136
9	0,149	0,167
10	0,19	0,191
11	0,444	0,345
12	3,048	1,747
13	51,206	26,449
14	888,734	435,019

Tabelle 6.2: Laufzeiten des Gurobi Optimizer in Sekunden zur Berechnung einer optimalen Lösung des maximal möglichen Flusswertes für serien-parallele Graphen  $G_k$  mit einer Knotenanzahl von  $\geq 3$  und  $\leq 20$  und einer Suszeptanz  $b(i, j) = 1 \forall (i, j) \in E_0 \cup E_{\text{neu}}$ .

Knotenanzahl	Laufzeit in Sekunden
3	0,003
4	0,008
5	0,032
6	0,060
7	0,065
8	0,100
9	0,141
10	0,183
11	0,499
12	1,064
13	4,046
14	11,887
15	30,276
16	32,0629
17	154,944
18	504,150
19	1790,691
20	5209,435

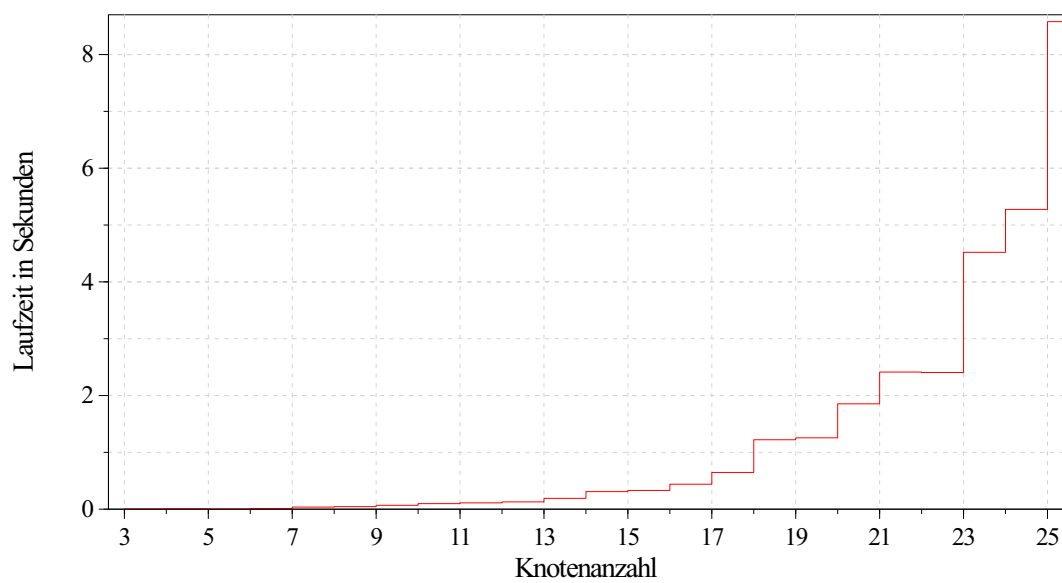


Abbildung 6.5: Laufzeit in Sekunden für die Berechnung einer optimalen Lösung mit dem Gurobi Optimizer für einen maximal möglichen Flusswert bei Graphen mit einer Knotenanzahl  $\geq 3$  und  $\leq 25$ . Dabei gilt für die Suszeptanz aller Kanten  $b(i, j) = \frac{1}{j-i}$ .

## 7. Zusammenfassung

In dieser Arbeit wurde ein Algorithmus zur Berechnung optimaler Erweiterungen für elektrische Netze vorgestellt. Die Berechnung wurde durch ein dynamisches Programm realisiert. Es wurden zwei Varianten für die Suszeptanz verwendet und zwei Graphenklassen betrachtet. Um die Ergebnisse des dynamischen Programms zu verifizieren und die Performance zu bewerten, wurde das Optimierungsproblem zusätzlich mit dem Gurobi Optimizer gelöst. Es wurde deutlich, dass das dynamische Programm dieselben Ergebnisse wie der Gurobi Optimizer liefert. Wurde die Graphenklasse der serien-parallelen Graphen betrachtet und die Suszeptanz  $b(i, j) = 1$  gewählt, nähert sich der maximal mögliche Flusswert, mit zunehmender Knotenanzahl, scheinbar einem Wert  $< 2$  an. Wurde die Suszeptanz  $b(i, j) = \frac{1}{j-i}$  gewählt, steigt der maximal mögliche Flusswert, mit zunehmender Knotenanzahl, immer wieder sprunghaft an oder bleibt für ein Intervall von Knotenanzahlen gleich. Bei Betrachtung von einfachen Graphen sind dieselben Beobachtungen, jedoch umgekehrt gemacht worden.

Bei den Laufzeiten war zu erkennen, dass für die Berechnung, bei einer gewählten Suszeptanz  $b(i, j) = \frac{1}{j-i}$  für alle Kanten, die Laufzeit geringer war als für die gewählte Suszeptanz  $b(i, j) = 1$ . Es wurde auch deutlich, dass das dynamische Programm eine höhere Laufzeit als der Gurobi Optimizer aufweist.

Es wurde nur ein Optimierungsproblem betrachtet. In weiteren Arbeiten kann das dynamische Programm an andere Optimierungsprobleme angepasst oder für das betrachtete Problem erweitert werden. So wurde im Zuge dieser Arbeit lediglich ein maximal möglicher Flusswert berechnet. Um die eigentliche Problemstellung bei TNEP, die anfallenden Kosten für eine Erweiterung zu minimieren, zu lösen, können für alle gefundenen Erweiterungen, mit einem optimalen maximal möglichen Flusswert, die Kosten berechnet und somit die Erweiterung mit den geringsten Kosten bestimmt werden. Des Weiteren können die Berechnungen auf einer leistungsstärkeren Maschine durchgeführt werden, um Daten für größere Graphen zu erhalten und somit präzisere Beobachtungen machen zu können. Außerdem bestünde die Möglichkeit, weitere Varianten für die Suszeptanz zu wählen und komplexere Graphenklassen zu betrachten.

Natürlich gibt es Möglichkeiten, sowohl den Algorithmus als auch das dynamische Programm selbst zu verbessern. Mit guten Programmierkenntnissen kann sich die Performance des Programms sicher erhöhen lassen. Darüber hinaus kann die Anzahl der berechneten Label vermutlich weiter reduziert und somit auch die Performance verbessert werden.



# Literaturverzeichnis

- [ALH01] T. Ackermann, R. Leutz und J. Hobohm: *World-wide offshore wind potential and European projects*. In: *2001 Power Engineering Society Summer Meeting. Conference Proceedings (Cat. No.01CH37262)*, Band 1, Seiten 4–9 vol.1, July 2001.
- [AMC03] N. Alguacil, A. L. Motto und A. J. Conejo: *Transmission expansion planning: a mixed-integer LP approach*. *IEEE Transactions on Power Systems*, 18(3):1070–1077, Aug 2003, ISSN 0885-8950.
- [BCGH14] R. Bent, C. Coffrin, R. R. E. Gumucio und P. Van Hentenryck: *Transmission Network Expansion Planning: Bridging the gap between AC heuristics and DC approximations*. In: *2014 Power Systems Computation Conference*, Seiten 1–8, Aug 2014.
- [BdOdA01] S. Binato, G. C. de Oliveira und J. L. de Araujo: *A greedy randomized adaptive search procedure for transmission expansion planning*. *IEEE Transactions on Power Systems*, 16(2):247–253, May 2001, ISSN 0885-8950.
- [BTB12] R. Bent, G. L. Toole und A. Berscheid: *Transmission Network Expansion Planning With Complex Power Flow Models*. *IEEE Transactions on Power Systems*, 27(2):904–912, May 2012, ISSN 0885-8950.
- [BV15] Daniel Bienstock und Abhinav Verma: *Strong NP-hardness of AC power flows feasibility*. Dezember 2015.
- [DE73] Y. P. Dusonchet und A. El-Abiad: *Transmission Planning Using Discrete Dynamic Optimizing*. *IEEE Transactions on Power Apparatus and Systems*, PAS-92(4):1358–1371, July 1973, ISSN 0018-9510.
- [dJSRR<sup>+</sup>05] I. de J Silva, M. J. Rider, R. Romero, A. V. Garcia und C. A. Murari: *Transmission network expansion planning with security constraints*. *IEE Proceedings - Generation, Transmission and Distribution*, 152(6):828–836, Nov 2005, ISSN 1350-2360.
- [Gar70] L. L. Garver: *Transmission Network Estimation Using Linear Programming*. *IEEE Transactions on Power Apparatus and Systems*, PAS-89(7):1688–1697, Sep. 1970, ISSN 0018-9510.
- [HHK13] Reza Hemmati, Rahmat Allah Hooshmand und Amin Khodabakhshian: *State-of-the-art of transmission expansion planning: Comprehensive review*. *Renewable and Sustainable Energy Reviews*, 23:312 – 319, 2013, ISSN 1364-0321. <http://www.sciencedirect.com/science/article/pii/S1364032113001743>.
- [HMG<sup>+</sup>00] S. Haffner, A. Monticelli, A. Garcia, J. Mantovani und R. Romero: *Branch and bound algorithm for transmission system expansion planning using a transportation model*. *IEE Proceedings - Generation, Transmission and Distribution*, 147(3):149–156, May 2000, ISSN 1350-2360.

- [Hol92] John H. Holland: *Genetic Algorithms*. Scientific American, 267(1):66–73, 1992, ISSN 00368733, 19467087. <http://www.jstor.org/stable/24939139>.
- [Jab13] R. A. Jabr: *Optimization of AC Transmission System Planning*. IEEE Transactions on Power Systems, 28(3):2779–2787, Aug 2013, ISSN 0885-8950.
- [Kos08] Thomas Koshy: *Catalan Numbers with Applications*. OXFORD UNIV PR, 2008, ISBN 978-0-19-533454-8.
- [LCAV03] G. Latorre, R. D. Cruz, J. M. Areiza und A. Villegas: *Classification of publications and models on transmission expansion planning*. IEEE Transactions on Power Systems, 18(2):938–946, May 2003, ISSN 0885-8950.
- [MAAR18] M. Mahdavi, C. Sabillon Antunez, M. Ajalli und R. Romero: *Transmission Expansion Planning: Literature Review and Classification*. IEEE Systems Journal, Seiten 1–12, 2018, ISSN 1932-8184.
- [MPS10] Luciano S. Moulin, Michael Poss und Claudia Sagastizábal: *Transmission expansion planning with re-design*. Energy Systems, 1(2):113–139, May 2010, ISSN 1868-3975. <https://doi.org/10.1007/s12667-010-0010-9>.
- [OCB95] G. C. Oliveira, A. P. C. Costa und S. Binato: *Large scale transmission network planning using optimization and heuristic techniques*. IEEE Transactions on Power Systems, 10(4):1828–1834, Nov 1995, ISSN 0885-8950.
- [OCS04] T. J. Overbye, Xu Cheng und Yan Sun: *A comparison of the AC and DC power flow models for LMP calculations*. In: *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*, Seiten 9 pp.–, Jan 2004.
- [OEI] *OEIS Foundation Inc. (2018), The On-Line Encyclopedia of Integer Sequences*. <http://oeis.org/A151374>.
- [PMD05] K. Purchala, L. Meeus, D. Van Dommelen und R. Belmans: *Usefulness of DC power flow for active power flow analysis*. In: *IEEE Power Engineering Society General Meeting, 2005*, Seiten 454–459 Vol. 1, June 2005.
- [RGR07] M. J. Rider, A. V. Garcia und R. Romero: *Power system transmission network expansion planning using AC model*. IET Generation, Transmission Distribution, 1(5):731–742, Sep. 2007, ISSN 1751-8687.
- [RM94] R. Romero und A. Monticelli: *A zero-one implicit enumeration method for optimizing investments in transmission expansion planning*. IEEE Transactions on Power Systems, 9(3):1385–1391, Aug 1994, ISSN 0885-8950.
- [RMGH02] R. Romero, A. Monticelli, A. Garcia und S. Haffner: *Test systems and mathematical models for transmission network expansion planning*. IEE Proceedings - Generation, Transmission and Distribution, 149(1):27–36, Jan 2002, ISSN 1350-2360.
- [RRCR10] M. Rahmani, M. Rashidinejad, E.M. Carreno und R. Romero: *Efficient method for AC transmission network expansion planning*. Electric Power Systems Research, 80(9):1056 – 1064, 2010, ISSN 0378-7796. <http://www.sciencedirect.com/science/article/pii/S0378779610000398>.
- [RRMS05] R. Romero, C. Rocha, J. R. S. Mantovani und I. G. Sanchez: *Constructive heuristic algorithm for the DC model in network transmission expansion planning*. IEE Proceedings - Generation, Transmission and Distribution, 152(2):277–282, March 2005, ISSN 1350-2360.



- [SA85] A. Sharifnia und H. Z. Aashtiani: *Transmission Network Planning: A Method for Synthesis of Minimum-Cost Secure Networks*. IEEE Transactions on Power Apparatus and Systems, PAS-104(8):2025–2034, Aug 1985, ISSN 0018-9510.
- [SGA00] E. L. Da Silva, H. A. Gil und J. M. Areiza: *Transmission network expansion planning under an improved genetic algorithm*. IEEE Transactions on Power Systems, 15(3):1168–1174, Aug 2000, ISSN 0885-8950.
- [SJA09] B. Stott, J. Jardim und O. Alsac: *DC Power Flow Revisited*. IEEE Transactions on Power Systems, 24(3):1290–1300, August 2009, ISSN 0885-8950.
- [ZMS11] R. Zimmermann und C. Murillo-Sanchez: *Matpower 4.1 User Manual*. Dec 2011.
- [ZMT09] R. D. Zimmerman, C. E. Murillo-Sanchez und R. J. Thomas: *MATPOWER's extensible optimal power flow architecture*. In: *2009 IEEE Power Energy Society General Meeting*, Seiten 1–7, July 2009.



# Anhang

Im Anhang werden einige Tabellen und eine Abbildung aufgeführt, die den Evaluationsteil 6 ergänzen.

Abbildung A.1 zeigt einen Graph  $G_k$  einer optimalen Lösung bei einer Knotenmenge  $V = \{1, 2, 3, 4, 5, 6\}$  und Suszeptanz  $b(e) = 1$ . In Kapitel 5.1 wurden zwei Varianten (Abbildungen 5.2 und 5.3) zum Ausbau mit dieser Knotenmenge und Suszeptanz gezeigt. Wie zu sehen ist, war keine dieser instinktiv gewählten Lösungen eine optimale Lösung.

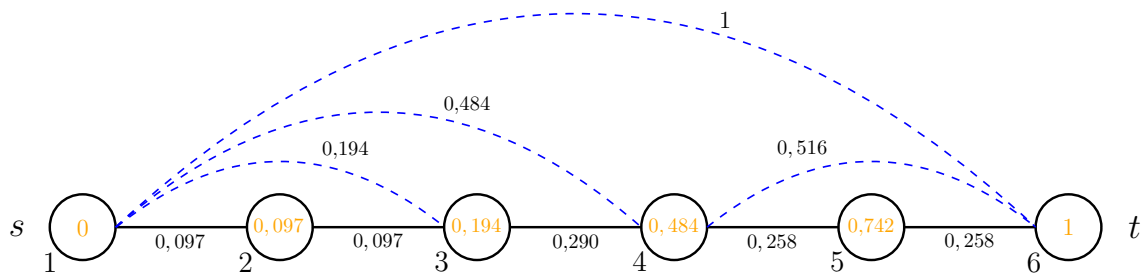


Abbildung A.1: Serien-paralleler Graph  $G_k$  mit Knotenmenge  $V = \{1, 2, 3, 4, 5, 6\}$ , Kantenmenge  $E_0 \cup E_{\text{neu}} = \{(1, 3), (1, 4), (1, 6), (4, 6)\}$  mit  $b(e) = 1$  und  $\text{tlim}(e) = 1 \forall e \in E_0 \cup E_{\text{neu}}$  und Spannungswinkel  $\theta$ . Die Zahlen an den Kanten geben jeweils den Fluss auf der jeweiligen Kante an. Alle Werte wurden auf die dritte Nachkommastelle gerundet.

Tabelle A.1: Maximal mögliche Flusswerte für die Suszeptanz  $b(i, j) = 1$  und  $b(i, j) = \frac{1}{j-i}$  für alle Kanten. Die Werte bei  $b(i, j) = 1$  ab einer Knotenanzahl von 21 sind keine optimalen Lösungen, da das Zeitlimit überschritten wurde. Die relative Abweichung zu einer oberen Schrank beträgt bei 21 Knoten 4,316 %, bei 22 Knoten 19,163 %, bei 23 Knoten 28,290 %, bei 24 Knoten 34,415 % und bei 25 Knoten 34,989 %. Alle Werte wurden auf die dritte Nachkommastelle gerundet.

Knotenanzahl	maximal möglicher Flusswert mit $b(i, j) = 1$	maximal möglicher Flusswert mit $b(i, j) = \frac{1}{j-i}$
3	1,5	2
4	1,6	2
5	1,75	3
6	1,774	3
7	1,808	3
8	1,836	3
9	1,875	4
10	1,881	4
11	1,889	4
12	1,896	4
13	1,905	4
14	1,912	4
15	1,92	4
16	1,928	4
17	1,938	5
18	1,939	5
19	1,941	5
20	1,943	5
21	1,945	5
22	1,947	5
23	1,949	5
24	1,95	5
25	1,953	5

---

Tabelle A.2: Laufzeiten des Gurobi Optimizer mit einer gewählten Suszeptanz  $b(i, j) = \frac{1}{j-i}$ .  
Die Werte wurden auf die dritte Nachkommastelle gerundet.

Knotenanzahl	Laufzeit in Sekunden
3	0,003
4	0,007
5	0,003
6	0,011
7	0,037
8	0,045
9	0,068
10	0,099
11	0,111
12	0,128
13	0,190
14	0,309
15	0,330
16	0,439
17	0,646
18	1,222
19	1,256
20	1,954
21	2,413
22	2,405
23	4,520
24	5,275
25	8,579

Tabelle A.3: Maximal mögliche Flusswerte für die Suszeptanz  $b(i, j) = 1$  und  $b(i, j) = \frac{1}{j-i}$  für alle Kanten, wenn die Bedingungen eines serien-parallelen Graphen weggelassen werden. Die Werte wurden alle auf die dritte Nachkommastelle gerundet.

Knotenanzahl	maximal möglicher Flusswert mit $b(i, j) = 1$	maximal möglicher Flusswert mit $b(i, j) = \frac{1}{j-i}$
3	1,5	2
4	2	2,444
5	2,5	2,75
6	3	2,984
7	3,5	3,174
8	4	3,334
9	4,5	3,473
10	5	3,595
11	5,5	3,704
12	6	3,802
13	6,5	3,892
14	7	3,974
15	7,5	4,051
16	8	4,122
17	8,5	4,188
18	9	4,25
19	9,5	4,309
20	10	4,365
21	10,5	4,418
22	11	4,468
23	11,5	4,515
24	12	4,561
25	12,5	4,605