

vs

Bachelorarbeit

Stückweise lineare Funktionen für Network Calculus

Peter von Zameck Glyscinski

Studiengang: Informatik
Prüfer/in: Prof. Dr. Kurt Rothermel
Betreuer/in: M.Sc. Jonathan Falk

Beginn am: 15. Dezember 2017
Beendet am: 15. Juni 2018

Kurzfassung

Network Calculus ist ein häufig verwendetes und mächtiges Tool, um Aussagen über die Performanz und mögliche Probleme in Kommunikationsnetzwerken treffen zu können. Mit Hilfe dieser Aussagen zur Performanz eines Kommunikationsnetzwerkes ist es möglich, vorab Anforderungen an ein Kommunikationsnetzwerk zu stellen und dessen Aufbau richtig zu planen. Es ist dadurch auch möglich die maximale Belastung eines bestehenden Kommunikationsnetzwerkes zu bestimmen, ohne es einem tatsächlichen Stresstest unterziehen zu müssen.

Das mathematische Grundgerüst des Network Calculus basiert dabei auf der $(\min,+)$ Algebra. Um die einzelnen Operationen der $(\min,+)$ Algebra berechnen zu können, bedarf es einer Darstellung der einzelnen Ankunftscurven und Servicecurven in dem Kommunikationsnetzwerk. Eine mögliche Art diese Ankunftscurven und Servicecurven darzustellen, sind stückweise lineare Funktionen.

In dieser Arbeit wird basierend auf der Arbeit von Bouillard und Thierry [BT08] eine Pythonbibliothek für die effiziente Berechnung mit stückweisen linearen Funktionen in der $(\min,+)$ Algebra erstellt. Die richtige Datenstruktur zum Darstellen einer stückweisen linearen Funktion, sowie Algorithmen zum Berechnen der Operationen in der $(\min,+)$ Algebra, werden erläutert. Wichtige Berechnungen wie die Faltung und Entfaltung, das Minimum und Maximum und die Addition und Subtraktion, sowie das Backlog und das Virtual Delay von zwei stückweisen linearen Funktionen werden dabei unterstützt.

Inhaltsverzeichnis

1	Einleitung	13
2	Hintergrund und Motivation	15
2.1	Network Calculus	15
2.2	Network Calculus Bibliotheken	16
3	Stückweise lineare Funktionen	19
3.1	Definitionen	19
3.2	Datenstruktur	23
3.3	Algorithmen	25
3.4	Implementierung	40
4	Evaluation	43
4.1	Testdaten	43
4.2	Testergebnisse	43
5	Zusammenfassung und Ausblick	65
	Literaturverzeichnis	69

Abbildungsverzeichnis

3.1	Faltung von zwei Segmenten f_1 und f_2	22
3.2	Entfaltung von zwei Segmenten f_1 und f_2	23
3.3	Ein lineares Teilstück mit seinen Parametern $x_1, y, yPlus, p$ und l	24
3.4	Aufbau der Datenstruktur zum halten einer linearen Funktion.	24
3.5	Eine stückweise lineare Funktion mit ihren Parametern T, d und c	25
3.6	Die stückweise lineare Funktion f_1 (rot) mit ihren Parametern T_1 und d_1	27
3.7	Die stückweise lineare Funktion f_2 (blau) mit ihren Parametern T_2 und d_2	27
3.8	f_1 und f_2 verlängert.	27
3.9	Ergebnis der Addition von f_1 und f_2 in grün mit neuen Parametern T und d	28
3.10	Die stückweise lineare Funktion f_1 (rot) mit ihren Parametern T_1 und d_1	30
3.11	Die stückweise lineare Funktion f_2 (blau) mit ihren Parametern T_2 und d_2	30
3.12	f_1 und f_2 verlängert.	31
3.13	Das Minimum von f_1 und f_2 in grün mit neuen Parametern T und d	31
3.14	Die stückweise lineare Funktion f_1 (rot) mit ihren Parametern T_1 und d_1	33
3.15	Die stückweise lineare Funktion f_2 (blau) mit ihren Parametern T_2 und d_2	33
3.16	Die einzelnen Faltungen der Segmente und Spots in schwarz und das Minimum dieser Faltungen in grün. a) ist dabei die Faltungen von $f_1' * f_2'$, b) die Faltungen von $f_1' * f_2''$, c) die Faltungen von $f_1'' * f_2'$ und d) die Faltungen von $f_1'' * f_2''$	34
3.17	Die einzelnen Faltungen in schwarz und das Minimum über diese Faltungen in grün.	35
3.18	Die Faltung von f_1 und f_2 in grün mit neuen Parametern T und d	36
3.19	Die stückweise lineare Funktion f_1 (rot) mit ihren Parametern T_1 und d_1	37
3.20	Die stückweise lineare Funktion f_2 (blau) mit ihren Parametern T_2 und d_2	37
3.21	Die einzelnen Entfaltungen von Segmenten und Spots in schwarz und das Maximum über diese Entfaltungen in grün.	38
3.22	Die Entfaltung von f_1 und f_2 in grün mit neuen Parametern T und d	38
3.23	Aufbau der Implementierung mit den wichtigsten Elementen.	40
4.1	Laufzeit für die Addition für eine variierende Anzahl von linearen Teilstücken in den stückweisen linearen Funktionen.	45
4.2	Laufzeit für das Minimum für eine variierende Anzahl von linearen Teilstücken in den stückweisen linearen Funktionen.	46
4.3	Laufzeit für die Faltung für eine variierende Anzahl von linearen Teilstücken in den stückweisen linearen Funktionen, für eine nicht periodische Länge von 100 a) und eine periodische Länge von 50 b)	47
4.4	Laufzeit für die Entfaltung für eine variierende Anzahl von linearen Teilstücken in den stückweisen linearen Funktionen.	48
4.5	Laufzeit für die Faltung für eine variierende Länge der stückweisen linearen Funktionen.	56

4.6 Laufzeit für die Faltung für eine variierende Länge der stückweisen linearen Funktionen. 57

Tabellenverzeichnis

4.1	Ergebnisse für eine variierende Anzahl linearer Teilstücke im nicht periodischen Teil bei der Addition in ms.	49
4.2	Ergebnisse für eine variierende Anzahl linearer Teilstücke im periodischen Teil bei der Addition in ms.	50
4.3	Ergebnisse für eine variierende Anzahl linearer Teilstücke im nicht periodischen Teil beim Minimum in ms.	51
4.4	Ergebnisse für eine variierende Anzahl linearer Teilstücke im periodischen Teil beim Minimum in ms.	52
4.5	Ergebnisse für eine variierende Anzahl linearer Teilstücke im nicht periodischen Teil bei der Faltung in s.	53
4.6	Ergebnisse für eine variierende Anzahl linearer Teilstücke im periodischen Teil bei der Faltung in s.	53
4.7	Ergebnisse für eine variierende Anzahl linearer Teilstücke im nicht periodischen Teil bei der Entfaltung in s.	54
4.8	Ergebnisse für eine variierende Anzahl linearer Teilstücke im periodischen Teil bei der Entfaltung in s.	54
4.9	Ergebnisse für eine variierende Länge des nicht periodischen Teils bei der Addition in ms.	58
4.10	Ergebnisse für eine variierende Länge des periodischen Teils bei der Addition in ms.	59
4.11	Ergebnisse für eine variierende Länge des nicht periodischen Teils beim Minimum in ms.	60
4.12	Ergebnisse für eine variierende Länge des periodischen Teils beim Minimum in ms.	61
4.13	Ergebnisse für eine variierende Länge des nicht periodischen Teils bei der Faltung in s.	62
4.14	Ergebnisse für eine variierende Länge des periodischen Teils bei der Faltung in s.	62
4.15	Ergebnisse für eine variierende Länge des nicht periodischen Teils bei der Entfaltung in s.	63
4.16	Ergebnisse für eine variierende Länge des periodischen Teils bei der Entfaltung in s.	63

Verzeichnis der Algorithmen

3.1	Addition von zwei stückweisen linearen Funktionen [BT08]	26
3.2	Minimum von zwei stückweisen linearen Funktionen [BT08]	30
3.3	Maximum von zwei stückweisen linearen Funktionen	32
3.4	Faltung von zwei stückweisen linearen Funktionen [BT08]	35
3.5	Entfaltung von zwei stückweisen linearen Funktionen [BT08]	37

1 Einleitung

Computernetzwerke sind heutzutage in einer großen Vielzahl vorhanden und können riesige Ausmaße annehmen. Um diese Netzwerke und deren Kommunikation stabil und robust zu konstruieren und zu planen, bedarf es ein tieferes Verständnis des dort herrschenden Netzwerkflusses und möglichen Extremfällen im Netzwerk. Network Calculus ist ein häufig verwendetes und mächtiges Tool, um solche Aussagen über die Performanz und mögliche Probleme in solchen Kommunikationsnetzwerken treffen zu können. Das mathematische Grundgerüst des Network Calculus basiert dabei auf der sogenannten $(\min,+)$ Algebra.

Definition 1.0.1

$(\min,+)$ Algebra [LT01]

Die $(\min,+)$ Algebra ist ein mathematischer Halbring über $(\mathbb{R} \cup +\infty, \min, +)$, wobei \min das Minimum und $+$ die Addition auf den reellen Zahlen \mathbb{R} darstellen.

Mit Hilfe der $(\min,+)$ Algebra ist es möglich, verschiedene Performanzmetriken wie etwa das Backlog oder das Virtual Delay [LT01] eines Systems bestimmen zu können. Mit Hilfe dieser Performanzmetriken können anschließend Aussagen zu den Anforderungen an ein Kommunikationsnetzwerk getroffen werden.

Um die einzelnen Operationen der $(\min,+)$ Algebra berechnen zu können, bedarf es einer Darstellung der einzelnen Ankunftscurven und Servicecurven in dem Kommunikationsnetzwerk. Diese Darstellung sollte dabei möglichst effizient sein, was die Berechnung der einzelnen Operationen der $(\min,+)$ Algebra angeht und nicht zu kompliziert umzusetzen sein. Eine mögliche Art diese Ankunftscurven und Servicecurven darzustellen, sind stückweise lineare Funktionen.

In dieser Arbeit wird basierend auf dem Paper von [BT08] eine Pythonbibliothek für die effiziente Berechnung mit stückweisen linearen Funktionen in der $(\min,+)$ Algebra implementiert. Die richtige Datenstruktur zum Darstellen einer stückweisen linearen Funktion, sowie Algorithmen zum Berechnen der Operationen in der $(\min,+)$ Algebra werden vorgestellt. Wichtige Berechnungen, wie die Faltung und Entfaltung, das Minimum und Maximum und die Addition und Subtraktion, sowie das Backlog und das Virtual Delay von zwei stückweisen linearen Funktionen werden dabei unterstützt. Eine weitere wichtige Eigenschaft der stückweisen linearen Funktionen dieser Bibliothek ist, dass sie pseudo-periodisch sein können und so Servicecurven welche durch zeitgesteuerte Tore entstehen können (siehe Kapitel 2.1) unterstützen.

Nach dieser Einleitung werden wir zunächst in Kapitel 2 einen genaueren Blick auf die Anwendung der $(\min,+)$ Algebra im Network Calculus werfen. Anschließend werden in Kapitel 2.2 bereits vorhandene Software Implementierungen für Network Calculus gezeigt, mit denen man in der $(\min,+)$ Algebra verschiedene Operationen berechnen kann und welche Vor- und Nachteile sie mit sich bringen.

In Kapitel 3 wird der Hauptteil dieser Arbeit zusammengefasst. Dazu werden stückweise lineare

Funktionen zunächst in Kapitel 3.1 rein formal definiert und einige wichtige Lemmas und Definitionen aufgestellt, die für den weiteren Verlauf der Arbeit relevant sind. In Kapitel 3.2 wird eine geeignete Datenstruktur gefunden, um stückweise lineare Funktionen darzustellen und auch Algorithmen auf ihnen ausführen zu können. Kapitel 3.3 zeigt wie die Algorithmen aufgebaut sind, welche zur Berechnung der einzelnen Operationen der $(\min,+)$ Algebra relevant sind. Zusätzlich wird in Kapitel 3.4 ein Überblick zu der Implementierung der entstandenen Pythonbibliothek gegeben.

Anschließend wird in Kapitel 4 getestet, welche Auswirkungen verschiedene Parameter auf die Performanz der Implementierung haben.

Zu guter letzt wird in Kapitel 5 eine Zusammenfassung der Arbeit mit einem Ausblick auf zukünftige Erweiterungen gemacht.

2 Hintergrund und Motivation

Ein wichtiges Anwendungsgebiet der $(\min,+)$ Algebra ist der Network Calculus [LT01], bei dem für sogenannte deterministische Warteschlangensysteme (eng. *deterministic queuing systems*) Vorhersagen für deren obere Laufzeitschranke getroffen werden können. Im Folgenden wird erläutert, welche Aussagen mit Hilfe der $(\min,+)$ Algebra im Network Calculus vorgenommen werden können und welche Implementierungen bereits für die $(\min,+)$ Algebra vorhanden sind.

2.1 Network Calculus

Network Calculus(NC) ist ein Framework zur Analyse von Kommunikationsnetzen, mit dessen Hilfe es möglich ist, Flaschenhälse im Netzwerkfluss von Kommunikationsnetzwerken zu finden [LT01]. Es ist außerdem möglich, mit Hilfe des NC, Aussagen über die Performanz eines Kommunikationsnetzwerkes zu treffen. Um Flaschenhälse und damit auch mögliche obere Schranken für den Netzwerkfluss zu finden, kann ein Computernetzwerk als ein deterministisches Warteschlangensystem (eng. *deterministic queuing system*) dargestellt werden, auf das verschiedene Systemtheorien angewendet werden [BT08]. Basierend auf der $(\min,+)$ Algebra, ist es so möglich, Aussagen über verschiedene Performanzmetriken des Kommunikationsnetzwerkes zu treffen. Der Ankunftsprozess von Daten durch Ankunftscurven (engl. *arival curves*) oder der Durchsatz von deterministischen Warteschlangen durch Servicecurven (engl. *service curves*), lässt sich so mit Hilfe von kumulativen nicht fallenden Funktionen R darstellen [LT01], welche definiert sind als

$$R = \{\forall t > 0 : \exists \epsilon > 0 : 0 \leq R(t) \leq R(t + \epsilon), R(t) = 0 \text{ sonst}\}. \quad (2.1)$$

Für die Bestimmung verschiedener Performanzmetriken des Network Calculus stehen folgende Operationen der $(\min,+)$ Algebra zur Verfügung [LT01]

1. *Addition*: $\forall t \in X : (f + g)(t) = f(t) + g(t)$.
2. *Subtraktion*: $\forall t \in X : (f - g)(t) = f(t) - g(t)$.
3. *Minimum*: $\forall t \in X : \min(f, g)(t) = \min(f(t), g(t))$.
4. *Maximum*: $\forall t \in X : \max(f, g)(t) = \max(f(t), g(t))$.
5. *Faltung*: $\forall t \in X : (f * g)(t) = \inf_{0 \leq s \leq t} \{f(t - s) + g(s)\}$.
6. *Entfaltung*: $\forall t \in X : (f \oslash g)(t) = \sup_{s \geq 0} \{f(t + s) - g(s)\}$.

Führt man beispielsweise die Faltung und Entfaltung auf Servicecurven und Ankunftscurven aus, so ist es möglich, den Gesamtfluss mehrerer Systeme miteinander zu konkatenieren und so ein realistisches Systemmodell zu entwerfen [LT01].

Weiterhin ist es möglich, mit Hilfe der Entfaltung das Backlog und das Virtual Delay eines Kommunikationsnetzwerkes zu bestimmen.

Definition 2.1.1

Backlog[Fid10]:

$$\text{Backlog} \leq \alpha \otimes \beta(0)$$

Definition 2.1.2

Virtual Delay[Fid10]:

$$\text{Virtual Delay} \leq \inf[w \geq 0 : \alpha \otimes \beta(-w) \leq 0]$$

Mit Hilfe des Backlogs kann der maximale Puffer für Daten eines Systems bestimmt werden[Fid10]. Das Virtual Delay ist die maximale horizontale Distanz, die zwischen den Ankunftscurven und den Servicecurven bestehen kann.

Angewendet werden können diese Performanzmetriken auf jedes Kommunikationssystem. Vor allem können für Switches in einem System durch die Addition bestimmt werden, wie viel Traffic an allen Eingängen auf einmal ankommen kann und dementsprechend wie viel Puffer ein solcher Switch maximal benötigt.

Eine Besonderheit in Kommunikationsnetzwerken stellen zeitgesteuerte Tore (*engl. time triggered gates*) dar, welche den Service einer Servicekurve β nur zu bestimmten Zeitpunkten an beispielsweise einen Server weiterleiten. Das zeitgesteuerte Tor wiederum wird zu fest vorgegebenen Zeitpunkten geöffnet und geschlossen. Diese Öffnungs- und Schließzeiten des zeitgesteuerten Tors können dabei dynamisch gesteuert werden, oder sich statisch wiederholen. Wiederholen sich die Öffnungs- und Schließzeiten, so entsteht eine sich periodisch wiederholende Servicekurve. Ein Beispiel für ein Systemelement welches solche zeitgesteuerten Servicecurven erzeugt, ist ein 802.1Q-compliant Switch [12] welcher als Erweiterung die Möglichkeit hat, den Netzwerkverkehr zeitgesteuert weiter zu leiten[16]. Um die gezeigten Operationen der $(\min,+)$ Algebra und die periodisch wiederholenden Servicecurven zu unterstützen, werden in dieser Arbeit stückweise lineare Funktionen verwendet.

2.2 Network Calculus Bibliotheken

Es gibt bereits mehrere Implementierungen mit denen es möglich ist, die Operationen in der $(\min,+)$ Algebra zu berechnen. Bekannte Bibliotheken sind etwa DiscoDNC [BS14], CyNC [SSH07], die Real Time Calculus Toolbox [WT06], oder die COINC Bibliothek[BCG+09].

DiscoDNC [BS14] ist eine in Java geschriebene Opensource Bibliothek für Network Calculus. DiscoDNC unterstützt dabei drei häufige Analysearten im Network Calculus [BS14]: *Total Flow Analysis*, *Separated Flow Analysis* und *Pay Multiplexing Only Once Analysis*. Auf der technischen Seite bietet die Bibliothek Klassen für stückweise lineare Kurven, Netzwerkkonfigurationen, Operationen der $(\min,+)$ Algebra, einfache Network Calculus Operationen und Network Calculus Analysen. Vorteil der DiscoDNC Bibliothek ist, dass sie Opensource ist, ein breites Spektrum an Möglichkeiten für Network Calculus bietet und außer dem benötigten *Oracle Java SE Development Kit 8* keine weiteren Abhängigkeiten hat und als *stand-alone tool* verwendet werden kann. Ein Nachteil ist, dass die stückweisen linearen Kurven keine periodische Wiederholung unterstützen.

CyNC (*Cyclic Network Calculus*)[SSH07] ist eine für Matlab entworfene Bibliothek für Network Calculus. Mit CyNC ist es möglich, verteilte eingebettete Systeme zu modellieren und zu analysieren[SSH07]. Die Bibliothek unterteilt sich dabei in verschiedene *modelling blocks*, welche sich in *Generators*, *network elements*, *auxiliary tools* und *display tools* gliedern lassen. Mit Hilfe der *auxiliary tools* ist es möglich, verschiedene Network Calculus Operationen, wie die *Aggregation*, die Faltung verschiedener Servicecurven, das maximale Virtual Delay und das maximale Backlog zu

berechnen. Ein Nachteil der Bibliothek ist, dass sie nur für Matlab erhältlich ist und keine periodisch wiederholenden stückweisen linearen Funktionen unterstützt werden. Auch war die Bibliothek zum Zeitpunkt, zu dem diese Arbeit geschrieben wurde, nicht mehr ohne Weiteres zugänglich.

Die **Real Time Calculus Toolbox**(RTC) Toolbox [WT06] ist eine für Matlab geschriebene Bibliothek für die Analyse von verteilten und eingebetteten Systemen. Die Bibliothek bietet eine, nach eigenen Angaben, effiziente Datenstruktur für stückweise lineare Funktionen und implementiert dazu die meisten $(\min,+)$ und $(\max,+)$ Algebra Operationen[WT06]. Darüber hinaus bietet die Bibliothek die Möglichkeit, Funktionen für die modulate Performanzanalyse für *Real-Time Calculus* berechnen zu lassen. Wie die *CyNC*[SSH07] Bibliothek kann die RTC Toolbox nur mit Matlab verwendet werden und bietet keine Unterstützung für periodisch wiederholende Servicekurven. Darüber hinaus ist die RTC Toolbox zwar frei nutzbar, aber die Bibliothek selbst nicht Opensource, was es schwierig macht sie an eigene Bedürfnisse anzupassen.

Die **COINC** Bibliothek [BCG+09] ist eine in C++ entworfene Bibliothek um verschiedene Operationen der $(\min,+)$ Algebra auf pseudo-periodische lineare stückweise Funktionen berechnen zu können. Dabei werden wichtige Operationen, wie die Addition und Subtraktion, das Minimum und Maximum und die Faltung und Entfaltung unterstützt. Leider war die Bibliothek zum Zeitpunkt dieser Arbeit nicht mehr verfügbar und das Projekt selbst war auch nicht Opensource.

Das Problem mit den meisten dieser Bibliotheken ist, dass sie oft nur eine sehr eingeschränkte Nutzung zulassen und beispielsweise keine Unterstützung für pseudo-periodische Funktionen haben, welche in Kapitel 2.1 vorgestellt wurden. Auch sind die meisten Implementierungen nicht Opensource und können so nicht auf individuelle Projekte ohne Weiteres angepasst werden. Ziel war es daher eine Bibliothek zu schaffen, die die wichtigsten Operationen der $(\min,+)$ Algebra abdeckt, dabei möglichst effizient ist und pseudo-periodische Funktionen unterstützt. Als Programmiersprache wurde dafür Python gewählt, da Python aktuell eine der gefragtesten Programmiersprachen ist [18] und von Vielen im Bereich der Forschung verwendet wird.

3 Stückweise lineare Funktionen

Um die in Kapitel 2.1 definierten kumulativen nicht fallenden Funktionen des NC darstellen und auf ihnen die Operationen der $(\min,+)$ Algebra berechnen zu können, werden in dieser Arbeit sogenannte stückweise lineare Funktionen verwendet. Für diese stückweisen linearen Funktionen wird in diesem Kapitel zunächst eine rein formale Definition aufgestellt. Neben dieser Definition für stückweise lineare Funktionen, werden zusätzlich einige weitere Definitionen und Lemmas aufgestellt, die im weiteren Verlauf der Arbeit wichtig sind. Mit Hilfe dieser Definition für stückweise lineare Funktionen, wird dann eine Datenstruktur entworfen, die leicht zu implementieren und auf der möglichst effizient Algorithmen ausgeführt werden können. Weiter werden Algorithmen, hauptsächlich basierend auf den Algorithmen aus der Arbeit von Bouillard und Thierry [BT08], vorgestellt, mit denen die in Kapitel 2.1 definierten mathematischen Operationen aus der $(\min,+)$ -Algebra berechnet werden können und die mit der vorgestellten Datenstruktur arbeiten. Anschließend gibt es noch eine kleine Zusammenfassung zu der Implementierung, in der der Aufbau der einzelnen Klassen und weitere Details zur Implementierung erklärt werden.

3.1 Definitionen

Es gibt schon mehrere Ansätze um stückweise lineare Funktionen zu definieren. Viele dieser Definitionen, wie die von Mohrenschildt [MOH98], sind sehr komplex, was ihre Auswertung für einen bestimmten Wert betrifft und können nicht so effizient berechnet werden. Mohrenschildts Piecewise Polynomial Function Ring (**PPFR**) [MOH98] beispielsweise operiert zum Auswerten der stückweisen linearen Funktionen über einen mathematischen Ring, welcher die Auswertung weniger effizient macht und zusätzlich mächtiger und damit auch komplexer ist, als die $(\min,+)$ Algebra mit ihrem Halbring.

Aus diesem Grund hat Carette [Car07] versucht, eine bessere Definition für stückweise lineare Funktionen zu finden, die es ermöglicht, einen großen Definitionsbereich von Funktionen abzudecken und gleichzeitig eine einfache und saubere Struktur für effiziente Algorithmen bietet. Für seinen Ansatz definiert Carette zunächst einen stückweisen Ausdruck, eine Domänenpartition und eine linear geordnete Menge.

Definition 3.1.1

Stückweiser Ausdruck [Car07]: Ein stückweiser Ausdruck ist eine Funktion f die von einer Domänenpartition \mathcal{R} auf eine Menge S abbildet. $f : \mathcal{R} \rightarrow S$

Definition 3.1.2

Domänenpartition [Car07]: Eine Domänenpartition \mathcal{R} einer linear geordneten Menge Λ ist eine endliche Menge \mathcal{B} von Punkten $\lambda_1 < \lambda_2 < \dots < \lambda_n$,

die in einer natürlichen Aufteilung von Λ in disjunkte Mengen $\Lambda_1, \dots, \Lambda_{n+1}$ liegen, wobei gilt

$$\Lambda_1 := \{x \in \Lambda \mid x < \lambda_1\}$$

$$\Lambda_i := \{x \in \Lambda \mid \lambda_{i-1} < x < \lambda_i\}, i = 2, \dots, n$$

$$\Lambda_{n+1} := \{x \in \Lambda \mid \lambda_n < x\}.$$

Definition 3.1.3

Linear geordnete Menge [Car07]: Eine Menge Λ ist **linear geordnet**, wenn $\forall a, b \in \Lambda, a \neq b$ eine Relation $>$ existiert, sodass gilt $a > b$ oder $b > a$.

Mit Hilfe von \mathcal{B} können also die einzelnen Bereiche der linearen Abschnitte einer stückweisen linearen Funktion von einander getrennt werden. Dabei symbolisiert jeder Punkt von \mathcal{B} den Beginn und gleichzeitig auch das Ende eines linearen Abschnittes, welcher einer der disjunkten Mengen $\Lambda_1, \dots, \Lambda_{n+1}$ ist. Als ein Beispiel für einen stückweisen Ausdruck sei $\Lambda = \mathbb{R}$, $\mathcal{B} = \{3\}$ und $S = \{x, x^4\}$ dann ist $f : \mathcal{R} \rightarrow S$ stückweiser Ausdruck mit

$$f(y) = \begin{cases} x & y = \Lambda_1 \\ x^4 & y = 3 \\ x & y = \Lambda_2 \end{cases} \quad (3.1)$$

Mit Hilfe dieses stückweisen Ausdrucks definiert Carette einen stückweisen Operator, mit dem er letztendlich stückweise lineare Funktionen definiert.

Definition 3.1.4

Stückweiser Operator [Car07]: Sei S eine Menge von Funktionen. Ein stückweiser Operator ist dann ein stückweiser Ausdruck $f : \Lambda \rightarrow S$.

Definition 3.1.5

Stückweise lineare Funktion nach Carette [Car07]: Eine stückweise lineare Funktion ist eine Funktion $F : \Lambda \rightarrow V$ mit $F(\Lambda) = f(\mathcal{X}(\Lambda))(\Lambda) = f_{\mathcal{B}}(\Lambda)(\Lambda)$, wobei f ein stückweiser Operator $f : \mathcal{R} \rightarrow S$ ist, bei dem S eine Menge von Funktionen $s : \Lambda \rightarrow V$ ist.

Vorteil dieser Definition für stückweise lineare Funktionen ist, dass im Vergleich zu Mohrenschildts Definition von stückweisen linearen Funktionen kein mathematischer Ring zur Auswertung der Funktion benötigt wird. Außerdem lassen sich die Abbildungen von $\Lambda \rightarrow V$ sehr gut als lineare Stücke (siehe Kapitel 3.2) realisieren, welche in der Datenstruktur Anwendung finden. Da die in Kapitel 3.3 vorgestellten Algorithmen nur einen eingeschränkten Bereich dieser stückweisen linearen Funktionen von Carette unterstützen, erhält das Set S auf das die domänen Partition abbildet, die Einschränkung, dass es nur Funktionen der Form $f(x) = a * x + n$ mit $a, x, n \in \mathbb{R}$ enthalten darf. Es ist außerdem noch anzumerken, dass diese Definition von linearen stückweisen Funktionen, im Gegensatz zu den in Kapitel 2.1 definierten kumulativen nicht fallenden Funktionen, auch abnehmende Funktionen unterstützen.

3.1.1 Definitionen und Lemmas zur Faltung

Um die Faltung und Entfaltung algorithmisch durchzuführen, benötigen wir einige Lemmas und Definitionen

Definition 3.1.6

Spots und Segmente[BT08]: Für die Menge aller stückweise linearen Funktionen F gilt

- Ein Spot ist eine Funktion $f \in F$ in der für ein $a \in \mathbb{R}_+$ gilt, dass $\forall t \in \mathbb{R}_+ \setminus \{a\}, f(t) = +\infty$ und $f(a) \neq +\infty$.
- Ein Segment ist eine Funktion $f \in F$ mit $a, b \in \mathbb{R}_+, a < b$ im Intervall $]a, b[$ gilt, dass $\exists p, n \in \mathbb{R}_+ : f(t) = p * (t - a) + n$, wenn $t \in]a, b[$ und sonst $f(t) = +\infty$. $]a, b[$ ist dabei der Support von f und p ist die Steigung von f .

Mit Hilfe dieser Definitionen können wir folgende Aussagen zu der Faltung mit Spots und Segmenten machen:

Lemma 3.1.1

Faltung zweier Spots[BT08]:

Seien f_1 und f_2 zwei Spots in a und b . Es gilt nun $f_1 * f_2$ ist ein Spot in $a + b$ und $f_1 * f_2(a + b) = f_1(a) + f_2(b)$.

Lemma 3.1.2

Faltung eines Segments und eines Spots[BT08]:

Sei f_1 ein Segment im Intervall $]a, b[$ und f_2 ein Spot in c . Es gilt nun $f_1 * f_2$ ist ein Segment in $]a + c, b + c[$ und $\forall t \in]a, b[, f_1 * f_2(t + c) = f_1(t) + f_2(c)$.

Lemma 3.1.3

Faltung zweier Segmente [LT01]:

Sei f_1 ein Segment im Intervall $]a, b[$ mit Steigung p_1 und f_2 ein Segment im Intervall $]c, d[$ mit Steigung p_2 und $p_1 \leq p_2$. Es gilt nun $f_1 * f_2$ ist ein Segment in $]a + c, b + d[$ und $\forall t \in]a + c, b + d[$,

$$f_1 * f_2(t) = \begin{cases} f_1(a+) + f_2(c+) + p_1(t - a - c) & \text{für } t \leq b + c \\ f_1(a+) + f_2(c+) + p_1(b - a) + p_2(t - b - c) & \text{für } t > b + c. \end{cases}$$

Graphisch gesehen bedeutet das, dass die Segmente, geordnet nach zunehmender Steigung, mit einander konkateniert werden (siehe Abbildung 3.1).

Die Beweise zu den Lemmas 3.1.1 und 3.1.2 befinden sich im Anhang. Der Beweis zu Lemma 3.1.3 kann im Artikel von Bouillard und Thierry [BT08] nachgelesen werden.

3.1.2 Lemmas zur Entfaltung

Ähnlich wie bei den Spots und Segmenten für die Faltung ist es möglich Spots und Segmente für die Entfaltung zu definieren, mit dem Unterschied, dass diese außerhalb ihres Supports einen Wert von $-\infty$ statt $+\infty$ haben. Mit dieser Änderung zur Definition für Spots und Segmente können wir zur Entfaltung folgende Aussagen machen, wobei \mathbb{R}_+ die positiven reellen Zahlen sind:

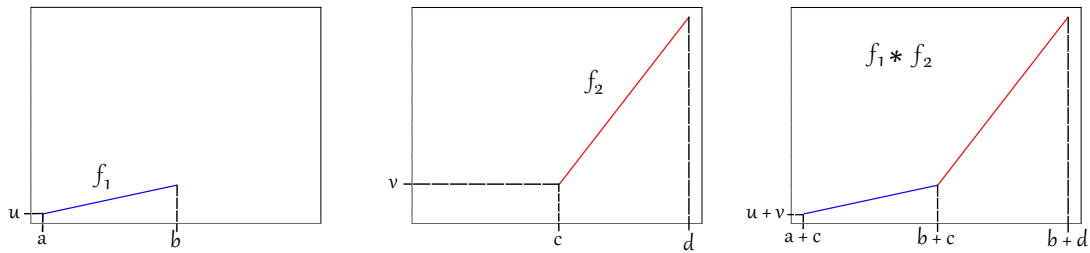


Abbildung 3.1: Faltung von zwei Segmenten f_1 und f_2

Lemma 3.1.4

Entfaltung zweier Spots[BT08]:

Seien f_1 und f_2 zwei Spots in a und b . Es gilt nun $f_1 \otimes f_2$ ist ein Spot in $a - b$ und

$$f_1 \otimes f_2(a - b) = \begin{cases} f_1(a) - f_2(b) & \text{wenn } a \geq b \\ -\infty & \text{sonst} \end{cases}$$

Lemma 3.1.5

Entfaltung eines Segments und eines Spots[BT08]:

Sei f_1 ein Segment im Intervall $]a, b[$ mit Steigung p und f_2 ein Spot in c . Es gilt nun $f_1 \otimes f_2$ ist ein Segment in $]a - c, b - c[\cap \mathbb{R}_+$ und $\forall t \in]a - c, b - c[\cap \mathbb{R}_+, f_1 \otimes f_2(t) = p * (t + c - a) + f_1(a+) - f_2(c)$.

Lemma 3.1.6

Entfaltung eines Spots und eines Segments[BT08]:

Sei f_1 ein Spot in a und f_2 ein Segment in $]b, c[$ mit Steigung p . Es gilt nun $f_1 \otimes f_2$ ist ein Segment in $]a - c, a - b[\cap \mathbb{R}_+$ und $\forall t \in]a - c, a - b[\cap \mathbb{R}_+, f_1 \otimes f_2(t) = p * (t + b - a) + f_1(a) - f_2(b+)$.

Lemma 3.1.7

Entfaltung zweier Segmente[BT08]:

Sei f_1 ein Segment im Intervall $]a, b[$ mit Steigung p_1 und f_2 ein Segment im Intervall $]c, d[$ mit Steigung p_2 . Es gilt nun $f_1 \otimes f_2$ ist ein Segment in $]a - d, b - c[\cap \mathbb{R}_+$ und $\forall t \in]a - d, b - c[\cap \mathbb{R}_+,$ wenn $p_1 \geq p_2$

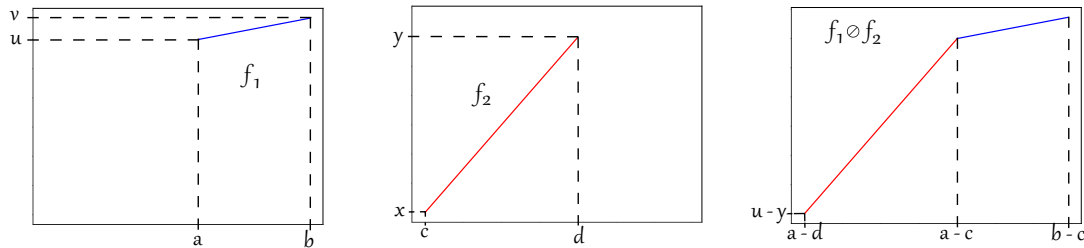
$$f_1 \otimes f_2(t) = \begin{cases} f_1(t + d) - f_2(d-) & \text{für } a - d < t \leq b - d \\ f_1(b-) - f_2(b - t) & \text{für } b - d \leq t < b - c \\ -\infty & \text{sonst} \end{cases}$$

und wenn $p_1 \leq p_2$

$$f_1 \otimes f_2(t) = \begin{cases} f_1(a+) - f_2(a - t) & \text{für } a - d < t \leq a - c \\ f_1(t + c) - f_2(c+) & \text{für } a - c \leq t < b - c \\ -\infty & \text{sonst.} \end{cases}$$

Graphisch gesehen bedeutet das, dass die Segmente, geordnet nach abnehmender Steigung, mit einander konkateniert werden (siehe Abbildung 3.2).

Der Beweis zu Lemma 3.1.4 befindet sich im Anhang. Die Beweise zu den Lemmas 3.1.5 - 3.1.7 können im Artikel von Bouillard und Thierry [BT08] nachgelesen werden.

Abbildung 3.2: Entfaltung von zwei Segmenten f_1 und f_2

3.2 Datenstruktur

Um die Datenstruktur für eine stückweise lineare Funktion erstellen zu können, benötigen wir zunächst lineare Teilstücke. Ein lineares Teilstück kann mit folgendem Konstruktor erstellt werden: *LinearPiece*($x1, y, y_plus, l, slope$). Das lineare Teilstück repräsentiert mit diesen Parametern dabei folgende Funktion

$$\forall t \in [x1, x1 + l] [f(t) = \begin{cases} y + t * p & \text{für } t > x1 \\ y_plus & \text{sonst} \end{cases} \quad (3.2)$$

$x1$ ist also die erste Position auf der x-Achse, für die das lineare Teilstück definiert ist, l die Länge des linearen Teilstücks, y der Wert auf der y-Achse, mit dem das lineare Teilstück für x-Werte $> x1$ beginnt, p die Steigung des linearen Teilstücks und y_plus der y-Wert, den das lineare Teilstück zum Zeitpunkt $x1$ hat. Ein lineares Teilstück stellt dabei einen Teil der Abbildung von $\Lambda \rightarrow V$ aus unserer Definition für stückweise lineare Funktionen aus Kapitel 3.1 dar.

Für den weiteren Verlauf sei die folgende Form $lp = [x1, y, y_plus, l, p]$ eine Kurzschreibweise für ein lineares Stück. Abbildung 3.3 illustriert nochmals die genannten Parameter mit dem Aufbau eines linearen Teilstücks.

Eine stückweise lineare Funktion (*LinearFunction*) kann nun mit Hilfe von linearen Teilstücken (*LinearPiece*) erstellt werden und wird nach außen hin mit folgendem Konstruktor erstellt: *LinearFunction*(*repeat_last_n*, **lin_pieces*). **lin_pieces* kann dabei entweder eine Liste von linearen Teilstücken oder eine variable Anzahl von linearen Teilstücken sein. Mit Hilfe des Parameters *repeat_last_n* ist es möglich, zu bestimmen wie viele der letzten n linearen Teilstücke von **lin_pieces* sich periodisch wiederholen sollen. Es ist dabei nicht zulässig, dass sich die Intervalle der einzelnen linearen Teilstücke überschneiden und eine stückweise lineare Funktion muss im gesamten Bereich vom ersten bis zum letzten linearen Teilstück definiert sein. Weiter wird definiert, dass eine lineare Funktion ihre Steigung zur linken Seite der x-Achse hält.

Die Datenstruktur selbst unterteilt die linearen Stücke in eine Liste mit den nicht periodischen linearen Teilstücken und eine Liste mit den periodischen linearen Teilstücken. Zusätzlich wird der Beginn des periodischen Teils auf der x-Achse T sowie die Länge des periodischen Teils d und das Inkrement einer periodischen Wiederholung c gespeichert. Die Wiederholung im periodischen Teil wird nun so realisiert, dass das Inkrement der periodischen Wiederholung c immer nach einem Durchlauf des periodischen Teils addiert wird, sodass für jedes lineare Teilstück im periodischen Teil gilt

$$\forall t \in [x1, x1 + l], \forall \tau \in \mathbb{N} : f(t + \tau * d) = slope * t + y + \tau * c. \quad (3.3)$$

Für den weiteren Verlauf sei die folgende Form $f = \{lp1, lp2, \dots, lpn, T, d, c\}$ eine Kurzschreibweise für stückweise lineare Funktionen, wobei $lp1, lp2, \dots, lpn$ lineare Teilstücke sind. Abbildung 3.4 zeigt noch einmal den Aufbau einer linearen Funktion und Abbildung 3.5 eine stückweise lineare Funktion mit eingezeichneten Parametern T, d und c .

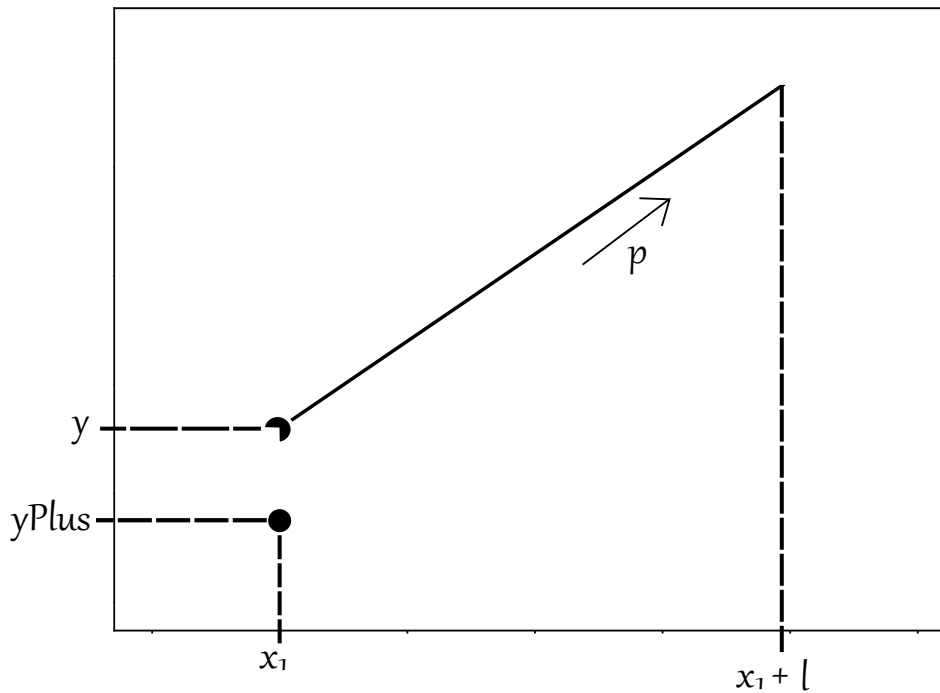


Abbildung 3.3: Ein lineares Teilstück mit seinen Parametern $x_1, y, yPlus, p$ und l .

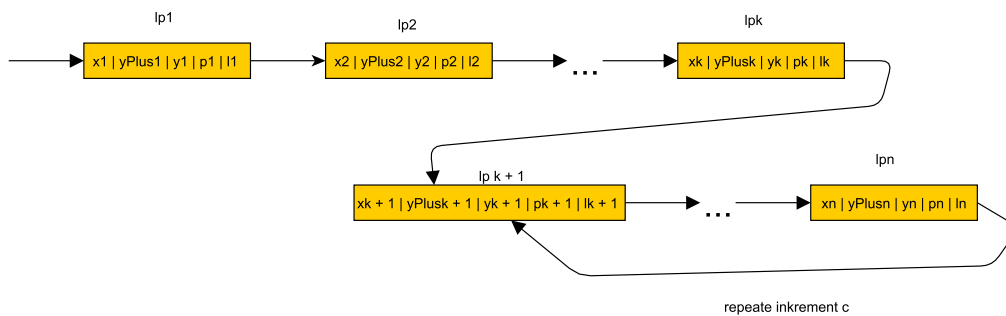


Abbildung 3.4: Aufbau der Datenstruktur zum halten einer linearen Funktion.

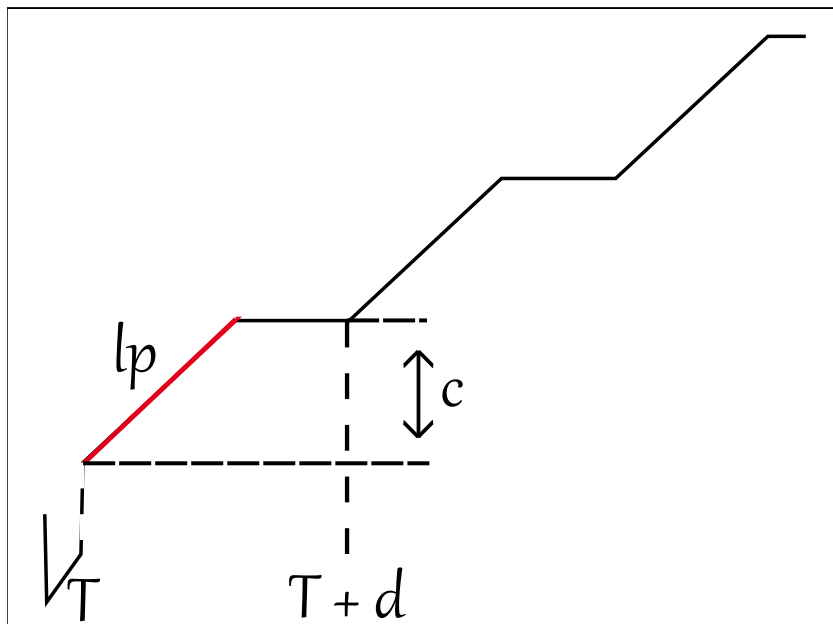


Abbildung 3.5: Eine stückweise lineare Funktion mit ihren Parametern T , d und c .

3.3 Algorithmen

Um die in Kapitel 2.1 definierten mathematischen Operationen der MinPlus-Algebra berechnen zu können, sind Algorithmen zur Bestimmung der Addition, Subtraktion, des Minimums, sowie des Maximums, der Faltung und Entfaltung nötig. Um diese mathematischen Operationen berechnen zu können, wurden Algorithmen aus dem Artikel von Bouillard und Thierry [BT08] verwendet. Die dabei gezeigten Algorithmen, erfordern dafür alle stückweise lineare Funktionen, welche ihren Definitionsbereich nur im positiven Intervall $[0, +\infty[$. Dabei können die stückweisen linearen Funktionen auch abnehmend sein und müssen nicht wie die in Kapitel 2.1 definierten Funktionen kumulativ nicht fallenden sein. Außerdem können die stückweisen linearen Funktionen für die Addition und Subtraktion, sowie für das Minimum und Maximum auch für $t < 0$ bestimmt werden. Zusätzlich werden noch die Berechnung des Backlogs, welches die Entfaltung zum Zeitpunkt 0 ist und die Berechnung des Virtual Delay unterstützt.

3.3.1 Addition und Subtraktion

Um die Addition oder die Subtraktion von zwei stückweisen linearen Funktionen $f_1 + f_2 = f'$ (resp. $f_1 - f_2 = f''$) bestimmen zu können, muss zunächst für die resultierende Funktion f' der Beginn des periodischen Intervalls $T = \max(T_1, T_2)$ und die Länge des periodischen Intervalls $d = \text{kgV}(d_1, d_2)$ bestimmt werden. Anschließend werden f_1 und f_2 so verlängert, dass sie diese neuen periodischen Eigenschaften T und d aufweisen. Danach lässt sich die Addition oder Subtraktion bestimmen, indem im Intervall $I = [0, \max(T_1, T_2) + d[$ für jede Diskontinuität und jedes passende Paar von linearen Teilstücken $f_1 + f_2$ bzw. $f_1 - f_2$ berechnet werden. Algorithmus 3.1 fasst die einzelnen Schritte und Berechnungen für die Addition noch einmal zusammen.

Algorithmus 3.1 Addition von zwei stückweisen linearen Funktionen [BT08]

Input $f_1, f_2 \in R$

Output: $f_1 + f_2$ mit Parametern T, d, c

Funktion ADDITION(f_1, f_2)

$T \leftarrow \max(T_1, T_2)$

$d \leftarrow \text{kgV}(d_1, d_2)$

Extend(f_1, T, d);

Extend(f_2, T, d);

Merge die linearen Teilstücke von f_1 und f_2

Berechne $f_1 + f_2$ für jede Diskontinuität und für jedes passende Paar von linearen Stücken aus f_1 und f_2

end Funktion

Abbildung 3.6-3.9 zeigen, wie sich die Addition aus zwei Funktionen f_1 und f_2 ergibt.

Dabei ist $f_1 = \{[0, 3, 0, 10, 1], [10, 13, 13, 2, 3], [12, 19, 19, 2, 0], 10, 4, 6\}$ und

$f_2 = \{[0, 0, 0, 5, 0], [5, 4, 0, 4, 2], [9, 12, 12, 6, 1], 9, 6, 6\}$.

Als neuen Beginn des periodischen Teils wird $T = \max(T_1, T_2) = \max(10, 9) = 10$ und als neue Länge des periodischen Intervalls $d = \text{kgV}(d_1, d_2) = \text{kgV}(4, 6) = 12$ gewählt. Die Funktionen f_1 und f_2 müssen also auf T und d verlängert (Abbildung 3.8) und anschließend deren lineare Teilstücke aufaddiert werden. Das Resultat ist in Abbildung 3.9 in grün eingezeichnet.

Proposition 3.3.1

[BT08] Seien f_1 und f_2 stückweise lineare Funktionen. Die Addition von f_1 und f_2 lässt sich mit Algorithmus 3.1 in $O(N_1 + N_2)$ berechnen, wobei für die Anzahl nichtperiodischer linearer Teilstücke n_x und periodischer linearer Teilstücke p_x gilt: $N_1 = n_1 + p_1 \frac{(T+d)-T_1}{d_1}$ und $N_2 = n_2 + p_2 \frac{(T+d)-T_2}{d_2}$ und $N_{\min(f_1, f_2)} \leq 2(N_1 + N_2)$.

Der Beweis zu dieser Proposition befindet sich im Anhang.

Die Komplexität der Subtraktion liegt wie die Addition in $O(N_1 + N_2)$ und kann analog zur Addition bewiesen werden.

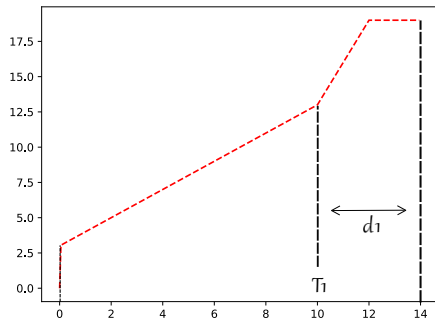


Abbildung 3.6: Die stückweise lineare Funktion f_1 (rot) mit ihren Parametern T_1 und d_1

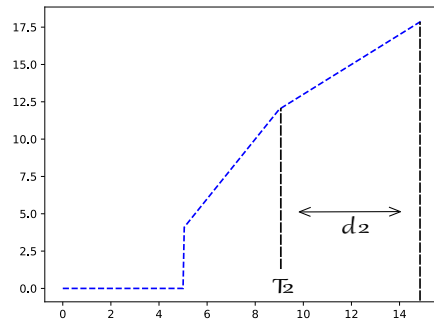


Abbildung 3.7: Die stückweise lineare Funktion f_2 (blau) mit ihren Parametern T_2 und d_2

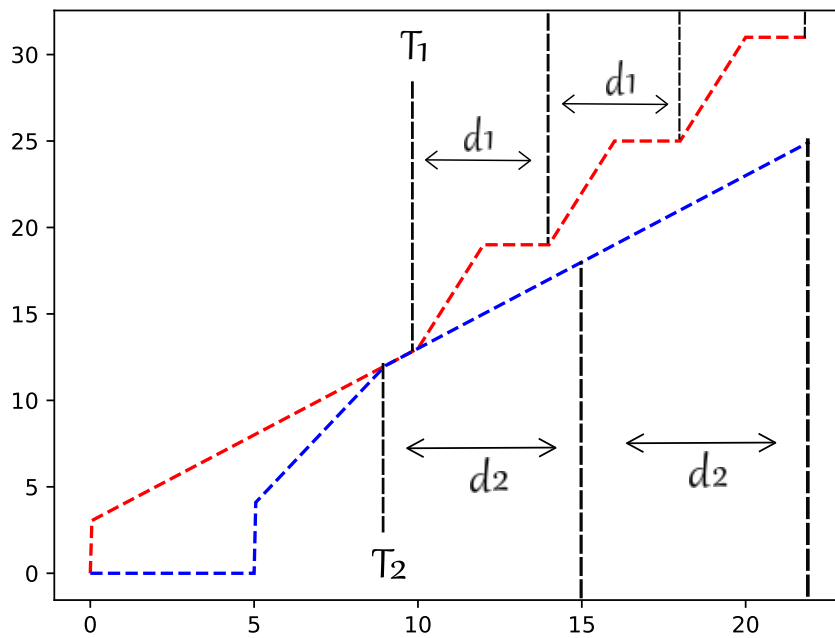


Abbildung 3.8: f_1 und f_2 verlängert.

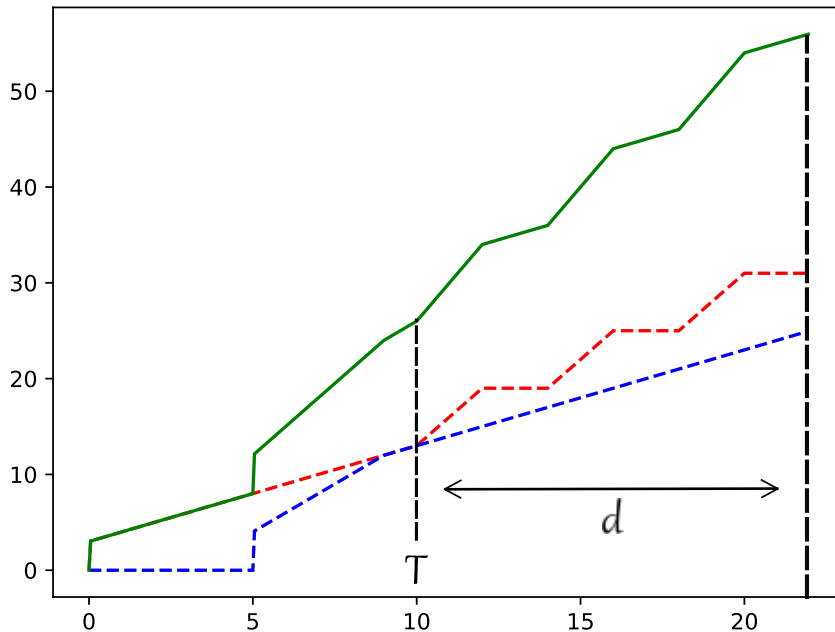


Abbildung 3.9: Ergebnis der Addition von f_1 und f_2 in grün mit neuen Parametern T und d

3.3.2 Minimum und Maximum

Das Minimum und Maximum von zwei stückweisen linearen Funktionen f_1 und f_2 ergibt sich aus den Minima bzw. Maxima der einzelnen linearen Teilstücke. Das Intervall, in dem das Minimum bzw. Maximum bestimmt werden muss, wird dabei vom Steigungsquotienten $\frac{c_i}{d_i}$ zwischen dem Inkrement des periodischen Teils c_i sowie der Länge des periodischen Intervalls d_i abhängig gemacht. Wie in Algorithmus 3.2 und 3.3 zu sehen ist, wird basierend auf dem Steigungsquotienten zwischen dem Inkrement des periodischen Teils (c_1, c_2), sowie der Länge des periodischen Intervalls (d_1, d_2) der stückweisen linearen Funktionen f_1 und f_2 die neue Länge des periodischen Intervalls d und der neue Beginn des periodischen Teils T gewählt.

Als neue Länge des periodischen Intervalls d wird bei einem unterschiedlichen Steigungsquotienten bei der Berechnung des Minimums immer die Periodenlänge der stückweisen linearen Funktion mit dem kleineren Steigungsquotienten gewählt, da diese stückweise lineare Funktion ab einem Punkt t_i kleiner sein wird als die andere Funktion

$$\exists t_i, \forall t \geq t_i : f_1(t) = t * \frac{c_1}{d_1} + n > f_2(t) = t * \frac{c_2}{d_2} + m \text{ fr } \frac{c_1}{d_1} > \frac{c_2}{d_2}. \quad (3.4)$$

Beim Maximum wird entsprechend die Periodenlänge der stückweisen linearen Funktion mit größerem Steigungsquotienten gewählt. Ist der Steigungsquotient der beiden stückweisen linearen Funktionen gleich, so kann wie bei der Addition und Subtraktion als neue Länge des periodischen Intervalls das kgV von d_1 und d_2 gewählt werden.

Um den Beginn des neuen periodischen Teils zu bestimmen, wird wie bei der Addition und Subtraktion das Maximum von T_1 und T_2 und zusätzlich einem weiteren Wert T_x , falls der Steigungsquotient der beiden Funktionen ungleich ist, gewählt. Da die beiden stückweisen linearen Funktionen sich zu Beginn im periodischen Teil schneiden können, stellt T_x den Punkt dar, an dem die beiden stückweise linearen Funktionen f_1 und f_2 sich nicht mehr schneiden können. Dafür wird zunächst im periodischen Teil für die stückweise lineare Funktion mit kleinerem Steigungsquotienten die größte Abweichung von der Steigungsquotientengerade m_1 und bei der stückweisen linearen Funktion mit größerem Steigungsquotienten die kleinste Abweichung von der Steigungsquotientengerade m_2 bestimmt. Die Differenz $m_1 - m_2$ gibt den zu erreichenden Wert auf der y-Achse an und muss durch das Wachstum der Differenz der beiden Steigungsquotienten geteilt werden, um den x-Wert bestimmen zu können, an dem es keine Überschneidung der beiden stückweisen linearen Funktionen mehr geben kann.

Wurde die neue Länge der periodischen Intervalls d und der Beginn des periodischen Teils T bestimmt und f_1 und f_2 auf die neuen Parameter d und T verlängert, so kann wie bei der Addition und Subtraktion das Minimum oder Maximum für jede Diskontinuität und jedes passende Paar von linearen Teilstücken aus f_1 und f_2 berechnet werden.

Abbildung 3.10-3.13 zeigen, wie sich das Minimum aus zwei stückweise linearen Funktionen f_1 und f_2 ergibt. Dabei ist $f_1 = \{[0, 0, 0, 5, 1], [5, 5, 5, 3, 3], [8, 14, 14, 2, 1], 5, 5, 11\}$ und $f_2 = \{[0, 1, 1, 5, 1], [5, 6, 6, 1, 9], [6, 15, 15, 4, 0], 5, 5, 9\}$. Da $\frac{c_1}{d_1} > \frac{c_2}{d_2}$ wird als neue Länge des periodischen Intervalls $d = d_2$ gewählt. Um T zu bestimmen werden zunächst

$$m_1 = \sup_{T_2 \leq t < T_2 + d_2} (f_2(t) - \frac{c_2}{d_2}t) = 4.2 \quad (3.5)$$

und

$$m_2 = \inf_{T_1 \leq t < T_1 + d_1} (f_1(t) - \frac{c_1}{d_1}t) = -6 \quad (3.6)$$

bestimmt, um anschließend

$$\frac{m_1 - m_2}{\frac{c_2}{d_2} - \frac{c_1}{d_1}} = \frac{4.2 + 6}{2.2 - 1.8} = 25.5 \quad (3.7)$$

zu berechnen. Aus den errechneten Werten ergibt sich schließlich $T = \max(5, 5, 25.5) = 25.5$. Anschließend werden f_1 und f_2 auf $T = 25.5$ und $d = 5$ verlängert und das Minimum über die einzelnen linearen Teilstücke bestimmt. Das Resultat ist in Abbildung 3.11 in grün eingezeichnet.

Proposition 3.3.2

[BT08] Seien f_1 und f_2 stückweise lineare Funktionen. Das Minimum von f_1 und f_2 lässt sich mit Algorithmus 3.2 in $O(N_1 + N_2)$ berechnen, wobei für die Anzahl nichtperiodischer linearer Teilstücke n_x und periodischer linearer Teilstücke p_x gilt: $N_1 = n_1 + p_1 \frac{(T+d)-T_1}{d_1}$ und $N_2 = n_2 + p_2 \frac{(T+d)-T_2}{d_2}$ und $N_{\min(f_1, f_2)} \leq 2(N_1 + N_2)$.

Der Beweis zu dieser Proposition befindet sich im Anhang.

Die Komplexität für die Berechnung des Maximums liegt wie die Berechnung des Minimums in $O(N_1 + N_2)$ und kann analog zur Berechnung des Minimums bewiesen werden.

3 Stückweise lineare Funktionen

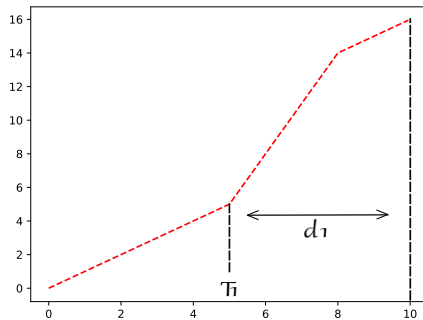


Abbildung 3.10: Die stückweise lineare Funktion f_1 (rot) mit ihren Parametern T_1 und d_1

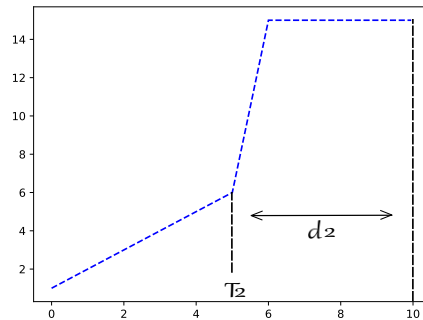


Abbildung 3.11: Die stückweise lineare Funktion f_2 (blau) mit ihren Parametern T_2 und d_2

Algorithmus 3.2 Minimum von zwei stückweisen linearen Funktionen [BT08]

Input $f_1, f_2 \in R$

Output: $\min(f_1, f_2)$ mit Parametern T, d, c

Funktion $\text{MIN}(f_1, f_2)$

```

if  $\frac{c_1}{d_1} < \frac{c_2}{d_2}$  then
     $d \leftarrow d_1$ ;
     $m_1 \leftarrow \sup_{T_1 \leq t < T_1 + d_1} (f_1(t) - \frac{c_1}{d_1}t)$ ;
     $m_2 \leftarrow \inf_{T_2 \leq t < T_2 + d_2} (f_2(t) - \frac{c_2}{d_2}t)$ ;
     $T_x \leftarrow \frac{m_1 - m_2}{\frac{c_2}{d_2} - \frac{c_1}{d_1}}$ ;
     $T \leftarrow \max(T_x, T_1, T_2)$ ;
else if  $\frac{c_1}{d_1} > \frac{c_2}{d_2}$  then
     $d \leftarrow d_2$ ;
     $m_1 \leftarrow \sup_{T_2 \leq t < T_2 + d_2} (f_2(t) - \frac{c_2}{d_2}t)$ ;
     $m_2 \leftarrow \inf_{T_1 \leq t < T_1 + d_1} (f_1(t) - \frac{c_1}{d_1}t)$ ;
     $T_x \leftarrow \frac{m_1 - m_2}{\frac{c_1}{d_1} - \frac{c_2}{d_2}}$ ;
     $T \leftarrow \max(T_x, T_1, T_2)$ ;
else
     $d \leftarrow \text{kgV}(d_1, d_2)$ ;
     $T \leftarrow \max(T_1, T_2)$ ;

```

end if

Extend(f_1, T, d);

Extend(f_2, T, d);

Merge die linearen Teilstücke von f_1 und f_2

Berechne $\min(f, g)$ für jede Diskontinuität und für jedes passende Paar von linearen Stücken aus f_1 und f_2

end Funktion

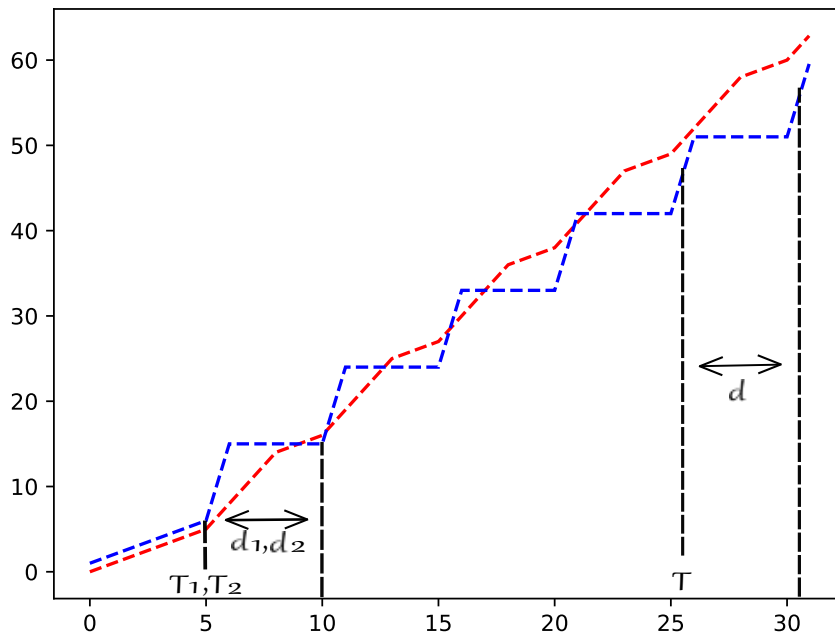


Abbildung 3.12: f_1 und f_2 verlängert.

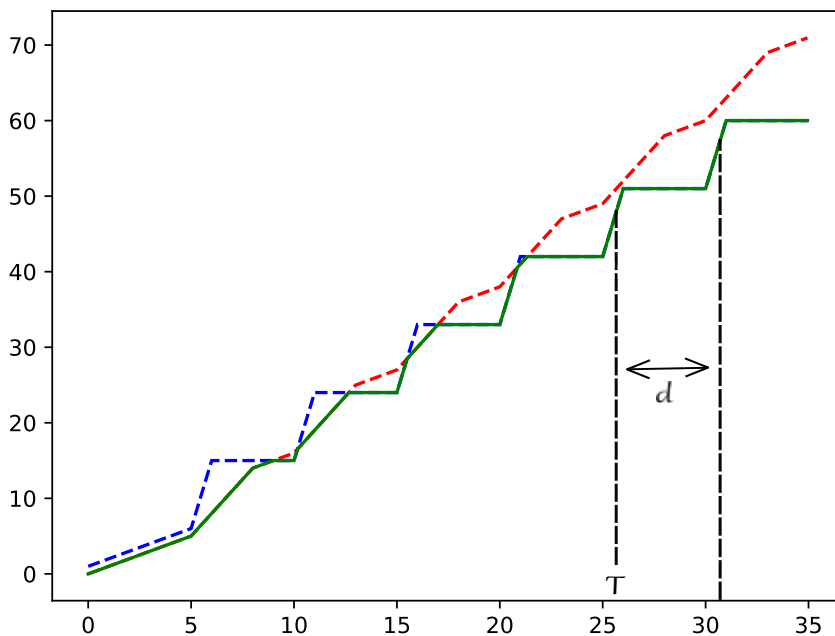


Abbildung 3.13: Das Minimum von f_1 und f_2 in grün mit neuen Parametern T und d

Algorithmus 3.3 Maximum von zwei stückweisen linearen Funktionen

Input $f_1, f_2 \in R$

Output: $\max(f_1, f_2)$ mit Parametern T, d, c

Funktion $\text{MAX}(f_1, f_2)$

```

if  $\frac{c_1}{d_1} > \frac{c_2}{d_2}$  then
     $d \leftarrow d_1$ ;
     $m_1 \leftarrow \sup_{T_1 \leq t < T_1 + d_1} (f_1(t) - \frac{c_1}{d_1}t)$ ;
     $m_2 \leftarrow \inf_{T_2 \leq t < T_2 + d_2} (f_2(t) - \frac{c_2}{d_2}t)$ ;
     $T_x \leftarrow \frac{m_1 - m_2}{\frac{c_2}{d_2} - \frac{c_1}{d_1}}$ ;
     $T \leftarrow \max(T_x, T_1, T_2)$ ;
else if  $\frac{c_1}{d_1} < \frac{c_2}{d_2}$  then
     $d \leftarrow d_2$ ;
     $m_1 \leftarrow \sup_{T_2 \leq t < T_2 + d_2} (f_2(t) - \frac{c_2}{d_2}t)$ ;
     $m_2 \leftarrow \inf_{T_1 \leq t < T_1 + d_1} (f_1(t) - \frac{c_1}{d_1}t)$ ;
     $T_x \leftarrow \frac{m_1 - m_2}{\frac{c_1}{d_1} - \frac{c_2}{d_2}}$ ;
     $T \leftarrow \max(T_x, T_1, T_2)$ ;
else
     $d \leftarrow \text{kgV}(d_1, d_2)$ ;
     $T \leftarrow \max(T_1, T_2)$ ;
end if

```

end if

Extend(f_1, T, d);

Extend(f_2, T, d);

Merge die linearen Teilstücke von f_1 und f_2

Berechne $\max(f, g)$ für jede Diskontinuität und für jedes passende Paar von linearen Stücken aus f_1 und f_2

end Funktion

3.3.3 Faltung

Seien f_1 und f_2 zwei stückweise lineare Funktionen mit Parametern T_1, T_2, d_1 und d_2 . Die Faltung von f_1 und f_2 lässt sich wie folgt bestimmen[BTO8]:

1. Die stückweise lineare Funktion f_1 wird in ihren nichtperiodischen Teil f_1' und in ihren periodischen Teil f_1'' aufgeteilt $f_1 = f_1' \oplus f_1''$. f_1' ist dabei definiert als

$$f_1' = \begin{cases} f_1(t) & \text{für } t \in [0, T_1[\\ +\infty & \text{sonst} \end{cases} \quad (3.8)$$

und f_1'' als

$$f_1'' = \begin{cases} f_1(t) & \text{für } t \in [T_1, +\infty[\\ +\infty & \text{sonst} \end{cases} \quad (3.9)$$

Analog wird $f_2 = f_2' \oplus f_2''$ definiert.

2. Die Faltung ergibt sich nun aus dem Berechnen des Minimums über die einzelnen Faltungen der aufgeteilten stückweisen linearen Funktionen $f_1 * f_2 = (f_1' * f_2') \oplus (f_1' * f_2'') \oplus (f_1'' * f_2') \oplus (f_1'' * f_2'')$.

3. Die Faltung zwischen $f_1' * f_2'$ wird berechnet, indem die einzelnen Segmente und Spots von f_1' und f_2' miteinander gefaltet werden. Seien also $(f_1', i)_{i \in I}$ (analog $(f_2', j)_{j \in J}$) die Menge aller Segmente und Spots von f_1' . Die Faltung ergibt sich dann aus der Faltung der einzelnen Spots und Segmente $(f_1', i)_{i \in I}$ und $(f_2', j)_{j \in J}$ $f_1' * f_2' = \min_{i \in I, j \in J} (f_1', i) * (f_2', j)$. Die Faltung des Spots und Segmente kann dabei mit Hilfe der Lemmas 3.1.1, 3.1.2 und 3.1.3 berechnet werden. Das Ergebnis dieser Faltung ist dann

$$\begin{cases} f_1' * f_2'(t) & \text{für } t \in [0, T_1 + T_2[\\ +\infty & \text{sonst} \end{cases} \quad (3.10)$$

4. Die Faltung von $f_1' * f_2''$ lässt sich so wie im vorherigen Schritt über die einzelnen Segmente und Spots berechnen. Es werden dabei die Segmente und Spots von f_2'' im Intervall $[0, T_1 + T_2 + d_2[$ und bei f_1' im Intervall $[0, T_1[$ miteinander gefaltet. Die resultierende Funktion hat dann die Parameter $T = T_1 + T_2$, $d = d_2$ und $c = c_2$.
5. $f_1'' * f_2'$ lässt sich analog zu Schritt 4 mit den Intervallen $[0, T_1 + T_2 + d_1[$ und $[0, T_2[$ und den neuen Parametern $T = T_1 + T_2$, $d = d_1$ und $c = c_1$ berechnen.
6. Die Faltung von $f_1'' * f_2''$ wird im Intervall von $[0, T_1 + T_2 + 2d[$ mit $d = \text{kgV}(d_1, d_2)$ berechnet und hat die Parameter $T = T_1 + T_2 + d$, $d = \text{kgV}(d_1, d_2)$ und $c = \text{dmin}(\frac{c_1}{d_1}, \frac{c_2}{d_2})$.
7. Das Minimum all dieser Faltungen ergibt dann die Faltung von f_1 und f_2 .

In Abbildung 3.14 - 3.18 wird die Faltung von zwei stückweisen linearen Funktionen f_1 und f_2 mit ihren Zwischenschritten dargestellt.

Dabei ist $f_1 = \{[0, 4, 0, 2, 1], [2, 6, 6, 1, 0], [3, 6, 6, 1, 2], 2, 2, 2\}$ und

$f_2 = \{[0, 0, 0, 3, 2], [3, 6, 6, 2, 0], [5, 6, 6, 1, 3], 3, 3, 3\}$.

Die einzelnen Faltungen der Segmente von $f_1' * f_2'$, $f_1' * f_2''$, $f_1'' * f_2'$ und $f_1'' * f_2''$, sowie deren Minimum sind in Abbildung 3.16 zu sehen.

Abbildung 3.17 zeigt das Minimum dieser Faltungen und in Abbildung 3.18 ist das Ergebnis von $f_1 * f_2$ zusammen mit den beiden stückweisen linearen Funktionen f_1 und f_2 zu sehen.

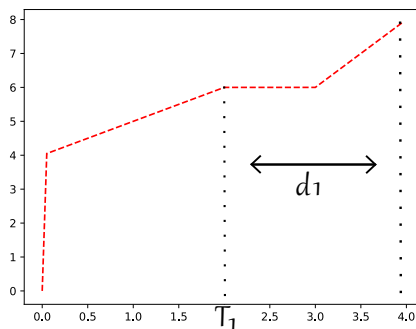


Abbildung 3.14: Die stückweise lineare Funktion f_1 (rot) mit ihren Parametern T_1 und d_1 .

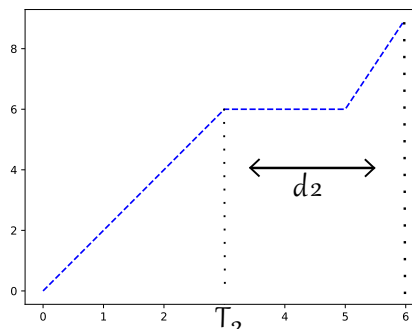


Abbildung 3.15: Die stückweise lineare Funktion f_2 (blau) mit ihren Parametern T_2 und d_2 .

3 Stückweise lineare Funktionen

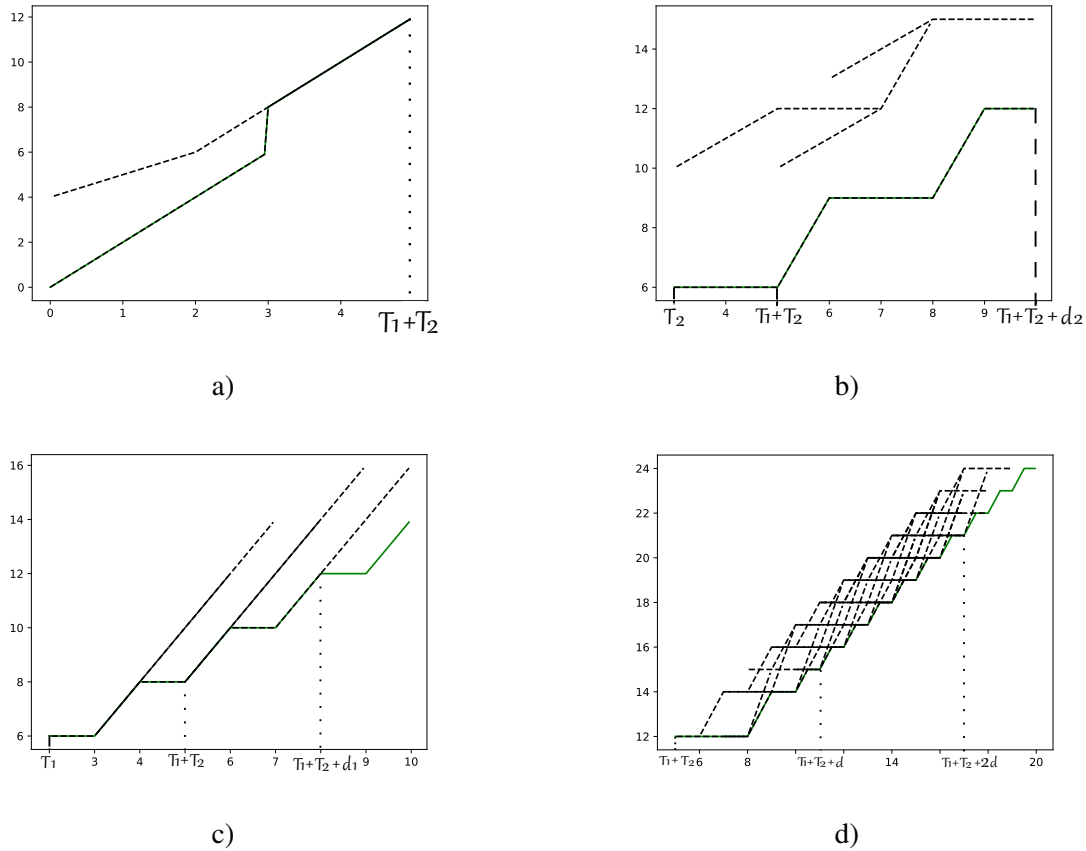


Abbildung 3.16: Die einzelnen Faltungen der Segmente und Spots in schwarz und das Minimum dieser Faltungen in grün. a) ist dabei die Faltungen von $f_1' * f_2'$, b) die Faltungen von $f_1' * f_2''$, c) die Faltungen von $f_1'' * f_2'$ und d) die Faltungen von $f_1'' * f_2''$

Proposition 3.3.3

[BT08] Seien f_1 und f_2 stückweise lineare Funktionen. Die Faltung von f_1 und f_2 lässt sich mit Algorithmus 3.4 in $O(N_1^e N_2^e \log \max(N_1^e, N_2^e) + N_1^e N_2^e \log \max(N_1^e, N_2^e))$ berechnen, wobei für die Anzahl nichtperiodischer linearer Teilstücke n_x und periodischer linearer Teilstücke p_x gilt: $N_1^e = n_1 + p_1 \lceil \frac{(T+d)-T_1}{d_1} \rceil$, $N_2^e = n_2 + p_2 \lceil \frac{(T+d)-T_2}{d_2} \rceil$, $N_1^e = n_1 + p_1 \max(\lceil \frac{(T+d)-T_1}{d_1} \rceil, \frac{2d}{d_1})$ und $N_2^e = n_2 + p_2 \max(\lceil \frac{(T+d)-T_2}{d_2} \rceil, \frac{2d}{d_2})$.

Der Beweis hierzu kann im Paper von Bouillard und Thierry [BT08] nachgelesen werden.

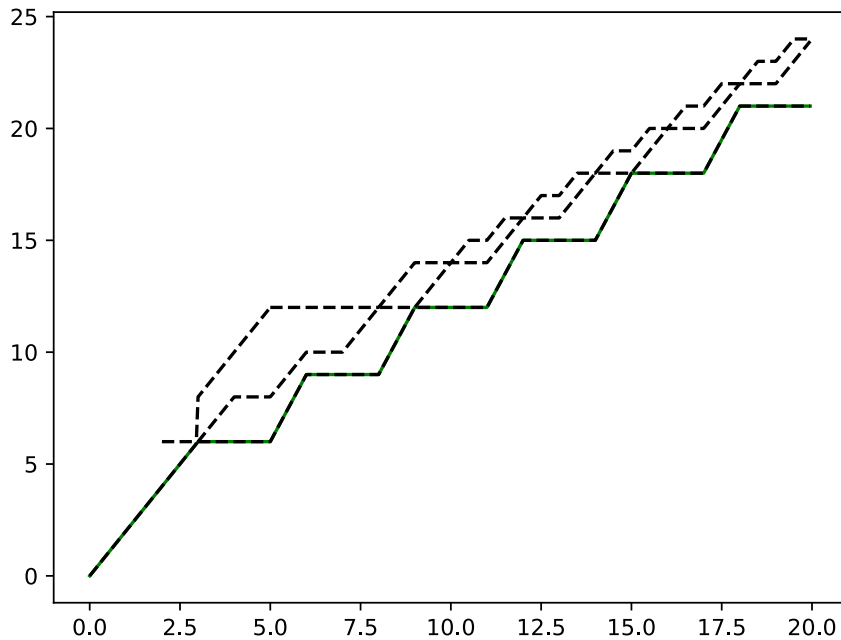


Abbildung 3.17: Die einzelnen Faltungen in schwarz und das Minimum über diese Faltungen in grün.

Algorithmus 3.4 Faltung von zwei stückweisen linearen Funktionen [BT08]

Input $f_1, f_2 \in R$

Output: $f_1 * f_2$ mit Parametern T, d, c

- 1: **Funktion** FALTUNG(f_1, f_2)
 - 2: $d \leftarrow \text{kgV}(d_1, d_2)$;
 - 3: Sei $(f_{j,i'})_{i' \in I_j'}$ die Menge an Spots und Segmenten in f_j' im Intervall $[0, T_j]$
 - 4: Sei $(f_{j,i''})_{i'' \in I_j''}$ die Menge an Spots und Segmenten in f_j'' im Intervall $[T_j, T_1 + T_2 + d_j[$
 - 5: Sei $(f_{j,i'''})_{i''' \in I_j'''}$ die Menge an Spots und Segmenten in f_j'' im Intervall $[T_j, T_j + 2d[$
 - 6: Berechne $\min_{i' \in I_1', j \in I_2'} f_{1,i'} * f_{2,j'} = f_1' * f_2'$ in \mathbb{R}_+
 - 7: Berechne $\min_{i' \in I_1', j \in I_2''} f_{1,i'} * f_{2,j''} = f_1' * f_2''$ im Intervall $[0, T_1 + T_2 + d_2[$, mit neuen
 - 8: Parametern $T = T_1 + T_2$ und $d = d_2$
 - 9: Berechne $\min_{i'' \in I_1'', j \in I_2'''} f_{1,i''} * f_{2,j'''} = f_1'' * f_2'$ im Intervall $[0, T_1 + T_2 + d_1[$, mit neuen
 - 10: Parametern $T = T_1 + T_2$ und $d = d_1$
 - 11: Berechne $\min_{i'' \in I_1''', j \in I_2''''} f_{1,i''} * f_{2,j''''} = f_1'' * f_2''$ im Intervall $[0, T_1 + T_2 + 2d[$, mit neuen
 - 12: Parametern $T = T_1 + T_2 + d$ und $d = d$
 - 13: Berechne $\min((f_1' * f_2'), (f_1' * f_2''), (f_1'' * f_2'), (f_1'' * f_2'')) = f_1 * f_2$
 - 14: **end Funktion**
-

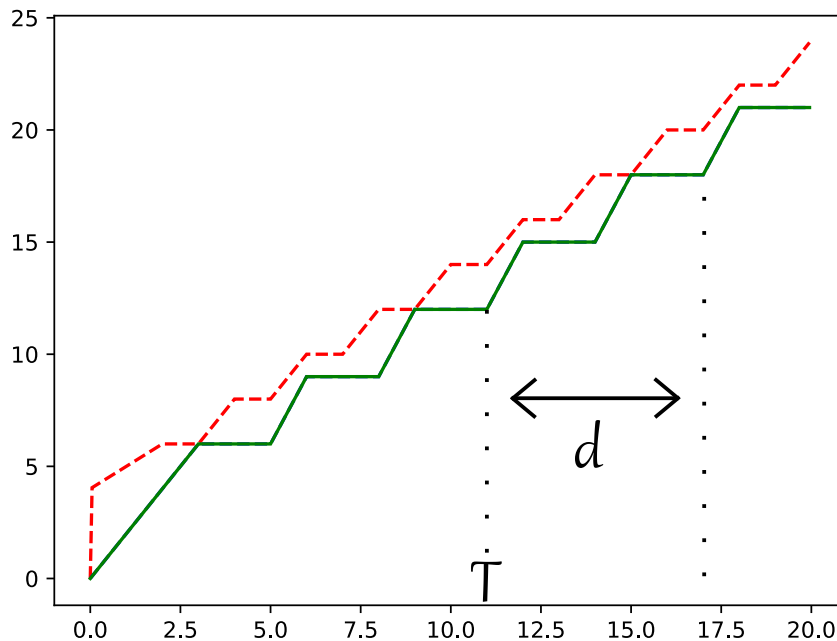


Abbildung 3.18: Die Faltung von f_1 und f_2 in grün mit neuen Parametern T und d .

3.3.4 Entfaltung

Die Berechnung der Entfaltung von zwei stückweisen linearen Funktionen f_1 und f_2 erfolgt ähnlich wie bei der Faltung [BT08]:

Da die Faltung nur Werte $< +\infty$ liefern kann, wenn der Steigungsquotient $\frac{c_1}{d_1}$ von f_1 kleiner oder gleich ist wie der Steigungsquotient $\frac{c_2}{d_2}$ von f_2

$$f \circledast g = \sup_{s \geq 0} (f(t+s) - g(s)) = +\infty, \text{ wenn } \frac{c_1}{d_1} > \frac{c_2}{d_2}, \text{ für } t \rightarrow +\infty \quad (3.11)$$

, wird zunächst geprüft ob $\frac{c_1}{d_1} > \frac{c_2}{d_2}$ und in diesem Fall $f_1 \circledast f_2 = +\infty$ für \mathbb{R}_+ zurück gegeben.

Ist $\frac{c_1}{d_1} \leq \frac{c_2}{d_2}$ wird als nächstes $T = \max(T_1, T_2) + kgV(d_1, d_2)$ bestimmt. Anschließend wird f_1 in seine Spots und Segmente $(f_1, i)_{i \in I}$ im Intervall $0, T + T_1 + d_1[$ aufgeteilt. Danach wird auch f_2 in seine Spots und Segmente $(f_2, j)_{j \in J}$ im Intervall $[0, T[$ aufgeteilt. Das Maximum über die einzelnen Entfaltungen der Spots und Segmente von f_1 und f_2 ergibt dann die Entfaltung von f_1 und f_2 : $\max_{i \in I, j \in J} (f_1, i) \circledast (f_2, j) = f_1 \circledast f_2$, wobei die Faltungen nur im Intervall $[0, T_1 + d_1[$ betrachtet werden und das Ergebnis die neuen Parametern T_1 und d_1 hat. Die Entfaltung der einzelnen Spots und Segmente ergibt sich dabei aus den Lemmas 3.1.4, 3.1.5, 3.1.6 und 3.1.7.

In Algorithmus 3.5 wird die Entfaltung nochmals zusammen gefasst.

In Abbildung 3.19 - 3.22 wird die Entfaltung von zwei stückweisen linearen Funktionen f_1 und f_2 mit ihren Zwischenschritten dargestellt.

Dabei ist $f_1 = \{[0, 0, 0, 2, 3], [2, 6, 6, 3, 2], [5, 12, 12, 1, 0], 2, 4, 6\}$ und

$f_2 = \{[0, 2, 0, 3, 1], [3, 5, 5, 3, 0], [6, 5, 5, 5, 3], 3, 8, 15\}$.

Die einzelnen Entfaltungen der Segmente von $f_1' \otimes f_2'$, sowie deren Maximum ist in Abbildungen 3.21 zu sehen.

In Abbildung 3.22 ist das Ergebnis von $f_1 \otimes f_2$ zusammen mit den beiden stückweisen linearen Funktionen f_1 und f_2 zu sehen.

Algorithmus 3.5 Entfaltung von zwei stückweisen linearen Funktionen [BT08]

Input $f_1, f_2 \in R$

Output: $f_1 \otimes f_2$ mit Parametern T, d, c wenn $\frac{c_1}{d_1} \leq \frac{c_2}{d_2}$ sonst $+\infty$

- 1: **Funktion** ENTFALTUNG(f_1, f_2)
 - 2: **if** $\frac{c_1}{d_1} > \frac{c_2}{d_2}$ **then**
 - 3: $f_1 \otimes f_2 = +\infty$ für \mathbb{R}_+
 - 4: **else**
 - 5: $T \leftarrow \max(T_1, T_2) + \text{kgV}(d_1, d_2)$;
 - 6: Sei $(f_1, i)_{i \in I}$ die Menge an Spots und Segmenten in f_1 im Intervall $[0, T + T_1 + d_1[$
 - 7: Sei $(f_2, j)_{j \in J}$ die Menge an Spots und Segmenten in f_2 im Intervall $[0, T[$
 - 8: Berechne $\max_{i \in I, j \in J} (f_1, i) \otimes (f_2, j) = f_1 \otimes f_2$ im Intervall $[0, T_1 + d_1[$,
 - 9: mit neuen Parametern T_1, d_1, c_1 .
 - 10: **end if**
 - 11: **end Funktion**
-

Proposition 3.3.4

[BT08] Seien f_1 und f_2 stückweise lineare Funktionen. Die Entfaltung von f_1 und f_2 lässt sich mit Algorithmus 3.5 in $O(N_1^e N_2^e \log \max(N_1^e, N_2^e))$ berechnen, wobei für die Anzahl nichtperiodischer linearer Teilstücke n_x und periodischer linearer Teilstücke p_x gilt: $N_1^e = n_1 + p_1 \lceil \frac{(T+d)-T_1}{d_1} \rceil$ und $N_2^e = n_2 + p_2 \lceil \frac{(T+d)-T_2}{d_2} \rceil$.

Der Beweis hierzu kann im Paper von Bouillard und Thierry [BT08] nachgelesen werden.

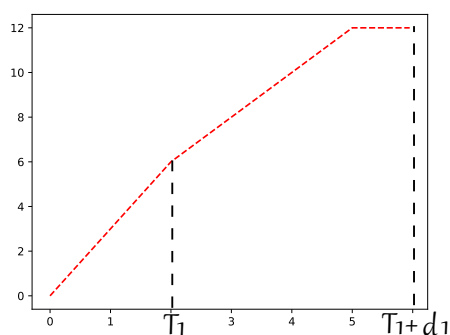


Abbildung 3.19: Die stückweise lineare Funktion f_1 (rot) mit ihren Parametern T_1 und d_1 .

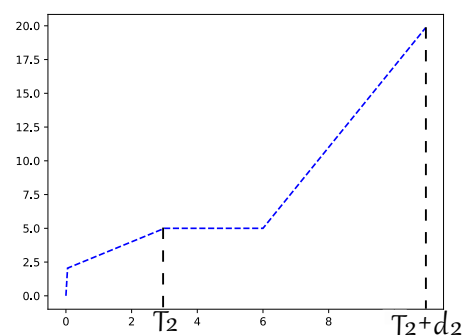


Abbildung 3.20: Die stückweise lineare Funktion f_2 (blau) mit ihren Parametern T_2 und d_2 .

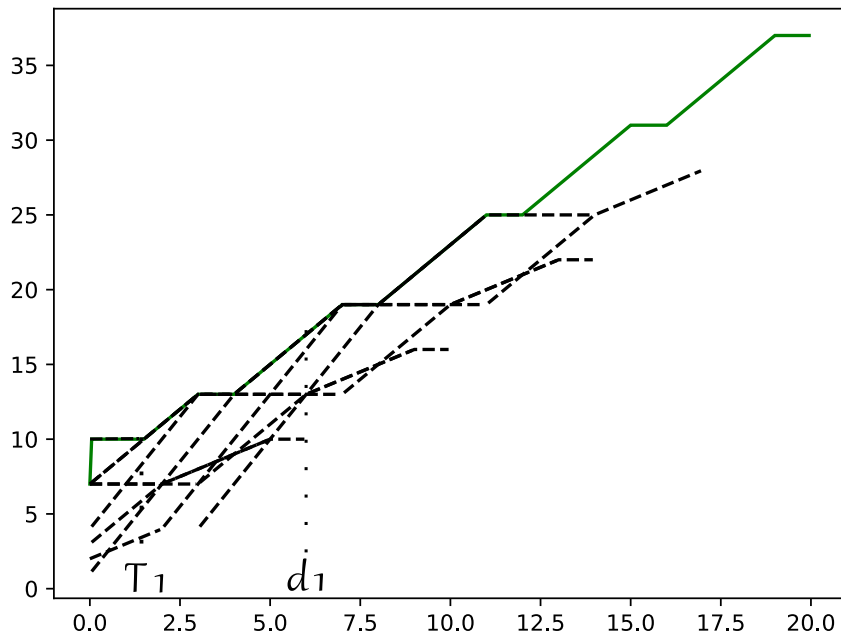


Abbildung 3.21: Die einzelnen Entfaltungen von Segmenten und Spots in schwarz und das Maximum über diese Entfaltungen in grün.

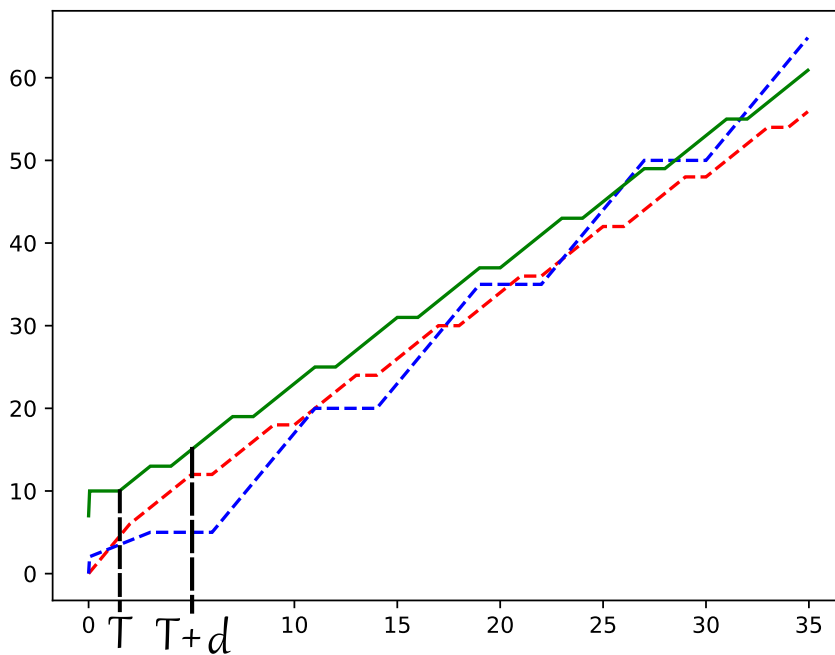


Abbildung 3.22: Die Entfaltung von f_1 und f_2 in grün mit neuen Parametern T und d .

3.3.5 Virtual Delay

Weil die stückweise lineare Funktion, die aus dem Algorithmus zur Entfaltung entsteht, nur im Bereich $[0, +\infty[$ definiert ist, wäre es nicht möglich das Virtual Delay direkt zu berechnen, da wir für das Virtual Delay den kleinsten Wert im negativen Bereich von $\alpha \oslash \beta$ suchen. Als Lösung für dieses Problem müssen wir also zunächst die Definition der Virtual Delays ein wenig umformen:

$$\text{virtual delay} \leq \inf[w \geq 0 : \alpha \oslash \beta(-w) \leq 0] \quad (3.12)$$

$$= \inf[w \geq 0 : \sup_{s \geq 0} \{\alpha(s - w) - \beta(s)\} \leq 0], \quad (3.13)$$

$$= \inf[w \geq 0 : \sup_{s \geq 0} \{\alpha(s) - \beta(s + w)\} \leq 0], \quad (3.14)$$

$$= \inf[w \geq 0 : \sup_{s \geq 0} \{-\beta(s + w) + \alpha(s)\} \leq 0], \quad (3.15)$$

$$= \inf[w \geq 0 : \sup_{s \geq 0} \{-\beta(s + w) + \alpha(s)\} \leq 0], \quad (3.16)$$

$$= \inf[w \geq 0 : \sup_{s \geq 0} \{-\beta(s + w) - (-\alpha(s))\} \leq 0], \quad (3.17)$$

$$= \inf[w \geq 0 : -\beta \oslash -\alpha(w) \leq 0]. \quad (3.18)$$

Mit Hilfe dieser neuen Ungleichung können wir das Virtual Delay wie folgt berechnen. Zuerst müssen die beiden stückweisen linearen Funktionen α und β negiert und die Entfaltung der beiden negierten stückweise linearen Funktionen $-\beta \oslash -\alpha$ berechnet werden. Anschließend bestimmen wir das kleinste w , für das $-\beta \oslash -\alpha \leq 0$ ist.

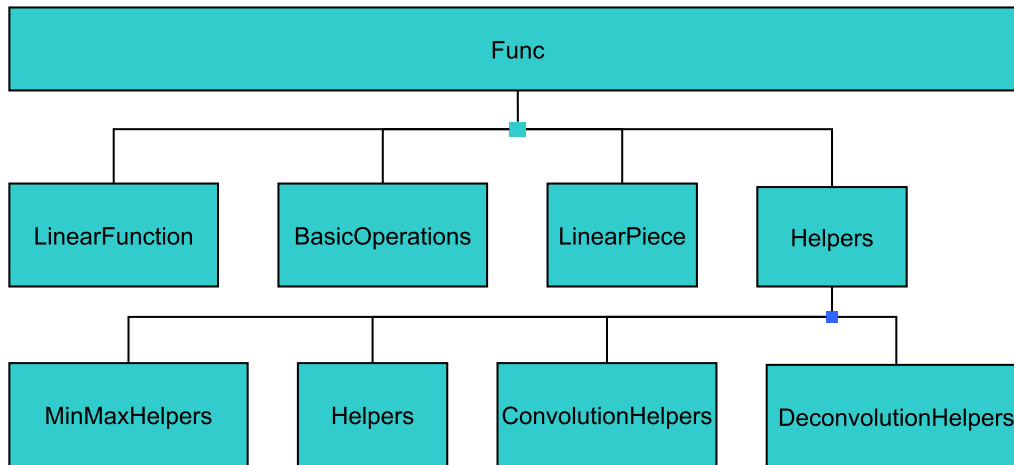


Abbildung 3.23: Aufbau der Implementierung mit den wichtigsten Elementen.

3.4 Implementierung

Der Aufbau der Implementierung ist in Abbildung 3.23 zu sehen.

Der wichtigste Teil befindet sich in den drei Dateien *LinearFunction.py*, *LinearPiece.py* und *BasicOperations.py*. *LinearPiece.py* enthält dabei die Datenstruktur für lineare Teilstücke, welche durch die Klasse *LinearPiece* realisiert wird. In *LinearFunction.py* ist die Datenstruktur für die stückweisen linearen Funktionen enthalten, welche durch die Klasse *LinearFunction* realisiert wird. *BasicOperations.py* enthält die Operationen der $(\min,+)$ Algebra, welche sich in verschiedene Klassen aufteilen: Die Addition und Subtraktion sind in der Klasse *AddSub*, das Minimum und Maximum in der Klasse *MinMax* und die Faltung und Entfaltung in der Klasse *Convolution*. Zusätzlich gibt es noch eine Klasse *Basicoperations*, welche noch eine Methode zum Berechnen des Backlogs *backlog()*, eine Methode zum Berechnen des Virtual Delays *virtual_delay()* und eine Methode um stückweise lineare Funktionen auf der x -Achse zu verschieben *shift_n()*. Die Aufteilung der einzelnen Operationen, wie die Addition und Subtraktion in eigene Klassen, wurde so gewählt, dass die verschiedenen Operationen klar von einander getrennt werden können und auch Helfermethoden konkret zugewiesen werden können.

Neben den drei Hauptdateien existieren es noch weitere Dateien mit Helferklassen, die im package *Func.Helpers* zu finden sind. Dort befinden sich jeweils Helferklassen für die Faltung *ConvolutionHelpers.py*, die Entfaltung *DeconvolutionHelpers.py*, das Minimum und Maximum *MinMaxHelpers.py* und allgemeine Helfermethoden, die von allen Basisoperationen benötigt werden *Helpers.py*, wie etwa das Verlängern der stückweisen linearen Funktionen *extend_lin_func()*. Auch hier wurden die einzelnen Helfermethoden in Klassen aufgeteilt, um den Überblick zu verbessern und um Importzyklen zu verhindern.

Zusätzlich befinden sich im Ordner *Tests* diverse Unit-Tests, die die stückweise lineare Funktion und die Basisoperationen testen sollen. Bei den Tests zur stückweisen linearen Funktion, wurde in *Tests.Linear_function_TESTS.FxTests.py* die Methode zum Auswerten einer stückweisen linearen Funktion *f()* getestet. Wichtig war hier, dass vor allem die periodische Wiederholung der stückweisen linearen Funktion richtig funktioniert und die stückweise lineare Funktion wie in

Kapitel 3.2 definiert, ihre Steigung zur linken Seite auf der x -Achse hin hält.

Die Tests für die Basisoperationen befinden sich im Ordner *Tests.basic_operation_TESTS* und haben pro Basisoperation eine eigene Testklasse. Bei den Tests für die Addition, Subtraktion, das Minimum und das Maximum werden verschiedene Extremfälle getestet, die bei diesen Operationen auftreten können. Beispielsweise wird beim Minimum und Maximum vor allem getestet, ob Überschneidungen im periodischen Teil richtig erkannt werden und die Funktionen entsprechend verlängert werden.

Bei den Tests für die Faltung und Entfaltung werden im Gegensatz zur Addition, Subtraktion, dem Minimum und Maximum, keine Extremfälle sondern verschiedene Lemmas und Regeln aus dem Buch von Le Boudec et. al.[LT01] überprüft.

Als ein Schwachpunkt der Implementierung hat sich nachträglich die Art, wie die Länge der einzelnen linearen Teilstücke gespeichert wird, herausgestellt. Diese wird aktuell als eine Gleitkommazahl realisiert, was bei mehrmaligem Teilen zu Ungenauigkeiten führen kann. Da die stückweisen linearen Funktionen für die Addition und Subtraktion, sowie für das Minimum und Maximum in gleiche Stücke aufgeteilt werden, um über gleiche Teilstücke die Berechnungen ausführen zu können, führt dies zu Problemen. Ein Beispiel sind die folgenden Codezeilen aus der Methode *split_linear_pieces()* in der Klasse *Func.Helper.Helpers*, welche für sehr klein aufgelöste Gleitkommazahlen für die Länge des linearen Teilstücks zu Problemen führen.

```
if p2.x1 < p1.x1 < p2.x1 + p2.length:
    s1 = LinearPiece(p2.x1, p2.y, p2.yPlus, p1.x1 - p2.x1, p2.slope)
    new_y = p2.y + (p2.slope * s1.length)
    s2 = LinearPiece(p1.x1, new_y, new_y, p2.x1 + p2.length - p1.x1, p2.slope)
    lin_pieces2.remove(p2)
    lin_pieces2.insert(i, s1)
    lin_pieces2.insert(i + 1, s2)
    break
# end if
```

Da $p2.x1$ und $p2.length$ eine sehr weit auseinanderliegende Auflösung haben können, kann es passieren, dass es bei $p2.x1 + p2.length$ zu einem Genauigkeitsverlust kommt [Gol91] und das Ergebnis kleiner ist als $p1.x1$, obwohl $p1.x1$ eigentlich innerhalb des Bereiches von $p2$ liegt. Dadurch wird das entsprechende lineare Teilstück nicht aufgeteilt und anschließend nicht mehr über die gleichen Abschnitte der linearen Teilstücke gerechnet. Bei Operationen wie der Faltung und Entfaltung, bei denen sehr oft das Minimum und Maximum über viele Teilstücke berechnet wird, kann es dadurch zu Fehlern kommen. Für eine Lösung müsste hier die Länge und der $x1$ Wert eines linearen Teilstücks so gehalten werden, dass $p.x1 + p.length$ ohne Genauigkeitsverluste berechnet werden kann.

4 Evaluation

Zur Evaluation wurden verschiedene zufällig generierte stückweise lineare Funktionen miteinander gefaltet, entfaltet, addiert und deren Minimum bestimmt. Dabei wurde auf die Auswirkung verschiedener Parameter, wie die Länge des nicht periodischen Intervalls, die Anzahl der linearen Stücke in den einzelnen Funktionen und die Länge des periodischen Intervalls, auf die Laufzeit betrachtet. Dazu wurden in gleichen Abständen die entsprechenden Testparameter für die einzelnen Funktionen erhöht und die einzelnen Zeiten für die Berechnungen gemessen. Da erste Testreihen mit kleinen Werten für die verschiedenen Testparameter keine großen Abweichung bei den Zeiten bei mehrmaligem Ausführen gezeigt hatten, wurden die Tests mit den größeren Werten für die einzelnen Testparameter nur jeweils einmal ausgeführt.

4.1 Testdaten

Die stückweisen linearen Funktionen zur Evaluation wurden mit einem Zufallsgenerator erstellt. Dieser kann stückweise lineare Funktionen mit zufälliger Anzahl an linearen Teilstücken und zufälliger Steigung und Länge erzeugen. Neben dem initialen Seed ist es möglich über optionale Parameter auch die Anzahl der linearen Teilstücke im periodischen, sowie im nicht periodischen Intervall, die Länge des periodischen und nicht periodischen Intervalls T und d und das Steigungsoffset des periodischen Teils c zu bestimmen. Der y und $yPlus$ Wert sowie die Steigung p von den einzelnen linearen Teilstücken werden zufällig basierend auf dem Seed generiert.

4.2 Testergebnisse

Alle Tests wurden auf einem Windows 10 System mit einem *Intel Core i5 6600k*, 8 Gigabyte RAM und der Pythonversion 3.5.4 ausgeführt. Da es wie in Kapitel 3.4 bereits angesprochen, zu Fehlern in den Berechnungen kommen kann, sind in den Tabellen mit den Testergebnissen die fehlgeschlagenen Durchläufe mit einer *n.a.* eingetragen. Es ist außerdem zu erwähnen, dass alle Tests ohne jegliche Parallelisierung gelaufen sind, da die Implementierung derzeit nur single threaded läuft.

4.2.1 Anzahl linearer Stücke

Als erstes wurde überprüft, welche Auswirkungen die Anzahl an linearen Teilstücken in den stückweisen linearen Funktionen auf die Laufzeit haben, während die Länge der stückweisen linearen Funktionen gleich bleibt. Dazu wurde zum einen die Anzahl der linearen Teilstücke im nicht periodischen Teil und zum anderen die Anzahl der linearen Teilstücke im periodischen Teil der stückweisen linearen Funktionen verändert, um deren Einfluss auf die Laufzeit zu bestimmen. Um

die Testergebnisse nicht von anderen Faktoren zu beeinflussen, wurden zwei Basisfunktionen gewählt, bei denen in den verschiedenen Durchläufen nur die Anzahl der linearen Teilstücke verändert wurde. Für die Addition und das Minimum hatten sowohl f_1 als auch f_2 bei den Tests für den nicht periodischen Teil eine periodische Länge von $d = 20$ die auf 10 lineare Teilstücke aufgeteilt wurde und einen periodischen Beginn von $T = 1050$. Bei den Tests für den periodischen Teil war der periodische Beginn bei $T = 20$ mit 10 linearen Teilstücken im nicht periodischen Bereich und der periodische Bereich hatte eine Länge von $d = 1050$.

Addition: In den Tabellen 4.1 und 4.2 sind die Ergebnisse der Tests für die Addition und eine sich verändernde Anzahl von linearen Teilstücken zu sehen. Bei den Tests wurden Funktionen mit 350 - 500 linearen Teilstücken miteinander addiert und die Zeiten der Berechnungen genommen. Wie zu erwarten, gab es keinen nennenswerten Unterschied zwischen den Tests zum periodischen und nicht periodischen Teil. Während die Addition mit jeweils 500 nicht periodischen linearen Teilstücken in f_1 und f_2 187.45 Millisekunden braucht, benötigt die Addition mit jeweils 250 periodischen linearen Teilstücken in f_1 und f_2 203.07 Millisekunden. Dies ist ein Unterschied von gerade einmal 8%.

Zum Wachstum der Laufzeit ist zu sagen, dass die Dauer der Addition nahezu linear mit der Anzahl der linearen Teilstücke wächst wie in Abbildung 4.1 zu sehen ist. Dort werden die einzelnen Werte aus der Diagonalen von Tabelle 4.1 und 4.2 dargestellt. Während eine Addition mit jeweils 350 periodischen linearen Teilstücken 109.34 Millisekunden braucht, benötigt eine Addition mit 500 periodischen linearen Teilstücken 203.07 Millisekunden, was 85% mehr sind. Somit lässt sich die vorhergesagte Laufzeit aus Proposition 3.3.1 bestätigen.

Minimum: In den Tabellen 4.3 und 4.4 sind die Ergebnisse der Tests für das Minimum und eine sich verändernde Anzahl von linearen Teilstücken zu sehen. Bei den Tests wurden bei Funktionen mit 350 - 500 linearen Teilstücken das Minimum bestimmt und die Zeiten der Berechnungen genommen. Die Entwicklung der Laufzeit verhält sich dabei ähnlich zur Addition und wächst mit steigender Anzahl linearer Teilstücke nahezu linear an, was die Laufzeit aus Proposition 3.3.2 bestätigt. Es gibt dabei wieder keinen nennenswerten Unterschied zwischen den Tests für die periodische Anzahl von linearen Teilstücken oder den Tests für die nicht periodische Anzahl von linearen Teilstücken. Während die Berechnung des Minimum mit jeweils 350 linearen Teilstücken 111.72 Millisekunden braucht, benötigt die Berechnung des Minimums mit 500 linearen Teilstücken 187.45 Millisekunden, was 67% mehr sind.

In Abbildung 4.2 wird der Wert der Diagonalen von Tabelle 4.3 und 4.4 nochmals grafisch dargestellt.

Faltung: Für die Faltung hatte f_1 und f_2 bei den Tests für den nicht periodischen Teil eine periodische Länge von $d = 2$, die auf ein lineares Teilstück aufgeteilt wurde und einen periodischen Beginn von $T = 110$. Bei den Tests für den periodischen Teil war der periodische Beginn bei $T = 2$ mit einem linearen Teilstück im nicht periodischen Bereich und der periodische Bereich hatte eine Länge von $d = 55$.

In den Tabellen 4.5 und 4.6 sind die Ergebnisse der Tests für die Faltung und eine sich verändernde Anzahl von linearen Teilstücken zu sehen. Wie zu sehen ist, hat die Anzahl der linearen Teilstücke eine große Auswirkung auf die Faltung. Sowohl bei dem Test für die nicht periodische Anzahl linearer Teilstücke als auch dem Test für die periodische Anzahl von linearen Teilstücken steigt die Laufzeit für eine Berechnung sehr stark an. Während eine Faltung mit nur 10 nicht periodischen linearen Teilstücken bei den beiden Funktionen f_1 und f_2 nur 44.2959 Sekunden braucht, dauert

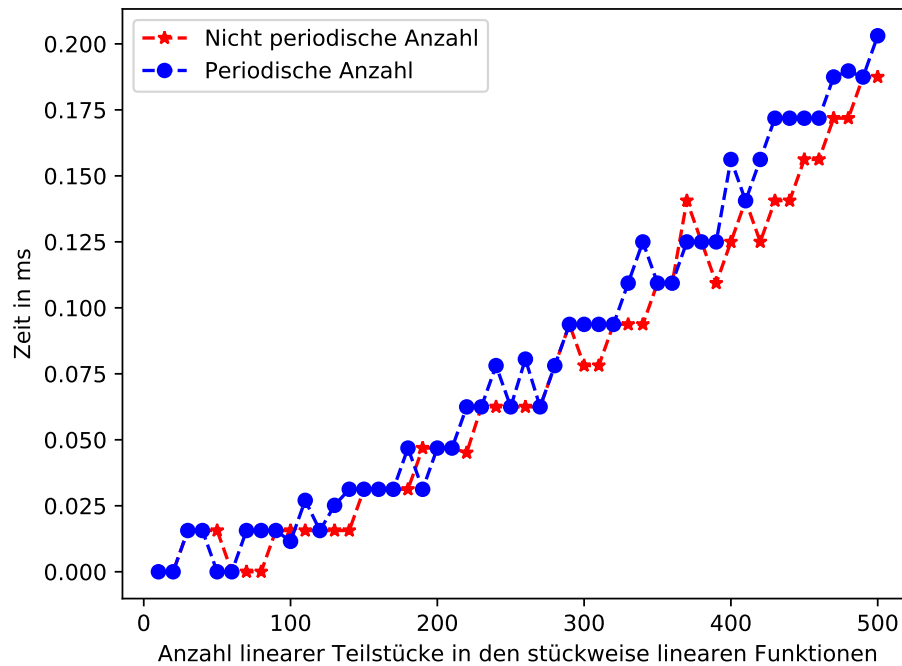


Abbildung 4.1: Laufzeit für die Addition für eine variierende Anzahl von linearen Teilstücken in den stückweisen linearen Funktionen.

eine Faltung mit 50 linearen Teilstücken bei beiden Funktionen schon 317.979 Sekunden und bei einer Anzahl von 100 linearen Teilstücken bei beiden Funktionen 1876.39 Sekunden. Bei den Tests zur periodischen Anzahl ist ein ähnliches Wachstum zu beobachten, mit 7.745 Sekunden für 10 lineare Teilstücke bei beiden Funktionen und 225.016 Sekunden bei 50 linearen Teilstücken bei beiden Funktionen. Die Werte der Tests für die Anzahl linearer Teilstücke im periodischen Teil sind in diesem Fall kleiner, weil die periodische Länge der stückweisen linearen Funktionen für die Tests kleiner waren. Wie im folgenden Kapitel 4.2.2 zu den Tests für die periodische und nicht periodische Länge zu sehen ist, hat die Länge des periodischen Teils eine größere Auswirkung auf die Faltung, als die Länge des nicht periodischen Teils. Daher wurde für die Tests des periodischen Teils in diesem Kapitel eine kürzere Länge für den periodischen Teil gewählt, da die Faltung für eine periodische Länge von 100 mit jeweils 40 linearen Teilstücken bereits mehr als 2 Stunden gebraucht hätte.

Weiter ist zu beobachten, dass sich die Faltung bei einer vertauschten Anzahl linearer Teilstücke in den Funktionen nicht gleich verhält. Während eine Faltung mit 90 linearen Teilstücken im nicht periodischen Teil in f_1 und 80 linearen Teilstücken im nicht periodischen Teil von f_2 1174.39 Sekunden benötigt, braucht die Faltung mit 80 nicht periodischen linearen Teilstücken in f_1 und 90 nicht periodischen linearen Teilstücken in f_2 nur 1077.75 Sekunden. Dies könnte durch eine ungünstige Reihenfolge beim Zusammensetzen der einzelnen Faltungen von Spots und Segmenten durch deren Minimum zu erklären sein (siehe Algorithmus 3.4).

Als eine letzte Beobachtung fällt auf, dass bei den Tests für den periodischen Teil sehr viele Fehler aufgetreten sind, während es bei den Tests für den nicht periodischen Teil keinen einzigen Fehler gab.

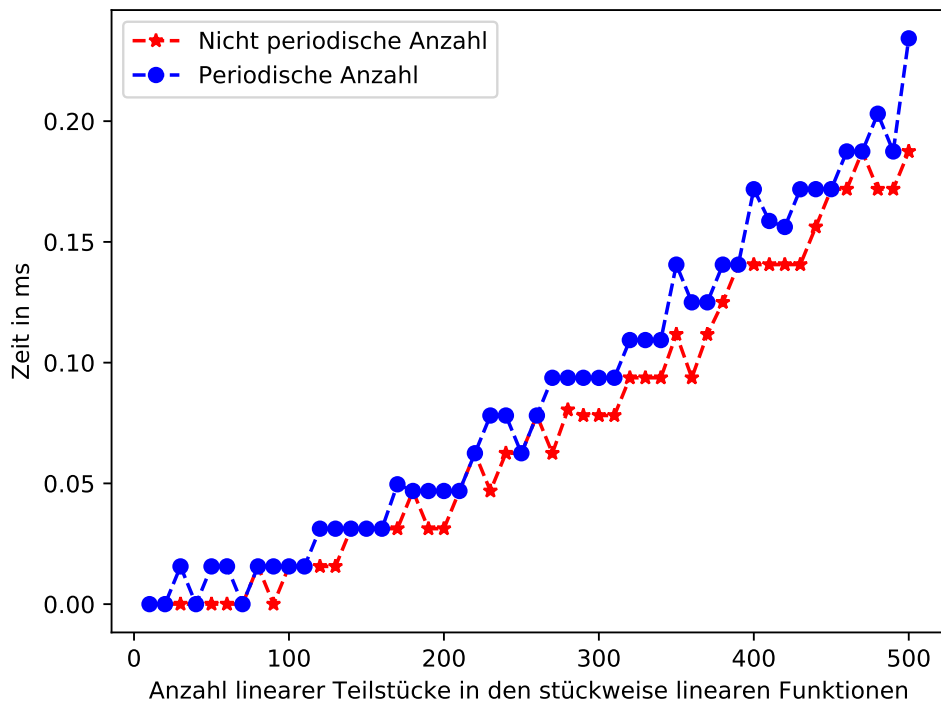


Abbildung 4.2: Laufzeit für das Minimum für eine variierende Anzahl von linearen Teilstücken in den stückweisen linearen Funktionen.

Die Faltung wird für eine große Anzahl an linearen Teilstücken also instabiler in ihrer Berechnung. In Abbildung 4.3 wird der Wert der Diagonalen von Tabelle 4.5 und 4.6 nochmals grafisch dargestellt.

Entfaltung: Für die Entfaltung hatten sowohl f_1 als auch f_2 bei den Tests für den nicht periodischen Teil eine periodische Länge von $d = 2$ die auf ein lineares Teilstück aufgeteilt wurde und einen periodischen Beginn von $T = 110$. Bei den Tests für den periodischen Teil war der periodische Beginn bei $T = 2$ mit einem linearen Teilstück im nicht periodischen Bereich und der periodische Bereich hatte eine Länge von $d = 110$.

In den Tabellen 4.7 und 4.8 sind die Ergebnisse der Tests für die Entfaltung und eine sich verändernde Anzahl von linearen Teilstücken zu sehen. Die Entwicklung für die Laufzeiten der einzelnen Tests verhält sich dabei ähnlich wie die bei der Faltung. Für eine wachsende Anzahl der linearen Teilstücke wächst die Laufzeit der Entfaltung stark an. Während eine Entfaltung mit zwei Funktionen, die jeweils 10 periodische lineare Teilstücke haben, nur 9,132 Sekunden braucht dauert eine Entfaltung mit jeweils 50 periodischen linearen Teilstücken schon 310.341 Sekunden und die Entfaltung mit jeweils 100 periodischen linearen Teilstücken 1220.9 Sekunden. Erstaunlich ist dieses Mal, dass die Entfaltungen mit einer größer werdenden Anzahl an nicht periodischen linearen Teilstücken, länger braucht, als die mit einer größer werdenden periodischen Anzahl von linearen Teilstücken. Die selbe Entfaltung mit jeweils 100 nicht periodischen linearen Teilstücken brauchte in dieser Versuchreihe 2290.88 Sekunden was fast das Doppelte ist.

Es fällt außerdem auf, dass die Entfaltung sowie die Faltung nicht symmetrisch ist, was die Laufzeit

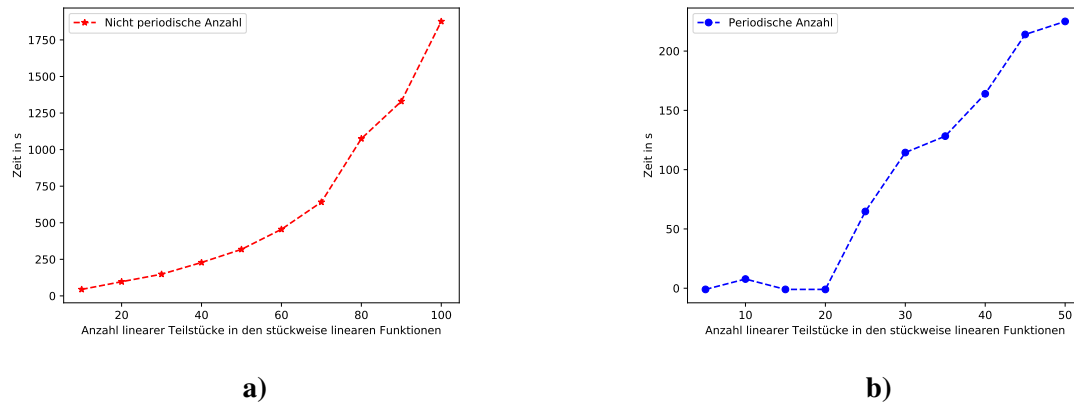


Abbildung 4.3: Laufzeit für die Faltung für eine variierende Anzahl von linearen Teilstücken in den stückweisen linearen Funktionen, für eine nicht periodische Länge von 100 a) und eine periodische Länge von 50 b).

angeht. Beispielsweise brauchte die Entfaltung von 50 periodischen linearen Teilstücken in f_1 und 60 periodischen linearen Teilstücken in f_2 366.975 Sekunden und die Faltung mit 60 periodischen linearen Teilstücken in f_1 und 50 periodischen linearen Teilstücken in f_2 braucht 372.878 Sekunden. Dies kann in Tabelle 4.7 und 4.8 mehrfach beobachtet werden.

In Abbildung 4.4 wird der Wert der Diagonalen von Tabelle 4.7 und 4.8 nochmals grafisch dargestellt.

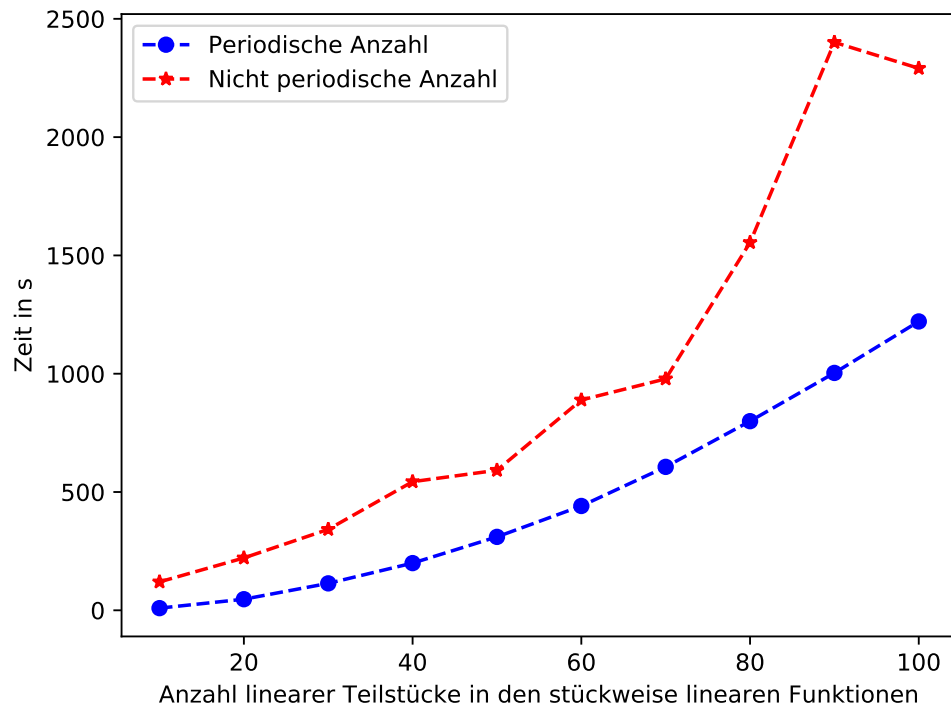


Abbildung 4.4: Laufzeit für die Entfaltung für eine variierende Anzahl von linearen Teilstücken in den stückweisen linearen Funktionen.

	350	360	370	380	390	400	410	420	430	440	450	460	470	480	490	500
350	109.34	93.728	93.729	109.34	109.34	109.35	109.34	124.97	124.97	124.97	124.97	124.97	124.97	124.97	140.59	124.96
360	95.968	109.34	109.36	109.32	109.35	109.34	140.59	109.34	124.97	124.97	124.99	124.95	124.97	124.97	140.59	140.59
370	109.34	111.98	140.59	109.34	109.34	124.97	124.97	124.97	124.97	124.97	124.97	140.59	124.97	124.96	140.59	140.59
380	110.77	109.34	124.97	124.97	124.99	124.94	124.96	140.59	140.59	124.97	140.59	124.97	140.59	140.59	140.59	156.21
390	109.34	140.59	124.97	109.34	109.34	124.97	124.97	124.97	124.97	124.97	156.21	140.59	140.59	140.59	140.61	140.59
400	109.34	109.34	109.34	156.21	124.96	124.97	124.97	140.59	124.97	140.59	140.61	140.57	140.59	124.99	140.59	156.21
410	124.96	124.97	124.97	124.97	156.21	124.97	140.59	140.59	124.97	124.97	140.57	140.59	156.21	140.59	156.21	156.21
420	124.97	124.97	124.97	124.97	156.21	124.99	140.59	124.99	140.59	140.59	140.59	156.21	156.21	156.21	140.59	140.59
430	124.97	124.97	156.21	140.59	124.97	140.62	140.58	140.59	140.59	140.59	171.83	156.21	142.84	156.21	156.21	156.23
440	156.21	140.61	124.98	124.97	140.61	140.57	140.61	140.57	140.59	140.59	142.98	156.23	156.19	156.21	156.21	171.83
450	124.97	124.97	143.28	140.59	156.21	140.59	140.59	140.59	156.21	156.21	156.21	156.21	156.21	156.21	171.83	156.21
460	140.59	140.59	140.61	140.56	140.59	156.21	156.21	171.83	156.21	156.21	156.21	156.21	171.85	156.19	171.83	171.83
470	140.59	171.83	140.59	156.21	156.21	140.59	156.21	156.21	171.83	156.21	156.21	158.77	171.85	171.81	156.21	174.26
480	140.61	156.19	156.21	156.21	140.59	147.10	156.21	156.23	171.81	171.83	156.21	156.21	171.83	171.83	171.83	171.83
490	156.21	156.21	156.21	156.21	156.21	156.21	156.21	171.83	187.45	156.21	156.21	171.83	171.83	171.83	187.45	171.85
500	156.21	156.21	156.21	171.85	156.19	156.21	171.83	171.83	156.21	156.23	156.19	187.45	189.74	187.45	171.83	187.45

Tabelle 4.1: Ergebnisse für eine variierende Anzahl linearer Teilstücke im nicht periodischen Teil bei der Addition in ms.

	350	360	370	380	390	400	410	420	430	440	450	460	470	480	490	500
350	109.34	124.97	124.97	109.34	109.34	140.59	124.97	124.97	140.59	124.97	156.21	140.59	140.59	140.59	140.59	140.59
360	124.97	109.34	109.34	109.34	124.97	156.21	140.59	124.97	124.97	140.59	140.59	140.59	156.21	156.21	140.59	140.59
370	124.97	124.99	124.97	124.94	140.59	124.99	124.95	140.59	124.97	140.59	140.61	140.57	140.61	156.21	156.19	140.59
380	124.97	124.97	124.97	124.97	124.97	124.97	124.96	140.59	140.59	124.95	140.59	140.59	156.21	140.59	156.21	171.83
390	124.97	124.97	124.97	140.59	124.97	124.97	140.59	140.59	156.23	140.57	156.21	140.59	140.59	171.83	182.04	156.21
400	140.59	124.97	140.59	140.59	140.59	156.21	140.59	140.59	159.46	156.21	140.61	156.19	187.45	156.21	156.21	171.83
410	140.59	134.73	156.21	140.61	156.19	140.61	140.57	156.21	156.21	156.21	156.21	156.21	171.83	156.21	171.83	156.21
420	140.59	140.59	140.59	140.59	140.59	140.59	156.21	156.21	156.21	156.23	156.19	156.21	156.21	187.45	171.83	156.21
430	140.59	156.21	140.59	140.59	140.59	156.21	140.59	140.57	171.83	156.21	171.83	156.21	171.83	171.83	171.83	187.45
440	156.21	156.21	156.23	156.19	156.21	156.23	157.48	156.24	171.80	171.83	171.83	171.85	171.81	171.83	171.83	187.45
450	156.23	156.19	156.21	140.59	156.21	171.83	156.21	156.23	187.43	171.83	171.83	171.83	171.83	187.45	187.45	203.07
460	156.21	156.21	156.21	171.83	156.21	156.21	171.83	171.83	171.83	171.83	171.83	171.86	187.42	190.27	187.45	187.45
470	156.21	158.66	156.21	156.21	171.83	171.83	171.83	171.83	171.83	171.83	171.83	187.45	187.45	171.83	187.45	187.45
480	156.21	156.19	171.85	156.20	171.84	156.21	171.83	171.80	171.83	187.47	171.81	171.83	171.83	189.74	187.43	203.09
490	156.17	156.21	171.86	156.18	174.18	171.81	187.48	171.80	171.85	187.45	171.83	171.81	203.07	187.48	187.44	187.43
500	156.21	156.19	187.45	171.83	171.83	171.85	171.83	171.83	187.43	187.45	171.83	187.45	187.47	187.45	203.05	203.07

Tabelle 4.2: Ergebnisse für eine variierende Anzahl linearer Teilstücke im periodischen Teil bei der Addition in ms.

	350	360	370	380	390	400	410	420	430	440	450	460	470	480	490	500
350	111.72	109.34	124.97	124.97	109.37	121.58	109.36	124.93	124.97	124.97	124.97	124.97	140.59	124.97	140.61	124.94
360	109.34	93.727	111.73	109.34	109.34	109.34	124.97	109.34	109.34	140.59	140.59	124.97	124.97	124.97	140.59	140.59
370	109.34	109.32	111.68	124.97	124.97	124.97	140.59	124.96	124.97	124.97	140.59	124.97	124.97	140.59	156.18	140.59
380	109.34	143.02	109.34	124.97	124.97	124.97	124.97	124.97	140.59	124.97	140.59	140.59	124.97	140.59	140.59	140.59
390	109.34	124.97	109.34	124.97	140.59	140.59	124.97	124.97	124.97	140.61	124.94	140.56	140.59	125.00	140.55	156.21
400	109.36	124.94	109.34	124.99	124.94	140.59	140.61	124.97	140.57	140.59	140.61	140.57	156.21	156.21	140.59	156.21
410	124.97	124.97	124.97	124.97	140.59	140.59	140.59	124.97	140.59	140.59	140.59	156.21	140.59	156.21	156.21	156.21
420	124.97	140.59	124.97	124.96	140.59	140.59	156.21	140.59	140.59	140.59	156.21	140.59	187.45	156.21	156.21	156.21
430	124.97	140.59	140.59	140.59	140.59	140.59	140.59	140.59	140.61	156.19	171.83	156.21	156.21	156.23	156.21	157.48
440	124.97	124.97	140.59	140.59	142.97	140.59	156.21	156.21	142.84	156.21	140.59	140.59	156.21	156.23	187.43	156.21
450	142.75	140.59	156.21	156.21	140.59	140.59	140.59	156.21	140.59	171.83	171.83	171.83	156.21	171.83	156.21	171.85
460	156.23	140.59	140.57	140.59	171.83	156.21	140.59	171.83	156.21	156.21	156.21	171.83	171.83	156.21	156.21	156.23
470	140.57	140.59	156.21	156.21	187.45	187.45	156.21	156.21	156.21	156.21	171.85	175.06	187.45	171.83	171.83	171.83
480	143.19	156.21	140.62	156.18	156.21	187.45	156.21	156.21	156.21	171.83	156.21	171.83	171.83	171.81	171.83	171.85
490	156.23	156.22	156.18	171.83	156.21	156.21	156.21	156.19	171.83	171.83	187.45	171.83	171.83	187.45	171.83	187.45
500	171.81	156.21	156.21	156.21	156.23	158.51	156.21	187.45	187.45	171.83	171.83	171.83	171.83	187.45	203.07	187.45

Tabelle 4.3: Ergebnisse für eine variierende Anzahl linearer Teilstücke im nicht periodischen Teil beim Minimum in ms.

	350	360	370	380	390	400	410	420	430	440	450	460	470	480	490	500
350	140.59	1171.6	517.86	328.04	312.42	249.94	234.34	234.32	203.05	218.69	203.07	203.07	187.45	203.07	203.07	203.07
360	249.94	124.97	1188.4	593.60	359.31	315.65	281.18	249.94	218.69	203.07	218.69	218.69	203.07	203.07	203.07	203.07
370	187.45	265.56	124.97	1171.6	562.34	374.91	328.04	265.59	249.93	218.67	220.95	234.31	203.09	218.67	203.07	203.07
380	171.85	218.67	343.67	140.59	1202.8	546.74	390.53	312.42	265.56	234.34	234.29	218.71	218.67	218.69	221.02	218.69
390	187.47	187.45	267.76	437.41	140.57	1140.3	562.38	359.30	296.77	265.56	249.94	234.32	218.69	234.34	221.61	218.69
400	171.83	203.05	203.09	249.94	438.77	171.81	1109.1	499.88	328.04	296.80	249.96	249.96	234.32	218.69	234.30	218.72
410	171.83	187.47	203.08	249.93	312.40	624.85	158.67	1062.2	452.99	343.68	296.80	265.56	265.58	234.30	218.69	234.31
420	187.44	187.45	203.07	234.32	265.56	343.68	518.32	156.19	1047.9	468.64	359.27	312.44	265.54	249.94	281.16	249.96
430	171.80	171.83	187.47	203.05	234.32	265.56	281.18	437.42	171.80	1329.1	531.12	437.39	328.06	312.40	281.18	296.82
440	171.83	171.81	171.86	187.46	234.28	234.31	249.94	312.42	563.65	171.83	1171.5	577.99	406.15	343.66	312.40	328.06
450	171.83	174.21	187.43	187.47	218.69	218.71	234.32	265.53	314.76	718.60	171.84	1484.0	562.34	453.01	374.94	359.27
460	187.45	171.83	203.07	187.45	203.07	221.07	234.32	234.31	249.95	390.52	671.69	187.45	1468.4	734.20	484.26	424.02
470	187.42	187.45	187.47	187.45	203.08	218.67	n.a.	234.31	249.96	296.80	359.27	546.76	187.45	1642.7	765.46	546.74
480	187.43	187.45	187.45	203.07	218.72	218.66	218.71	234.29	249.96	281.18	312.40	390.53	751.18	203.07	1796.4	874.81
490	187.45	187.47	187.43	203.09	203.05	234.31	221.09	234.34	220.08	265.56	312.40	312.42	437.42	890.40	187.45	1890.1
500	187.45	189.73	187.43	203.07	218.69	218.69	218.71	203.07	218.67	265.54	281.20	281.19	343.67	453.00	734.21	234.28

Tabelle 4.4: Ergebnisse für eine variierende Anzahl linearer Teilstücke im periodischen Teil beim Minimum in ms.

	10	20	30	40	50	60	70	80	90	100
10	44.2959	68.9090	86.5749	111.237	126.457	152.998	174.181	193.934	215.578	233.677
20	65.4341	96.9077	117.291	149.958	167.312	197.811	232.614	255.211	282.114	305.886
30	85.3628	119.745	148.216	181.508	210.519	248.056	285.821	315.524	337.992	364.735
40	110.131	155.816	188.404	228.287	264.743	315.384	380.171	427.857	457.664	518.311
50	129.370	177.929	218.089	257.332	317.979	372.926	441.074	499.595	518.589	571.388
60	152.565	227.224	272.816	316.013	393.812	455.040	587.911	668.900	714.084	796.325
70	173.692	237.811	293.112	357.443	445.956	515.805	641.193	717.025	758.951	810.160
80	197.026	282.500	388.993	451.818	570.185	680.511	919.063	1074.90	1077.75	1241.08
90	225.401	343.506	464.778	519.181	661.778	857.993	1161.50	1174.39	1329.93	1426.72
100	261.019	423.290	500.983	669.391	781.298	978.908	1448.91	1527.26	1688.88	1876.39

Tabelle 4.5: Ergebnisse für eine variierende Anzahl linearer Teilstücke im nicht periodischen Teil bei der Faltung in s.

	5	10	15	20	25	30	35	40	45	50
5	n.a.	3.47088	n.a.	9.50151	12.2716	18.4984	23.7431	33.4803	39.1871	54.1569
10	n.a.	7.74524	13.3954	n.a.	25.9902	40.6400	49.1622	57.4816	68.0858	91.0820
15	n.a.	n.a.	n.a.	25.4451	33.9201	n.a.	54.4654	n.a.	81.3277	95.8476
20	n.a.	n.a.	n.a.	n.a.	n.a.	57.0802	54.4394	n.a.	n.a.	n.a.
25	14.4905	21.3734	36.6792	51.4886	64.7314	104.994	108.439	n.a.	154.826	n.a.
30	20.3834	n.a.	n.a.	57.5526	n.a.	114.417	119.270	143.384	168.348	159.420
35	n.a.	n.a.	51.5726	66.8730	88.2998	n.a.	128.272	162.335	183.014	177.981
40	32.3356	38.4393	n.a.	73.0087	97.4294	131.115	127.128	163.982	186.368	n.a.
45	n.a.	n.a.	n.a.	82.4575	120.573	n.a.	n.a.	187.904	214.012	148.295
50	n.a.	n.a.	n.a.	102.136	n.a.	229.299	182.745	253.255	n.a.	225.016

Tabelle 4.6: Ergebnisse für eine variierende Anzahl linearer Teilstücke im periodischen Teil bei der Faltung in s.

	10	20	30	40	50	60	70	80	90	100
10	120.049	191.109	274.124	365.453	417.750	513.648	597.094	718.025	828.299	956.808
20	141.785	221.371	311.821	465.675	466.525	607.235	691.248	844.655	1044.81	1204.53
30	159.073	225.153	342.139	499.908	501.210	661.042	703.035	952.139	1166.73	1354.32
40	164.690	241.109	359.370	543.693	557.992	735.568	824.458	1031.19	1334.28	1520.11
50	187.884	276.089	407.082	610.732	591.837	878.090	867.146	1273.06	1682.12	1775.99
60	199.189	284.143	422.018	683.791	659.148	888.891	954.711	1365.23	1791.48	1908.33
70	198.667	301.780	445.741	712.822	718.740	1018.07	978.226	1533.02	2092.83	2315.91
80	198.085	301.298	442.238	726.517	656.834	1003.96	996.373	1554.50	2132.35	2223.70
90	211.249	298.517	476.666	746.064	743.887	1137.29	1036.69	1699.35	2400.56	2207.81
100	223.392	335.400	527.657	821.821	780.390	1330.13	1083.81	1896.48	2584.34	2290.88

Tabelle 4.7: Ergebnisse für eine variierende Anzahl linearer Teilstücke im nicht periodischen Teil bei der Entfaltung in s.

	10	20	30	40	50	60	70	80	90	100
10	9.13206	22.5477	36.9473	50.9853	66.1461	78.2317	90.8607	107.004	122.662	136.030
20	20.7570	46.9097	72.2218	99.3655	126.981	149.555	173.887	203.010	229.919	255.743
30	32.7261	73.6821	113.839	154.801	191.987	225.143	257.747	303.386	347.245	385.364
40	46.4227	98.1903	150.864	199.472	251.484	297.603	346.360	400.301	455.653	506.351
50	57.8146	123.720	190.676	250.004	310.341	366.975	429.851	493.357	556.267	618.663
60	69.8015	146.717	230.285	299.840	372.878	440.972	516.114	593.521	673.276	748.647
70	85.2304	171.789	279.634	347.126	438.559	518.158	606.255	694.939	790.698	880.584
80	98.6419	201.925	320.132	407.808	502.707	593.751	696.834	799.429	896.103	993.275
90	109.897	218.592	367.941	442.554	550.994	651.273	762.231	873.883	1003.33	1140.51
100	123.584	245.252	409.459	494.979	621.575	727.162	851.146	979.945	1099.06	1220.90

Tabelle 4.8: Ergebnisse für eine variierende Anzahl linearer Teilstücke im periodischen Teil bei der Entfaltung in s.

4.2.2 Länge der linearen Funktionen

Als weiterer Punkt wurde überprüft, welche Auswirkungen die Länge der stückweisen linearen Funktionen auf die Laufzeit hat. Dazu wurde zum einen die Länge des nicht periodischen Teils und zum anderen die Länge des periodischen Teils der stückweisen linearen Funktionen verändert. Damit die Testergebnisse nicht von anderen Faktoren beeinflusst werden, wurden wieder zwei Basisfunktionen gewählt, bei denen dieses Mal in den verschiedenen Durchläufen nur die Länge des periodischen oder nicht periodischen Teils verändert wurde, die Anzahl der linearen Teilstücke aber gleich blieb. Für die Addition und das Minimum hatten sowohl f_1 als auch f_2 bei den Tests für den nicht periodischen Teil eine periodische Länge von $d = 20$ die auf 10 lineare Teilstücke aufgeteilt wurde und 10 lineare Teilstücke im nicht periodischen Teil. Bei den Tests für den periodischen Teil war der periodische Beginn bei $T = 20$ mit 10 linearen Teilstücken im nicht periodischen Bereich und der periodische Bereich hatte 10 lineare Teilstücke.

Addition: In den Tabellen 4.9 und 4.10 sind die Ergebnisse der Tests für die Addition und eine sich verändernde Länge der stückweisen linearen Funktionen zu sehen. Bei der Addition ist wie zu erwarten, kein nennenswerter Unterschied bei der Länge der stückweisen linearen Funktionen zu erkennen. Weder bei den periodischen noch bei nicht periodischen Tests gab es eine merkbare Veränderung der Testergebnisse für eine größer werdende Länge der stückweisen linearen Funktion. Während die Addition mit einer nicht periodischen Länge von jeweils 350 in f_1 und f_2 15.620 Millisekunden braucht, benötigt die Addition mit einer periodischen Länge von 500 in f_1 und f_2 15.622 Millisekunden. Generell schwanken die Werte der Messung für die periodische und auch nicht periodische Länge zwischen 2-4 unterschiedlichen Werten.

Eine Auswirkung der Länge des periodischen oder nicht periodischen Teils auf die Laufzeit der Addition konnte also nicht festgestellt werden.

Minimum: In den Tabellen 4.11 und 4.12 sind die Ergebnisse der Tests für das Minimum und eine sich verändernde Länge der stückweisen linearen Funktionen zu sehen. Wie bei der Addition gibt es keinen nennenswerten Unterschied für eine wachsende Länge der stückweisen linearen Funktionen. Auch hier konnte also keine Auswirkung der Länge des periodischen oder nicht periodischen Teils auf die Laufzeit festgestellt werden

Faltung: Für die Faltung hatten sowohl f_1 als auch f_2 bei den Tests für den nicht periodischen Teil eine periodische Länge von $d = 2$, die auf ein lineares Teilstücke aufgeteilt wurde und 2 lineare Teilstücke im nicht periodischen Teil. Bei den Tests für den periodischen Teil war der periodische Beginn bei $T = 2$ bei einem linearen Teilstück im nicht periodischen Bereich und der periodische Bereich hatte 2 lineare Teilstücke.

In den Tabellen 4.13 und 4.14 sind die Ergebnisse der Tests für die Faltung und eine sich verändernde Länge der stückweisen linearen Funktionen zu sehen. Wie zu sehen ist, hat die Länge des periodischen Teils der stückweise linearen Funktionen eine deutlich größere Auswirkung auf die Laufzeit als die Länge des nicht periodischen Teils der stückweisen linearen Funktionen. Die längste Dauer für eine Faltung für die nicht periodischen Tests lag bei 9.01581 Sekunden, mit einer nicht periodischen Länge von 100 für die beiden stückweise linearen Funktionen. Bei den Tests zur periodischen Länge brauchte die längste Faltung 470.52 Sekunden mit einer periodischen Länge von 90 für f_1 und 100 für f_2 .

Weiterhin ist zu beobachten, dass die Zeiten für die periodischen Tests sehr stark schwanken. Die

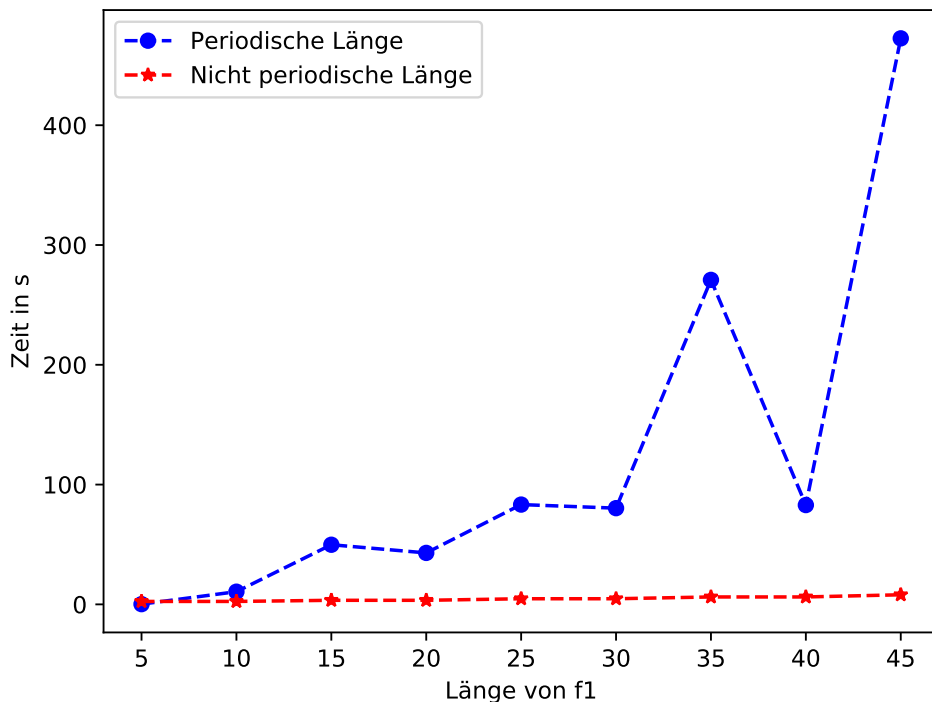


Abbildung 4.5: Laufzeit für die Faltung für eine variierende Länge der stückweisen linearen Funktionen.

Messung für eine periodische Länge von 90 für f_1 und 80 für f_2 brauchte beispielsweise 401.781 Sekunden, während die Messung für eine periodische Länge von 90 für beide stückweisen linearen Funktionen nur 0.09635 Sekunden brauchte. Dies ist so zu erklären, dass die beiden stückweisen linearen Funktionen f_1 und f_2 abhängig vom kgV der beiden Längen des periodischen Teils verlängert werden. Daher sind die Tests auf der Diagonalen der Ergebnisstabelle sehr klein, weil das kgV der beiden Funktionen dort immer den Wert der Diagonalen hat und die beiden Funktionen dementsprechend nicht verlängert werden müssen.

Außerdem ist erneut aufgefallen, dass es auch beim Verändern der periodischen Länge bei der Faltung oft zu Fehlern kommt.

In Abbildung 4.5 sind die Ergebnisse aus den Tabellen nochmals grafisch dargestellt. Dieses Mal wurde dazu für jede Länge der ersten stückweisen linearen Funktion f_1 der größte Wert für die Faltung genommen.

Entfaltung: Für die Entfaltung hatten sowohl f_1 als auch f_2 bei den Tests für den nicht periodischen Teil eine periodische Länge von $d = 2$, die auf ein lineares Teilstück aufgeteilt wurde und 10 lineare Teilstücke im nicht periodischen Teil. Bei den Tests für den periodischen Teil war der periodische Beginn bei $T = 2$ bei einem linearen Teilstück im nicht periodischen Bereich und der periodische Bereich hatte 10 lineare Teilstücke.

In den Tabellen 4.15 und 4.16 sind die Ergebnisse der Tests für die Entfaltung und eine sich ändernde Länge der stückweisen linearen Funktionen zu sehen.

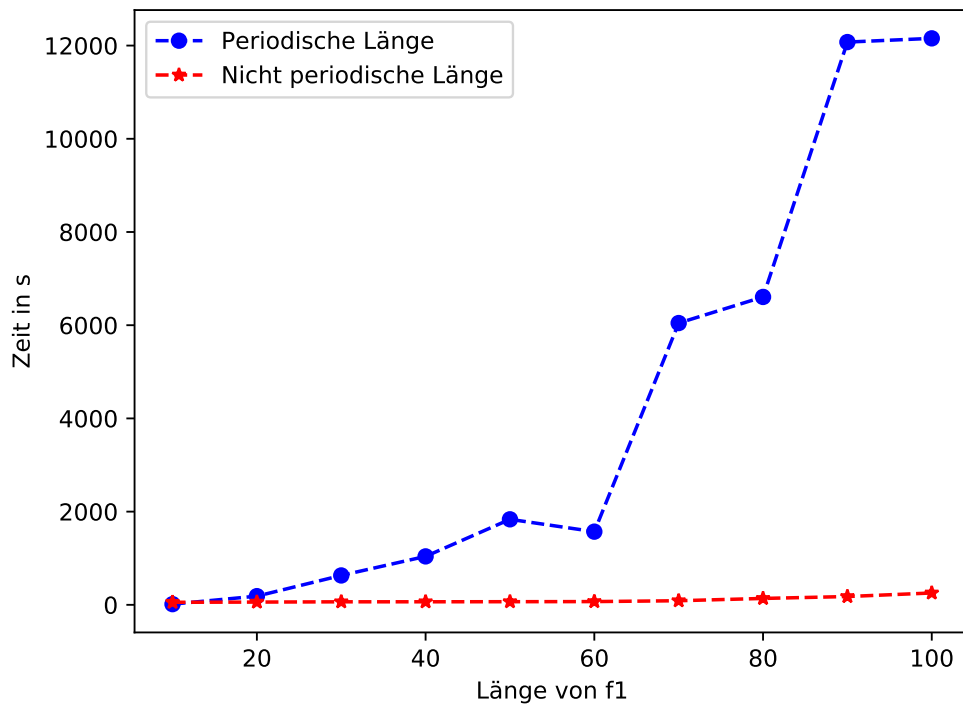


Abbildung 4.6: Laufzeit für die Faltung für eine variierende Länge der stückweisen linearen Funktionen.

Wie zu sehen ist, hat auch hier die Länge des periodischen Teils der stückweise linearen Funktionen eine deutlich größere Auswirkung auf die Laufzeit als die Länge des nicht periodischen Teils der stückweisen linearen Funktionen. Die längste Dauer für eine Entfaltung für die nicht periodischen Tests lag bei 254.489 Sekunden, mit einer nicht periodischen Länge von 100 für die erste und eine nicht periodische Länge von 40 für die zweite stückweise lineare Funktion. Bei den Tests zur periodischen Länge brauchte die längste Faltung 12154.9 Sekunden mit einer periodischen Länge von 10 für die erste und 90 für die zweite stückweise lineare Funktion.

Wie bei der Faltung gab es auch bei den Tests zur Entfaltung starke Unterschiede zwischen den Ergebnissen auf der Diagonalen der Tabelle und den anderen Werten. Die Entfaltung auf der Diagonalen brauchte hier im schlechtesten Fall 8.12586 Sekunden bei den periodischen Tests, während der schlechteste Wert insgesamt bei 12154.9 Sekunden für die periodischen Tests lag.

In Abbildung 4.6 sind die Ergebnisse aus den Tabellen nochmals grafisch dargestellt. Auch hier wurde wie bei der Faltung für jede Länge der ersten stückweisen linearen Funktion f_1 der größte Wert für die Entfaltung genommen.

	350	360	370	380	390	400	410	420	430	440	450	460	470	480	490	500
350	15.621	15.620	15.620	15.642	15.621	31.222	15.621	31.242	15.621	15.621	31.242	31.243	31.242	31.241	15.621	15.620
360	15.620	31.242	15.620	15.622	31.242	15.620	31.242	15.621	31.242	15.621	31.243	31.242	15.621	31.243	31.242	31.242
370	15.621	15.621	15.621	31.242	15.621	15.621	15.621	15.620	15.620	31.242	15.621	31.242	15.621	15.621	31.242	31.263
380	31.242	15.621	15.620	31.242	15.621	15.620	31.242	15.621	31.241	15.622	31.241	15.621	31.241	31.243	15.621	31.242
390	15.621	15.621	31.264	15.620	15.601	31.242	15.621	15.621	31.243	15.620	31.242	15.621	15.621	31.242	31.242	31.242
400	15.621	15.621	15.621	15.621	31.242	15.621	15.621	31.243	31.242	31.242	15.621	31.242	15.621	15.621	31.242	15.621
410	31.242	31.243	15.621	15.620	15.621	15.621	15.641	15.621	15.621	31.242	15.621	15.620	15.621	31.242	15.621	31.242
420	31.263	15.621	31.242	15.620	15.600	15.621	15.621	15.621	31.242	15.621	15.620	15.621	15.621	15.621	31.243	15.621
430	31.242	15.621	31.242	15.620	31.242	15.621	15.620	15.621	15.621	15.621	31.242	15.621	31.242	15.621	31.242	15.620
440	15.621	31.243	15.620	31.243	15.621	31.242	15.621	15.621	31.264	15.619	15.600	31.262	15.601	31.242	15.621	31.242
450	31.242	15.621	17.683	15.622	31.242	15.621	31.242	15.621	31.243	15.621	15.621	31.242	15.620	31.242	15.621	31.242
460	31.243	15.620	31.243	15.620	19.809	15.620	31.265	15.599	15.641	15.641	15.620	15.601	15.621	15.621	31.221	15.642
470	31.242	15.621	15.621	31.242	15.621	31.242	15.621	31.243	15.620	15.621	31.242	15.621	15.621	15.621	15.621	31.242
480	15.620	31.242	31.242	15.621	31.242	15.621	31.242	15.621	31.242	15.621	15.621	15.642	15.600	31.243	15.620	15.621
490	15.621	15.621	31.241	31.243	15.621	15.621	31.242	15.621	15.621	15.621	31.242	15.620	15.621	15.621	15.600	31.242
500	31.242	15.621	15.620	31.242	15.621	15.621	31.243	15.621	31.242	15.620	31.242	15.621	31.243	15.621	15.621	31.242

Tabelle 4.9: Ergebnisse für eine variierende Länge des nicht periodischen Teils bei der Addition in ms.

	350	360	370	380	390	400	410	420	430	440	450	460	470	480	490	500
350	15.620	109.34	109.34	124.97	124.97	15.621	124.97	15.620	124.99	124.99	0.0000	156.19	171.83	156.21	0.0000	15.620
360	109.36	0.0000	109.34	31.222	15.621	0.0000	140.61	0.0000	140.56	15.621	0.0000	46.896	157.55	0.0000	171.84	46.831
370	109.36	124.95	0.0000	124.97	124.96	124.99	124.97	140.59	140.59	156.19	140.62	171.80	156.23	156.21	171.81	171.86
380	109.35	31.241	124.94	15.621	124.98	31.222	140.61	31.243	140.59	31.244	156.19	46.884	156.21	46.863	187.43	46.863
390	109.35	31.236	109.34	109.35	0.0000	124.97	140.57	15.621	156.21	156.23	15.617	156.21	171.83	17.024	171.83	187.48
400	0.0000	15.622	140.57	31.242	140.61	0.0000	140.59	46.864	140.59	15.622	15.621	46.843	156.24	0.0000	171.83	0.0000
410	124.96	143.07	124.96	124.99	140.60	140.59	0.0000	156.21	140.59	156.21	171.80	171.83	171.85	171.83	187.45	171.81
420	0.0000	15.600	124.99	46.864	15.601	31.262	156.19	0.0000	156.21	46.890	15.595	46.864	171.85	15.620	0.0000	62.486
430	124.95	124.97	124.95	156.23	140.59	140.59	140.59	140.59	0.0000	156.21	156.21	187.43	171.83	187.48	187.44	187.45
440	140.56	15.621	140.59	46.884	140.59	15.601	140.61	46.863	156.19	0.0000	156.24	46.832	187.45	15.652	203.04	46.863
450	15.601	0.0000	140.59	140.59	15.621	0.0000	156.24	15.617	171.80	156.21	0.0000	187.45	171.83	15.621	187.45	15.634
460	140.59	31.243	140.61	46.864	140.59	46.843	156.24	46.864	171.80	46.863	187.51	0.0000	187.39	46.885	187.43	46.864
470	156.19	140.59	140.61	140.59	156.19	156.21	156.23	171.81	171.85	171.81	187.45	171.83	0.0000	187.45	203.05	203.07
480	124.97	15.621	142.00	46.843	15.642	0.0000	171.81	0.0000	171.86	15.618	31.235	46.843	205.59	0.0000	203.09	46.844
490	15.620	140.59	140.57	156.21	156.21	171.83	171.83	0.0000	171.83	171.83	187.47	187.45	203.05	203.09	0.0000	204.44
500	15.614	46.863	156.19	46.863	156.21	15.621	156.23	46.844	171.83	62.506	0.0000	62.464	187.45	62.506	203.05	15.622

Tabelle 4.10: Ergebnisse für eine variierende Länge des periodischen Teils bei der Addition in ms.

	350	360	370	380	390	400	410	420	430	440	450	460	470	480	490	500
350	0.0000	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	15.620	0.0000	15.621	0.0000	15.621	0.0000	15.620	15.641	15.621
360	15.621	0.0000	0.0000	0.0000	0.0000	15.628	0.0000	0.0000	15.623	0.0000	15.623	0.0000	15.621	0.0000	0.0000	15.626
370	0.0000	0.0000	15.622	0.0000	0.0000	0.0000	15.620	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	15.620	0.0000	15.621
380	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	15.642	0.0000	0.0000	15.600	0.0000	15.640	0.0000	15.622
390	15.621	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	15.620	0.0000	0.0000	15.621	0.0000	15.620	15.621
400	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	15.622	0.0000	0.0000	15.640	0.0000	15.622	0.0000
410	0.0000	0.0000	0.0000	15.600	0.0000	0.0000	0.0000	0.0000	15.622	0.0000	0.0000	15.620	0.0000	0.0000	15.620	0.0000
420	15.620	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	15.601	0.0000	0.0000	15.641	0.0000	0.0000
430	15.621	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	15.641
440	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	15.601	0.0000	0.0000	0.0000	0.0000	15.652	0.0000	0.0000	15.590
450	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	0.0000	15.620	0.0000	0.0000	0.0000
460	0.0000	15.617	0.0000	0.0000	15.625	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
470	4.2567	0.0000	0.0000	15.649	0.0000	0.0000	15.595	0.0000	0.0000	0.0000	0.0000	15.641	0.0000	0.0000	0.0000	0.0000
480	0.0000	15.641	0.0000	0.0000	15.621	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	15.602	0.0000	0.0000	0.0000
490	0.0000	15.621	0.0000	15.621	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	15.621
500	0.0000	15.620	0.0000	15.621	0.0000	0.0000	15.620	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	15.633	0.0000

Tabelle 4.11: Ergebnisse für eine variierende Länge des nicht periodischen Teils beim Minimum in ms.

	350	360	370	380	390	400	410	420	430	440	450	460	470	480	490	500
350	0.0000	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	15.641	0.0000	0.0000	0.0000	0.0000	15.620	0.0000
360	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000
370	15.621	0.0000	15.620	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
380	15.621	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	15.620	0.0000	0.0000	0.0000	0.0000
390	15.621	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	15.620	0.0000	0.0000	0.0000	0.0000
400	0.0000	0.0000	0.0000	15.621	0.0000	15.621	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	15.628	0.0000
410	0.0000	15.621	0.0000	0.0000	15.621	0.0000	0.0000	15.620	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000
420	0.0000	0.0000	15.621	0.0000	0.0000	15.621	0.0000	0.0000	15.621	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000
430	15.621	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	16.143	1.0652	0.0000	0.0000	0.0000	0.0000
440	15.621	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	15.621	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	0.0000
450	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	0.0000
460	0.0000	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	15.621
470	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	15.621	0.0000	0.0000	15.621	0.0000
480	0.0000	0.0000	0.0000	0.0000	15.620	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	15.621
490	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	15.621
500	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	15.621	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	15.621	0.0000

Tabelle 4.12: Ergebnisse für eine variierende Länge des periodischen Teils beim Minimum in ms.

	10	20	30	40	50	60	70	80	90	100
10	0.12499	0.21867	0.34597	0.54674	0.73422	1.04662	1.37465	1.68764	2.04639	2.45255
20	0.20533	0.35927	0.49988	0.71858	0.96852	1.34343	1.66072	2.03077	2.45749	2.89245
30	0.34367	0.49988	0.70298	0.95532	1.24972	1.64024	2.06591	2.43691	2.92758	3.38983
40	0.51550	0.72088	0.96850	1.24283	1.57773	2.01519	2.48379	2.95876	3.45458	3.96782
50	0.70299	0.98544	1.23406	1.59337	1.98623	2.49941	2.98765	3.48356	4.09514	4.69266
60	0.98414	1.25102	1.59335	1.93706	2.47039	2.96805	3.56661	4.09467	4.73705	5.39169
70	1.28092	1.62692	1.95266	2.40568	2.97318	3.56991	4.17322	4.79798	5.50316	6.18974
80	1.62459	2.01655	2.45253	2.92250	3.51478	4.18778	4.86056	5.51722	6.22196	7.08114
90	2.03077	2.48510	2.92121	3.47644	4.15826	4.86061	5.64164	6.36020	7.19283	8.00223
100	2.50161	2.98506	3.51482	4.06642	4.84405	5.64169	6.46950	7.28680	8.17527	9.01581

Tabelle 4.13: Ergebnisse für eine variierende Länge des nicht periodischen Teils bei der Faltung in s.

	5	10	15	20	25	30	35	40	45	50
5	n.a.	0.18747	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
10	0.18568	0.07808	n.a.	0.21869	2.35882	0.39056	7.94274	0.92165	10.5855	1.04665
15	0.31242	1.36293	0.07810	6.14558	7.60298	0.24996	22.5850	26.8223	0.49986	49.7759
20	0.54811	0.15621	n.a.	0.07808	8.78188	0.79666	32.1190	0.26556	42.9571	3.32954
25	1.09349	4.25039	n.a.	18.6169	0.09374	44.9607	71.6232	83.3039	n.a.	0.20307
30	1.42154	0.37491	n.a.	1.37844	28.5522	0.10956	80.3204	6.35550	n.a.	9.75293
35	2.06203	6.50231	n.a.	30.1439	40.9609	n.a.	0.10934	230.349	n.a.	270.884
40	3.49918	2.78345	n.a.	1.92273	82.9056	n.a.	n.a.	1.79645	n.a.	24.2054
45	3.56311	10.7534	0.45304	50.6800	77.6138	n.a.	n.a.	401.781	0.09635	472.520
50	4.84490	1.23406	n.a.	n.a.	0.25129	n.a.	n.a.	n.a.	n.a.	0.10934

Tabelle 4.14: Ergebnisse für eine variierende Länge des periodischen Teils bei der Faltung in s.

	10	20	30	40	50	60	70	80	90	100
10	1.10913	2.59447	4.59405	8.89131	15.0653	20.2866	29.5553	40.8534	54.7456	53.8361
20	3.84525	2.95243	6.03241	12.1921	18.9863	27.5390	34.0048	48.1649	58.3523	59.0112
30	8.79839	7.62458	6.92392	13.5815	20.4730	29.9900	39.2386	51.8887	62.3053	64.8556
40	16.0314	15.3639	12.7193	13.3480	18.5355	31.7428	38.1511	50.3880	64.1975	65.3476
50	28.7223	27.3035	23.9410	31.6488	17.6115	29.6772	41.2398	52.7759	66.0433	67.2782
60	45.6031	42.2914	38.8986	51.9049	31.2584	25.8010	35.6169	52.2598	62.3578	69.7940
70	78.2042	68.0439	66.9859	87.9811	50.6378	48.4466	37.9214	52.1037	65.5009	71.9217
80	117.196	108.470	107.865	137.186	75.4981	76.4905	69.2977	55.0727	60.2200	68.9776
90	161.171	159.442	153.492	178.135	123.461	106.590	106.268	101.235	55.9513	68.5337
100	230.345	217.104	211.509	254.489	163.704	147.295	144.903	179.751	106.133	64.4097

Tabelle 4.15: Ergebnisse für eine variierende Länge des nicht periodischen Teils bei der Entfaltung in s .

	10	20	30	40	50	60	70	80	90	100
10	1.07789	2.34451	4.45346	7.15736	10.5471	14.8623	0.00000	0.00000	0.00000	0.00000
20	3.62416	1.78320	18.7079	4.60826	49.3613	9.36219	99.7507	16.6586	186.396	25.8181
30	8.18825	21.5657	2.64000	77.7481	126.632	7.15732	268.461	372.949	14.6766	630.538
40	14.9389	7.28339	86.9382	3.42366	245.753	41.1039	549.207	9.58833	1039.45	121.275
50	25.1567	71.5846	157.486	278.698	4.56377	681.460	965.137	1380.89	1834.89	11.2988
60	35.4568	17.3469	11.4085	49.4634	696.120	5.11314	1570.20	200.883	71.0128	327.550
70	50.0709	152.090	341.473	621.375	1053.14	1632.05	5.87644	3413.75	4579.02	6046.06
80	67.9807	33.2205	489.866	15.0796	1519.13	220.369	3429.54	6.84350	6606.28	686.000
90	90.7054	290.008	28.9257	1232.02	2049.80	84.3264	4796.80	6752.76	7.41581	12075.1
100	117.538	57.3983	873.090	177.464	20.3636	402.473	6399.16	765.362	12154.9	8.12586

Tabelle 4.16: Ergebnisse für eine variierende Länge des periodischen Teils bei der Entfaltung in s .

4.2.3 Zusammenfassung der Testergebnisse

Zusammenfassend ist zu den Testergebnissen zu sagen, dass sich eine große Auswirkung der verschiedenen Parameter nur bei der Faltung und der Entfaltung zeigen.

Bei der Addition und dem Minimum hatte die Anzahl an linearen Teilstücken zwar eine Auswirkung auf die Laufzeit, diese war aber selbst bei einer Anzahl von 500 linearen Teilstücken nicht sehr stark gewesen. Das ist damit zu begründen, dass der Rechenaufwand für die Addition und das Minimum linear ist und deutlich weniger Operationen und Berechnungen als bei der Faltung und Entfaltung gemacht werden müssen. Da in der Praxis eine Anzahl von 500 linearen Teilstücken kaum vorkommen wird, da ein aktueller Switch auf dem Markt, der time triggered gates unterstützt, nur eine Gatecontrollist mit Länge im niedrigen dreistelligen Bereich hat, kann dieser Effekt vernachlässigt werden. Darüber hinaus hat die Länge der stückweisen linearen Funktionen bei der Addition und dem Minimum keine nennenswerte Auswirkung auf die Laufzeit gehabt.

Bei der Faltung haben sowohl die Anzahl der linearen Teilstücke, als auch die Länge der stückweisen linearen Funktionen Auswirkungen auf die Laufzeit einer Berechnung gehabt. Vor allem die periodische Länge der stückweisen linearen Funktionen in Verbindung mit der Anzahl der periodischen linearen Teilstücke hatte in den Tests große Auswirkungen auf die Dauer der Berechnungen, während die Länge des nicht periodischen Teils der stückweisen linearen Funktionen keine große Auswirkung hatte. Dies ist damit zu erklären, dass für die Verlängerung der stückweisen linearen Funktionen bei der Faltung das kgV der periodischen Längen bestimmt wird und dieses für größere werdene Zahlen ebenfalls größer wird. So werden die stückweisen linearen Funktionen mehr verlängert, was wiederum mehr lineare Teilstücke mit sich bringt, welche die Laufzeit erhöhen.

Zur Anzahl der linearen Teilstücke kann man sagen, dass die Laufzeit sich ähnlich zu dem theoretischen Wachstum aus Proposition 3.3.3 verhält. Es hat sich auch gezeigt, dass es bei der Faltung für eine größere Anzahl an linearen Teilstücken im periodischen Teil und auch für eine größere Länge des periodischen Teils häufiger zu Fehlern kommt.

Bei der Entfaltung zeigten sich sehr ähnliche Auswirkungen der einzelnen Testparameter auf die Dauer der Berechnungen. Auch hier zeigte sich, für die Anzahl der periodischen Teilstücke und die Länge des periodischen Teils der stückweisen linearen Funktionen, eine große Auswirkung auf die Laufzeit. Auch hier kommt dies durch die Verlängerung der einzelnen stückweisen linearen Funktionen, welche wieder über das kgV der periodischen Länge bestimmt wird und so mit wachsender periodischer Länge für mehr lineare Teilstücke in den stückweisen linearen Funktionen sorgt.

Außerdem ist zu sagen, dass die Entfaltung deutlich stabiler ist als die Faltung und in dieser Versuchreihe keinen einzigen Fehler aufwies.

Abschließend ist zu sagen, dass während der Tests der Kern der CPU, welcher für die Tests zuständig war, die meiste Zeit voll ausgelastet war und maximal 1 GB RAM benötigt wurde.

5 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Ansatz zum Berechnen verschiedener Operationen der $(\min,+)$ Algebra auf stückweisen linearen Funktionen gezeigt und anschließend auch implementiert. Dafür wurde zunächst eine passende Definition für stückweise lineare Funktionen aufgestellt, die es uns im weiteren Verlauf ermöglicht hat, die nötigen Operationen der $(\min,+)$ Algebra zu berechnen. Anschließend wurde eine passende Datenstruktur, basierend auf dem Artikel von Bouillard und Thierry [BT08], gefunden, die leicht zu implementieren ist. Anschließend wurden in Kapitel 3.3 Algorithmen vorgestellt, die die Addition und Subtraktion, das Minimum und das Maximum, die Faltung und Entfaltung, sowie das Backlog und das Virtual Delay berechnen können. Mit Hilfe der Implementierung wurden in der Evaluation die Auswirkungen verschiedener Faktoren auf die Laufzeit der einzelnen Operationen getestet.

Ausblick

Für die Zukunft wäre es erstrebenswert, das in der Implementierung und der Evaluation angesprochene Problem mit den Gleitkommazahlen zu beheben. Da diese Problematik bei der Faltung und Entfaltung häufiger auftreten kann, würde eine Änderung für das Speichern der Länge eines linearen Teilstücks die Faltung und Entfaltung deutlich stabiler machen.

Außerdem könnte eine Verbesserung der Performanz durchaus noch möglich sein. Als mögliche Verbesserungen der Performanz könnten die Methoden zur Vorbereitung der Addition, der Subtraktion, dem Minimum und dem Maximum verbessert werden. Vor allem das Aufteilen der stückweisen linearen Funktionen, so dass alle linearen Teilstücke die selben Längen haben *get_lists_with_equal_intervalls()*, ist nicht sehr effizient und könnte durch eine iterative Methode ersetzt werden. Für leistungsstarke Systeme könnte auch die Parallelisierung verschiedener Prozesse hilfreich sein. Gerade in der Faltung ließen sich die einzelnen Faltungen der aufgeteilten Funktionen $f_1' * f_2'$, $f_1'' * f_2'$, $f_1' * f_2''$ und $f_1'' * f_2''$ sehr gut parallelisieren.

Als letzten Punkt ist anzumerken, dass in dieser Arbeit nur zeitvariante Berechnungen unterstützt werden. Eine zusätzliche Unterstützung von zeitinvarianten Berechnungen mit stückweisen linearen Funktionen [Fid10] wäre daher eine weitere Möglichkeit, die Bibliothek zu erweitern.

Anhang A

Beweis zu Lemma 3.1.1. Da sowohl f_1 als auch f_2 Spots sind, muss gelten, dass $s = b$ und $t - s = t - b = a \rightarrow t = a + b$, da $\forall s \neq b$ und $\forall t \neq a + b$, $f_1 = +\infty$ oder $f_2 = +\infty$. Daraus folgt, dass $f_1 * f_2(a + b) = \inf_{0 \leq s \leq a+b} (f_1(a + b - s) + f_2(s)) = f_1(a + b - b) + f_2(b) = f_1(a) + f_2(b)$ für $s = b$ und $+\infty$ sonst.

Beweis zu Lemma 3.1.2. Wie bei der Faltung zweier Spots, muss für f_2 gelten, dass $s = c$ ist, da $\forall s \neq c$, $f_2(s) = +\infty$ und damit $\inf_{0 \leq s \leq t} (f_1(t - s) + f_2(s)) = +\infty$. Da der Support von f_1 nur in $]a, b[$ liegt, muss außerdem gelten, dass $t - s = t - c \in]a, b[\rightarrow t \in]a + c, b + c[$. Daraus folgt $f_1 * f_2(t) = f_1(t - c) + f_2(c)$, $t \in]a + c, b + c[$.

Beweis zu Lemma 3.1.4. Da sowohl f_1 als auch f_2 Spots sind, muss gelten, dass $s = b$ und $t + s = t + b = a \rightarrow t = a - b$, da $\forall s \neq b$ und $\forall t \neq a - b$, $f_1 = -\infty$ oder $f_2 = -\infty$. Daraus folgt, dass $f_1 \oslash f_2(a - b) = \sup_{s \geq 0} (f_1(a - b + s) - f_2(s)) = f_1(a - b + b) - f_2(b) = f_1(a) - f_2(b)$ für $s = b$ und $-\infty$ sonst.

Beweis zu proposition 3.3.1. Die neuen Parameter d und T können in linearer Zeit in $\mathcal{O}(1)$ bestimmt werden. Das Verlängern der linearen Funktionen braucht maximal $\mathcal{O}(N_1 + N_2)$ Schritte, da sich der periodische Teil der linearen Funktionen um den Faktor $\frac{(T+d)-T_x}{d_x}$ verlängert. Das Errechnen der Addition von zwei linearen Teilstücken erfolgt in $\mathcal{O}(1)$ und muss $N_1 + N_2$ durchgeführt werden, woraus sich eine gesamt Komplexität von $\mathcal{O}(N_1 + N_2) + \mathcal{O}(N_1 + N_2) + \mathcal{O}(p_1 + p_2) = \mathcal{O}(N_1 + N_2)$ für die Berechnung der Addition ergibt.

Beweis zu proposition 3.3.2. Das Errechnen der neuen Parameter d und T braucht im schlechtesten Fall $\mathcal{O}(p_1 + p_2)$ Schritte, da für m_1 und m_2 $p_1 + p_2$ lineare Stücke überprüft werden müssen. Das Verlängern der linearen Funktionen braucht dann maximal $\mathcal{O}(N_1 + N_2)$ Schritte, da sich der periodische Teil der linearen Funktionen um den Faktor $\frac{(T+d)-T_x}{d_x}$ verlängert. Das Errechnen des Minimums von zwei linearen Teilstücken erfolgt in $\mathcal{O}(1)$ und muss $N_1 + N_2$ durchgeführt werden, woraus sich eine Gesamtkomplexität von $\mathcal{O}(N_1 + N_2) + \mathcal{O}(N_1 + N_2) + \mathcal{O}(p_1 + p_2) = \mathcal{O}(N_1 + N_2)$ für die Berechnung des Minimums ergibt.

Literaturverzeichnis

- [12] „IEEE Standard for Local and metropolitan area networks–Media Access Control (MAC) Bridges and Virtual Bridges“. In: *IEEE Std 802.1Q, 2012 Edition, (Incorporating IEEE Std 802.1Q-2011, IEEE Std 802.1Qbe-2011, IEEE Std 802.1Qbc-2011, IEEE Std 802.1Qbb-2011, IEEE Std 802.1Qaz-2011, IEEE Std 802.1Qbf-2011, IEEE Std 802.1Qbg-2012, IEEE Std 802.1aq-2012, IEEE Std 802.1Q-2012* (Dez. 2012), S. 1–1782. DOI: [10.1109/IEEESTD.2012.6606799](https://doi.org/10.1109/IEEESTD.2012.6606799) (zitiert auf S. 16).
- [16] „IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic“. In: *IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q-2014/Cor 1-2015)* (März 2016), S. 1–57. DOI: [10.1109/IEEESTD.2016.7440741](https://doi.org/10.1109/IEEESTD.2016.7440741) (zitiert auf S. 16).
- [18] *Developer Survey Results 2018*. <https://insights.stackoverflow.com/survey/2018/>. 2018. URL: <https://insights.stackoverflow.com/survey/2018/> (zitiert auf S. 17).
- [BCG+09] A. Bouillard, B. Cottenceau, B. Gaujal, L. Hardouin, S. Lagrange, M. Lhommeau, E. Thierry. „COINC Library : A toolbox for Network Calculus“. In: *Fourth International Conference on Performance Evaluation Methodologies and Tools, Valuetools*. Pisa, Italy, 2009. URL: <https://hal.inria.fr/hal-00788929> (zitiert auf S. 16, 17).
- [BS14] S. Bondorf, J. B. Schmitt. „The DiscoDNC v2 - A Comprehensive Tool for Deterministic Network Calculus“. In: *VALUETOOLS*. 2014 (zitiert auf S. 16).
- [BT08] A. Bouillard, É. Thierry. „An Algorithmic Toolbox for Network Calculus“. In: *Discrete Event Dynamic Systems* 18.1 (2008), S. 3–49. DOI: <https://doi.org/10.1007/s10626-007-0028-x> (zitiert auf S. 3, 13, 15, 19, 21, 22, 25, 26, 29, 30, 32, 34–37, 65).
- [Car07] J. Carette. „A canonical form for some piecewise defined functions“. In: *CoRR* abs/cs/0702010 (2007). arXiv: [cs/0702010](https://arxiv.org/abs/cs/0702010). URL: <http://arxiv.org/abs/cs/0702010> (zitiert auf S. 19, 20).
- [Fid10] M. Fidler. „Survey of deterministic and stochastic service curve models in the network calculus“. In: *IEEE Communications Surveys Tutorials* 12.1 (First 2010), S. 59–86. ISSN: 1553-877X (zitiert auf S. 16, 65).
- [Gol91] D. Goldberg. „What Every Computer Scientist Should Know About Floating-point Arithmetic“. In: *ACM Comput. Surv.* 23.1 (März 1991), S. 5–48. ISSN: 0360-0300. DOI: [10.1145/103162.103163](https://doi.org/10.1145/103162.103163). URL: <http://doi.acm.org/10.1145/103162.103163> (zitiert auf S. 41).
- [LT01] J.-Y. Le Boudec, P. Thiran. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Berlin, Heidelberg: Springer-Verlag, 2001. ISBN: 3-540-42184-X (zitiert auf S. 13, 15, 21, 41).

- [MOH98] M. V. MOHRENSCHILDT. „A Normal Form for Function Rings of Piecewise Functions“. In: *Journal of Symbolic Computation* 26.5 (1998), S. 607–619. ISSN: 0747-7171. DOI: <https://doi.org/10.1006/jSCO.1998.0229>. URL: <http://www.sciencedirect.com/science/article/pii/S0747717198902292> (zitiert auf S. 19).
- [SSH07] H. Schioler, H. P. Schwefel, M. B. Hansen. „CyNC: A MATLAB/SimuLink Toolbox for Network Calculus“. In: *Proceedings of the 2Nd International Conference on Performance Evaluation Methodologies and Tools*. ValueTools '07. Nantes, France: ICST (Institute for Computer Sciences, Social-Informatics und Telecommunications Engineering), 2007, 60:1–60:10. ISBN: 978-963-9799-00-4. URL: <http://dl.acm.org/citation.cfm?id=1345263.1345340> (zitiert auf S. 16, 17).
- [WT06] E. Wandeler, L. Thiele. *Real-Time Calculus (RTC) Toolbox*. <http://www.mpa.ethz.ch/Rtctoolbox>. 2006. URL: <http://www.mpa.ethz.ch/Rtctoolbox> (zitiert auf S. 16, 17).

Alle URLs wurden zuletzt am 11. 06. 2018 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift