

Das Ein-Maschinen-Durchlaufzeitproblem auf ($q, q-3$)-Partialordnungen

Inaugural-Dissertation
zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Universität zu Köln

vorgelegt von
Vanessa A. Stausberg
aus Leverkusen

Köln, 2001

Berichterstatter: Prof. Dr. Rainer Schrader
Prof. Dr. Ulrich Faigle

Tag der mündlichen Prüfung: 6.12.2001

Inhaltsverzeichnis

1	Einleitung	1
2	Definitionen und Grundlagen	3
2.1	Grundbegriffe	3
2.2	p -zusammenhängende Graphen	4
2.3	Partialordnungen	6
2.4	Das Durchlaufzeitproblem	15
3	($q, q-3$)-Graphen	25
3.1	($q, q-4$)-Graphen	26
3.2	($q, q-3$)-Graphen	29
4	($q, q-3$)-Partialordnungen	35
4.1	Spinnen	37
4.2	Kreise und deren Komplemente	38
4.3	Pfade	39
4.4	Spinnennetze	40
4.5	Graphen beschränkter Grösse	45
5	$1 \mid (q, q-3) \mid \sum w_j C_j$	51
5.1	Pfadähnliche ($q, q-3$)-Posets	53
5.2	Spinnennetze	59
6	Zusammenfassung und Ausblick	69
	Literatur	72
	Index	79
	English Summary	83

Für die stete Ansprechbarkeit und sehr gute Zusammenarbeit bedanke ich mich herzlich bei Professor Dr. Rainer Schrader.

Kapitel 1

Einleitung

Das Ein-Maschinen-Durchlaufzeitproblem sucht eine optimale Zuweisung von Aktivitäten auf eine Maschine, die die Summe der gewichteten Durchlaufzeiten, d.h die Zeitpunkte, zu denen die einzelnen Aktivitäten beendet sind, minimiert. Aktivitäten können Arbeitsschritte an einem Werkstück, Instruktionen eines Computerprogramms usw. sein. Das Problem tritt in verschiedensten Formen immer dann in der Praxis auf, wenn möglichst viele Aktivitäten zu einem frühen Zeitpunkt beendet sein sollen.

In dieser Arbeit wird das Ein-Maschinen-Durchlaufzeitproblem mit Reihenfolgerestriktionen, die durch einen transitiv orientierbaren $(q, q-3)$ -Graphen, $q \geq 7$, gegeben sind, untersucht.

Allgemein heißt ein Graph (q, t) -Graph, wenn q Knoten maximal t sehnenlose Wege der Länge drei, kurz P_4 , induzieren. Sie wurden von Babel in seiner Habilitationsschrift „On the P_4 -Structure of Graphs” [Bab97] eingeführt.

Viele bekannte Graphenklasse, für die nicht nur das Durchlaufzeitproblem effizient lösbar ist, gehören zu den $(q, q-4)$ -Graphen. *Cographs* sind definiert als Graphen, die keinen induzierten P_4 besitzen, sind also $(4, 0)$ -Graphen. P_4 -dünne Graphen besitzen per Definition auf fünf Knoten maximal einen P_4 , entsprechen damit $(5, 1)$ -Graphen. Es wird sich herausstellen, daß P_4 -erweiterbare Graphen, eine echte Obermenge der P_4 -dünnen Graphen, zusammenfallen mit $(6, 2)$ -Graphen, wenn sie keinen induzierten Kreis der Länge fünf besitzen.

$(q, q-3)$ -Graphen sind eine echte Obermenge der $(q, q-4)$ -Graphen, wie ein sehnenloser Weg der Länge k , $k \geq 5$, belegt. Die Charakterisierung von $(q, q-3)$ -Graphen basiert auf dem 1995 von B. Jamison und S. Olariu eingeführten Konzept p -zusammenhängender Graphen. Ein Graph ist p -

zusammenhängend, wenn für jede Partitionierung der Knoten in zwei nicht-leere Mengen ein P_4 existiert, der Knoten aus beiden Mengen besitzt. Die Einschränkung der $(q, q - 3)$ -Graphen auf transitiv orientierbare, d.h. Komparabilitätsgraphen, führt zur Klasse der $(q, q - 3)$ -*Partialordnungen*.

Das Ein-Maschinen-Durchlaufzeitproblem mit beliebigen Reihenfolgerestriktionen ist NP-schwer [Law78]. Das Problem ist polynomiell lösbar für serienparallele [Law78], N-dünne [vASW96] und N-erweiterbare Partialordnungen [PW97], allgemein solche mit beschränkter Dekompositionsweite (bzw. beschränktem Dekompositionsdurchmesser) [SS86].

Motiviert ist die Arbeit zum einen durch die Tatsache, daß $(q, q - 3)$ -Graphen eine handliche Charakterisierung besitzen. Zudem verallgemeinern $(q, q - 3)$ -Partialordnungen serienparallele, N-dünne und N-erweiterbare Partialordnungen. Im Unterschied zur Verallgemeinerung durch Partialordnungen mit beschränkter Dekompositionsweite gehören auch einige Partialordnungen mit unbeschränkter Dekompositionsweite zur Klasse der $(q, q - 3)$ -Partialordnungen. Damit impliziert die polynomielle Lösbarkeit des Ein-Maschinen-Durchlaufzeitproblems über $(q, q - 3)$ -Partialordnungen die positive Beantwortung der offenen Frage nach der Existenz effizient lösbarer unbeschränkter Partialordnungen.

Die Arbeit gliedert sich wie folgt.

In Kapitel 2 führen wir die zum Verständnis notwendigen Begriffe zur Charakterisierung p -zusammenhängender Graphen ein und stellen die notwendigen Begriffe aus der Theorie der Partialordnungen dar. Desweiteren geben wir einen Überblick über die wichtigsten Ergebnisse zum Ein-Maschinen-Durchlaufzeitproblem.

In Kapitel 3 stellen wir $(q, q - 3)$ -Graphen und die Subklasse der $(q, q - 4)$ -Graphen vor.

Da nicht jeder $(q, q - 3)$ -Graph transitiv orientierbar ist, prüfen wir in Kapitel 4 zunächst alle $(q, q - 3)$ -Graphen auf diese Eigenschaft. Wir schränken die Klasse ein und geben eine Konstruktionsvorschrift zum sukzessiven Aufbau von $(q, q - 3)$ -Partialordnungen an.

In Kapitel 5 entwickeln wir zwei neue Algorithmen, die das Durchlaufzeitproblem für zwei Teilklassen der $(q, q - 3)$ -Vorgängerrelationen effizient lösen. Leichte Abänderungen dieser beiden Algorithmen ermöglichen dann die effiziente Lösbarkeit aller $(q, q - 3)$ -Partialordnungen.

Abschließend folgen Zusammenfassung und Ausblick.

Kapitel 2

Definitionen und Grundlagen

In diesem Kapitel werden in kompakter Form die für das weitere Verständnis notwendigen Grundlagen zusammengestellt. Diese umfassen das Konzept der p -zusammenhängenden Graphen, Partialordnungen und einen Überblick über die bekannten Ergebnisse zum Durchlaufzeitproblem. Im übrigen wird verwiesen auf die einschlägige Literatur: Möhring ([Möh85],[Möh89]) und Rival ([Riv82],[Riv85]) zur Ordnungstheorie; Brucker [Bru95], Pinedo [Pin95], Parker [Par95] und Karger et al. [KSW98] zur Einführung in die Reihenfolgeplanung.

2.1 Grundbegriffe

Alle Graphen in dieser Arbeit sind endlich und einfach. Zusätzlich zu graphentheoretischen Standardnotationen, wie etwa in [Die96] oder [BM76], führen wir einige weitere Definitionen ein.

Sei $G = (V, E)$ ein Graph mit Knotenmenge V und Kantenmenge E . Wir geben explizit an, ob der Graph ungerichtet oder gerichtet ist und schreiben etwas abweichend von den klassischen Vorlagen für eine ungerichtete Kante $e = xy$ und $e = (x, y)$ für eine gerichtete Kante von x nach y . Mit n wird die Kardinalität von V , $n := |V|$, bezeichnet. Ist v ein Knoten von G , dann ist $N(v)$ die Menge aller Knoten, die adjazent zu v sind. $u \in N(v)$ heißt auch *Nachbar* von v . Mit $\overline{N}(v)$ bezeichnen wir die Menge aller Nicht-Nachbarn von v . Zu $U \subseteq V$ ist $G(U)$ der durch U induzierte Teilgraph. Man spricht auch von *Untergraph*. Statt $G(V \setminus \{v\})$ schreiben wir $G \setminus v$. Eine *Clique* ist eine Menge paarweise benachbarter Knoten. Eine Knotenteilmenge heißt *stabil*, falls je zwei Knoten nicht adjazent sind. Eine Knotenteilmenge $U \subseteq V$ mit $1 < |U| < |V|$ heißt *homogen*, wenn jeder Knoten $v \in V \setminus U$ entweder zu

allen $u \in U$ adjazent oder zu allen $u \in U$ nicht adjazent ist. Eine homogene Menge U ist maximal, wenn keine andere homogene Menge U' die Menge U echt enthält.

Der Graph, der entsteht, wenn jede maximale homogene Menge eines Graphen G zu einem einzigen Knoten geschrumpft wird, heißt *charakteristischer Graph* von G .

Zwei Graphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ heißen *isomorph*, wenn eine Bijektion $\varphi : V_1 \rightarrow V_2$ existiert, so daß $xy \in E_1 \Leftrightarrow \varphi(x)\varphi(y) \in E_2$ für alle $x, y \in V_1$ gilt.

Das Komplement \overline{G} eines Graphen $G = (V, E)$ ist der Graph auf V , in dem zwei Knoten genau dann benachbart sind, wenn sie es in G nicht sind.

Ein Graph G heißt *cozusammenhängend*, wenn sein Komplement \overline{G} zusammenhängt. Analog heißen die Zusammenhangskomponenten von \overline{G} *Cozusammenhangskomponenten* von G .

Wie üblich ist P_n der sehnenslose Weg auf n Knoten mit $n - 1$ Kanten. Wir sprechen auch von *Pfad*. Die *Länge* eines P_n ist $n - 1$. In einem P_4 mit Knoten v_1, v_2, v_3, v_4 und Kanten v_1v_2, v_2v_3 und v_3v_4 heißen v_1 und v_4 *Endknoten* und v_2 und v_3 *Mittelknoten*. Ein Knoten heißt *p-Endknoten* eines Graphen G , wenn er genau zu einem P_4 in G gehört. Ein Graph ohne induzierten P_4 heißt *Cograph* (Abkürzung für: complement reducible graph).

Ein C_n ist ein sehnensloser Kreis mit n Knoten und n Kanten.

2.2 p -zusammenhängende Graphen

Die Ergebnisse dieses Abschnitts basieren auf dem Konzept p -zusammenhängender Graphen, das B. Jamison und S. Olariu 1995 eingeführt haben. Wir übernehmen die Terminologie und stellen die für uns relevanten Ergebnisse aus [JO95b] vor.

Ein Graph $G = (V, E)$ heißt *p-zusammenhängend*, wenn für jede Partition von V in zwei nicht-leere disjunkte Mengen V_1 und V_2 ein P_4 existiert, der sowohl Knoten in V_1 als auch in V_2 besitzt. Ein solcher P_4 heißt *P_4 -Übergang* zwischen V_1 und V_2 . Die *p-zusammenhängenden Komponenten* eines Graphen sind die maximal induzierten Teilgraphen, die p -zusammenhängend sind.

p -zusammenhängende Komponenten besitzen entweder einen oder mindestens vier Knoten. Sie sind unter Komplementbildung abgeschlossen und sowohl in G also auch in \overline{G} zusammenhängende Teilgraphen. Ein Endknoten

in G ist ein Mittelknoten in \overline{G} und umgekehrt. Zudem besitzt jeder Graph eine eindeutige Zerlegung in p -zusammenhängende Komponenten.

Ein p -zusammenhängender Graph heißt *separabel* (oder *separierbar*), wenn seine Knotenmenge in zwei nicht-leere, disjunkte Knotenmengen V_1 und V_2 aufgeteilt werden kann, so daß jeder P_4 -Übergang zwischen V_1 und V_2 seine Mittelknoten in V_1 und seine Endpunkte in V_2 hat. (V_1, V_2) heißt *Separation* von G .

Offensichtlich ist das Komplement eines separierbaren p -zusammenhängenden Graphen auch separierbar mit der Separierung (V_2, V_1) in \overline{G} . Figur 2.1 zeigt die Separation (V_1, V_2) von G .

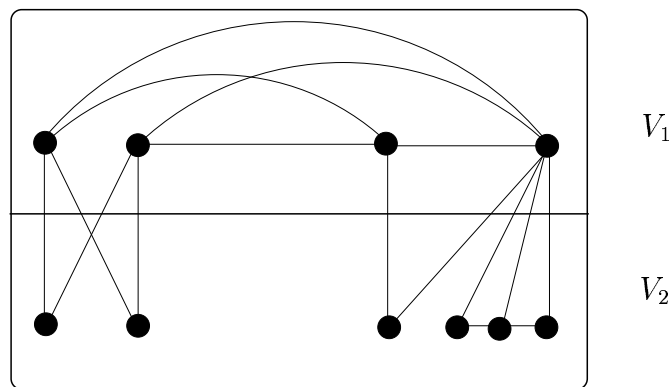


Abbildung 2.1: Separation von G

Die Einführung und das Studium separierbarer p -zusammenhängender Graphen liegt begründet im folgenden Strukturtheorem für beliebige Graphen.

Theorem 2.1 (Strukturtheorem, JO95, Theorem 3) Sei $G=(V,E)$ ein Graph. Dann gilt genau eine der folgenden Bedingungen:

- (i) G ist nicht zusammenhängend.
- (ii) \overline{G} ist nicht zusammenhängend.
- (iii) Es existiert eine eindeutige echte separable p -zusammenhängende Komponente H mit Separation (H_1, H_2) , so daß jeder Knoten außerhalb von H adjazent zu allen Knoten in H_1 ist und zu keinem Knoten in H_2 .
- (iv) G ist p -zusammenhängend.

Das Strukturtheorem impliziert in natürlicher Weise die Darstellung eines beliebigen Graphen durch einen *Wurzelbaum* (d.h. einen Baum mit einem ausgezeichneten Knoten, der Wurzel). Die Wurzel entspricht dem Graphen G , die inneren Knoten repräsentieren die ersten drei Fälle und die Blätter entsprechen den p -zusammenhängenden Komponenten und einzelnen Knoten.

Die Darstellung ist bis auf Isomorphien eindeutig.

Aus [JO95] ist bekannt, daß die Repräsentation eines beliebigen Graphen durch einen solchen Baum in polynomieller Zeit in der Anzahl der Knoten bestimmt werden kann. [Bau96] gibt einen Algorithmus an, der lineare Laufzeit in der Grösse des Graphen $O(|V| + |E|)$ benötigt.

2.3 Partialordnungen

Sei $V = \{v_1, v_2, \dots, v_n\}$ eine endliche Menge. Eine *Partialordnung* auf V ist eine irreflexive, transitive Relation $<$ auf $V \times V$, d.h.

$$\begin{aligned} \forall v \in V : & \quad v \not< v, \\ \forall v, w, x \in V : & \quad v < w \text{ und } w < x \Rightarrow v < x. \end{aligned}$$

Es ist leicht einzusehen, daß eine Partialordnung auch antisymmetrisch ist, d.h. $v < w$ impliziert $w \not< v$ für alle $v, w \in V$.

Eine *partiell geordnete Menge* (engl: *partially ordered set*, kurz: *poset*) $P = (V, <_P)$ besteht aus einer endlichen Menge V und einer Partialordnung $<_P$ auf V . Wir schreiben $<$ statt $<_P$, falls Irrtümer ausgeschlossen sind.

P bezeichnet fortan sowohl die Grundmenge V als auch die Partialordnung $(V, <)$. Die Begriffe partiell geordnete Menge, Partialordnung und Poset benutzen wir synonym.

Zwei Elemente u und v einer Partialordnung heißen *vergleichbar*, wenn $u < v$ oder $v < u$ gilt. Sind die beiden Elemente nicht vergleichbar, heißen sie *parallel*, kurz: $u \parallel v$. Eine Menge paarweiser vergleichbarer Elemente heißt *Kette*. Ein Poset, das aus einer einzigen Kette besteht, heißt *Totalordnung*. Eine Menge paarweiser paralleler Elemente heißt *Antikette*. Die *Weite* einer Partialordnung ist die Anzahl der Elemente in einer größten Antikette.

Betrachtet man die Elemente einer Partialordnung als Knoten eines Graphen und ergänzt für je zwei vergleichbare Elemente eine gerichtete Kante, erhält man einen gerichteten, azyklischen und transitiv abgeschlossenen (d.h. für

jedes Kantenpaar $(u, v), (v, w)$ gehört auch (u, w) zur Kantenmenge) Graphen G . Umgekehrt definiert offensichtlich jeder gerichtete azyklische Graph $D = (V, A)$ (engl.: directed acyclic graph, kurz: DAG) durch

$$u < v \Leftrightarrow \exists \text{ ein gerichteter Weg von } u \text{ nach } v \text{ in } G$$

eine Partialordnung P auf der Knotenmenge V .

Der zugrundeliegende ungerichtete Graph $G(P)$ von P heißt *Komparabilitätsgraph* von P . Ist andererseits G ein Komparabilitätsgraph, dann heißt jede Partialordnung P mit $G = G(P)$ *assoziiert* mit G . Jede mit G assoziierte Partialordnung entspricht einer *transitiven Orientierung* der Kanten von G und umgekehrt [Gol80].

Ob ein Graph $G = (V, E)$ Komparabilitätsgraph ist, kann in $O(|V| + |E|)$ Zeit getestet werden [MS99]. Dies ist eine Verfeinerung von Möhrings Algorithmus [Möh85], dessen Verfahren, das eine Laufzeit von $O(|V| \cdot |E|)$ hat, wir hier erläutern. Die Grundidee besteht darin, sogenannte Λ^* -Klassen der Kantenmenge des Graphen zu identifizieren und diese unabhängig voneinander auf die Existenz transitiver Orientierungen zu testen.

Ein Λ in einem ungerichteten Graphen bezeichnet zwei Kanten, die einen induzierten $P_3 = v_1 v_2 v_3$ bilden (siehe Abbildung 2.2). Man sagt auch: die Kanten $v_1 v_2$ und $v_2 v_3$ *stehen in Λ -Relation* zueinander.

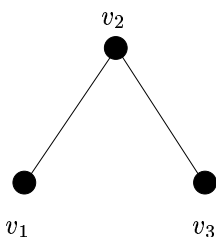


Abbildung 2.2: Λ

Man kann sich leicht überlegen, daß die Orientierung einer der beiden Kanten, z.B. (v_1, v_2) , die transitive Orientierung der anderen Kante *forcirt*, nämlich (v_3, v_2) , da sonst die Transitivität zerstört ist. Der *transitive Abschluß* Λ^* eines Λ ist die größtmögliche Kantenmenge, so daß keine weitere Kante im Graphen existiert, die mit einer Kante $e \in \Lambda^*$ ein Λ bildet. Der transitive Abschluß bildet eine Äquivalenzrelation auf den Kanten. Die Äquivalenzklassen heißen Λ^* -Klassen. Demzufolge zerfällt ein Graph vollständig in

Λ^* -Klassen und einzelne Kanten (die Kanten eines Kreises der Länge drei zerfallen in einzelne Kanten). In einer Λ^* -Klasse forciert nun die Orientierung einer einzelnen Kante e die Orientierung der gesamten Klasse (siehe Beispiel in Abbildung 2.3). Da e in zwei Richtungen orientiert werden kann, besitzt eine Λ^* -Klasse entweder genau zwei transitive Orientierungen (die eine ist die entgegengesetzte der anderen) oder gar keine. Letzteres tritt ein, wenn alle Kanten in beide Richtungen orientiert werden müssen und damit Kreise erzeugen (siehe Beispiel in Abbildung 2.4). Für Details siehe [Gol80].

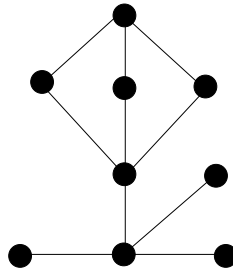


Abbildung 2.3: Komparabilitätsgraph mit einer Λ^* Klasse

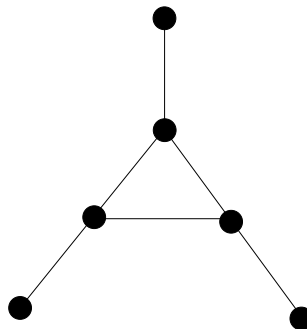


Abbildung 2.4: Graph ohne transitive Orientierung

Lemma 2.2 (Gol80, Korollar 5.5) *Besitzt jede Λ^* -Klasse eines Graphen G genau zwei transitive Orientierungen, dann ist G Komparabilitätsgraph.*

Wir weisen daraufhin, daß zur transitiven Orientierung des gesamten Graphen die Orientierungen der einzelnen Λ^* -Klassen nicht beliebig gewählt werden dürfen. Das klassische Gegenbeispiel ist der Kreis der Länge drei C_3 , dessen drei Λ^* -Klassen aus je einer Kante bestehen. Von den $8 = 2^3$ möglichen Orientierungen sind jedoch die beiden gerichteten Kreise nicht transitiv.

Besitzt ein Komparabilitätsgraph $G(P)$ genau eine Λ^* -Klasse bezeichnen wir die beiden assoziierten Partialordnungen mit P und P^{-1} .

Wir kommen nun zur gängigsten Möglichkeit, eine Partialordnung graphisch darzustellen.

Ein Element v einer Partialordnung P heißt *direkter Nachfolger* von u , wenn $u < v$ ist und kein Element w existiert mit $u < w < v$, kurz $u \dot{<} v$. Man sagt auch v *überdeckt* u . Analog ist der *direkte Vorgänger* definiert. Zeichnet man einen gerichteten Graphen, dessen Knoten die Elemente der Partialordnung P darstellen und dessen Kantenmengen aus allen Überdeckungen besteht, so erhält man den *Überdeckungsgraphen*. Zeichnet man den Überdeckungsgraphen so in die Ebene, daß alle Kanten in eine Richtung, im allgemeinen nach oben zeigen, dann nennt man den zugrundeliegende ungerichteten Graphen *Hasse-Diagramm* $H(P)$ von P .

Eine Teilmenge $I \subseteq P$ heißt *Ideal*, wenn sie bezüglich der Ordnungsrelation nach unten abgeschlossen ist, d.h. $v \in I$ und $u < v$ impliziert $u \in I$. Ein Ideal heißt *echt*, wenn $I \neq P$. Mit $I(w) := \{v \in P \mid v < w\}$ bezeichnen wir das von w erzeugte *Hauptideal*. Analog heißt eine Teilmenge $F \subseteq P$ *Filter*, wenn sie bezüglich der Ordnungsrelation nach oben abgeschlossen ist.

Eine *konvexe* Menge $S \subseteq P$ ist eine Teilmenge, die zu je zwei Elementen u und v auch alle Elemente beinhaltet, die zwischen u und v liegen, d.h. $\forall w \in P$ mit $u < w < v \Rightarrow w \in S$. $Min(P)$ (bzw. $Max(P)$) bezeichnet die Menge der minimalen (maximalen) Elemente in P .

Eine *lineare Erweiterung* $L = v_{\pi(1)}v_{\pi(2)}\dots v_{\pi(n)}$ einer Partialordnung $P = (V, <_P)$ mit n Elementen ist eine Permutation auf V , so daß $v <_P w$ impliziert, daß v in der Permutation Vorgänger von w ist. Man sagt auch, daß die durch die lineare Erweiterung definierte Totalordnung *kompatibel* mit P oder eine Permutation *zulässig* für P ist. $\mathcal{L}(P)$ bezeichnet die Menge aller linearen Erweiterungen von P .

Wir geben nun einige Definitionen und Tatsachen über die *modulare Dekomposition* von Partialordnungen (auch *Substitutions-Dekomposition* oder *Lexikographische Dekomposition* genannt). Ein ausführlicher Überblick findet sich in Möhring und Radermacher [MR84] und Möhring [Möh89].

Sei $Q = (V, <_Q)$ mit $V = \{v_1, v_2, \dots, v_n\}$ eine Partialordnung mit n Elementen und seien $P_i = (V_i, <_i)$ für $i = 1, \dots, n$ n disjunkte Partialordnungen. Dann bezeichnet $P = Q[P_1, P_2, \dots, P_n]$ die Partialordnung $P = (V, <_P)$, die entsteht, in dem man die Elemente v_i aus Q durch die Partialordnungen P_i

für $i = 1, \dots, n$ ersetzt. Also ist $\mathbb{V} = \bigcup_{i=1}^n V_i$ und

$$u <_P v \Leftrightarrow \begin{cases} \text{entweder} & u, v \in V_i & \text{und } u <_i v & \text{für ein } i \\ \text{oder} & u \in V_i, v \in V_j & \text{und } v_i <_Q v_j & \text{für } i \neq j. \end{cases}$$

Die Komposition ist *echt*, wenn $1 < |V_i| < |V|$ für ein i gilt und die Partialordnung heißt *zerlegbar*, sonst *unzerlegbar*, *irreduzibel* oder *prim*.

Satz 2.3 (Möh89, Theorem 4.2) *Für jede zerlegbare Partialordnung trifft genau einer der drei folgenden Fälle zu:*

- (1) $P = Q[P_1, \dots, P_n]$, wobei Q eine Antikette ist. P heißt dann *Parallelkomposition* von P_1, \dots, P_n . Wir schreiben auch $P = P_1 \parallel P_2 \parallel \dots \parallel P_n$.
- (2) $P = Q[P_1, \dots, P_n]$, wobei Q eine Kette ist. P heißt dann *Serienkomposition* von P_1, \dots, P_n . Wir schreiben auch $P = P_1 \rightarrow P_2 \rightarrow \dots \rightarrow P_n$.
- (3) $P = Q[P_1, \dots, P_n]$, wobei Q prim ist. Wir sprechen dann von einer *Prim-Komposition*.

Satz 2.4 (Gol80) *Eine unzerlegbare Partialordnung besitzt genau zwei transitive Orientierungen, von denen eine die Umkehrung der anderen ist.*

Eine nicht-leere Teilmenge $M \subseteq V$ einer Partialordnung $P = (V, <)$ heißt *Modul*, wenn sich jedes Element außerhalb des Moduls gleich zu allen Elementen innerhalb des Moduls verhält. D.h. M ist ein Modul, wenn für jedes Element $v \in V \setminus M$ entweder

- (a) $u < v$ für alle $u \in M$ oder
- (b) $v < u$ für alle $u \in M$ oder
- (c) $u \parallel v$ für alle $u \in M$ gilt.

In einer Komposition $P = Q[P_1, \dots, P_n]$ heißt Q *äußerer Faktor*, P_1, \dots, P_n sind *innere Faktoren*. Es ist leicht zu sehen, daß jeder innere Faktor P_i ein Modul von P ist.

Die größte Anzahl an Elementen (bzw. die Weite) eines Faktors heißt *De-kompositionsdurchmesser* (bzw. *-weite*) von P .

Durch Satz 2.3 ist jede Partialordnung darstellbar durch einen *Kompositionsbau* $T(P)$. Die inneren Knoten von $T(P)$ sind parallel (\parallel), seriell (\rightarrow)

oder als prim markiert und geben die Kompositionsart für die Söhne, d.h. den inneren Faktoren, an. Die Blätter entsprechen den einzelnen Elementen aus P . Der Kompositionsbaum ist eindeutig, wenn (1) und (2) so groß wie möglich gefordert wird.

Der Kompositionsbaum kann berechnet werden durch das Identifizieren der Zusammen- und Cozusammenhangskomponenten des Komparabilitätsgraphen $G(P)$ von P . Ist der Graph zusammenhängend, aber nicht cozusammenhängend, erzeugt man einen seriellen inneren Knoten, umkehrt einen parallelen. Ist der Graph sowohl zusammenhängend als auch cozusammenhängend, identifiziert man die maximalen Teilmodule (d.h. Module, die in keinem anderen Modul echt enthalten sind) und fügt diese als Söhne einer Prim-Komposition in den Baum ein. Dieses Vorgehen heißt auch *modulare Dekomposition*.

Dank eines viel allgemeineren Ergebnisses von Dahlhaus, Gustedt and McConnell [DGM96] und unabhängig Baumann [Bau96] kann der Kompositionsbaum in linearer Zeit in der Größe des Graphen berechnet werden.

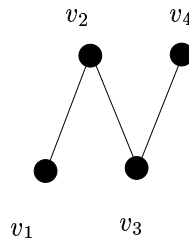
Die modulare Dekomposition entspricht auf dem (ungerichteten) zugrundeliegenden Komparabilitätsgraphen der Zerlegung in homogene Mengen. Die heißt dementsprechend *homogene Dekomposition* und ist eine Verfeinerung der Dekomposition, die durch das Strukturtheorem 2.1 vorgegeben ist. Vereinfacht gesprochen werden in der homogenen Dekomposition zusätzlich homogene Mengen durch einen einzigen Knoten ersetzt.

Wir stellen einige spezielle Partialordnungen vor.

Eine Partialordnung P heißt *serien-parallel*, wenn P aus einelementigen Partialordnungen nur durch Serien- und Parallelkompositionen konstruierbar ist. Weitere Charakterisierungen sind:

Satz 2.5 (HM87, Theorem 2.1) *Für eine serien-parallele Partialordnung P sind folgende Eigenschaften äquivalent:*

- (i) *Keine vier Elemente induzieren ein N , d.h es gibt keine vier Elemente v_1, v_2, v_3, v_4 in P zwischen denen nur die Relationen $v_1 < v_2, v_2 > v_3$ und $v_3 < v_4$ bestehen (siehe Figure 2.5).*
- (ii) *Der Komparabilitätsgraph $G(P)$ ist Cograph.*
- (iii) *Der Kompositionsbaum enthält nur seriell oder parallel markierte Knoten.*

Abbildung 2.5: Hasse-Diagramm der Ordnung N

Spezielle Unterklassen serien-paralleler Ordnungen sind beispielsweise: *Intree-Ordnungen* (jedes Element mit Ausnahme der Wurzel besitzt genau einen Nachfolger), *Outtree-Ordnungen* (jedes Element mit Ausnahme der Wurzel besitzt genau einen Vorgänger), *schwache Ordnungen* (die Serienkomposition einer Menge von Antiketten) und *Threshold-Ordnungen* (die Elemente können so gewichtet werden, daß eine Teilmenge der Elemente genau dann eine Antikette ist, wenn ihr Gewicht einen vorgegebenen Schwellenwert nicht überschreitet).

Partialordnungen, deren Hasse-Diagramm kein induziertes N enthält, heißen *N -frei*. Natürlich sind serien-parallele Ordnungen N -frei. Andersherum gilt dies nicht: Figur 2.6 zeigt das Hasse-Diagramm einer N -freien, aber nicht serien-parallelen Ordnung. N -freie Ordnungen können nach Faigle und Gierz [FG84] (vgl. auch Habib und Jegou [HJ85]) durch folgende zusätzlich zur Serien- und Parallelkomposition hinzukommende Komposition rekursiv konstruiert werden.

Sei P_1 eine N -freie Ordnung, P_2 eine zu P_1 disjunkte Antikette und $A \subseteq \text{Max}(P_1)$. Dann ist die *quasi-serielle Komposition* $P_1 \oplus P_2$ definiert durch

$$u < v : \Leftrightarrow \begin{cases} u, v \in P_1, u <_{P_1} v, \\ u \in I(A), v \in P_2. \end{cases}$$

Diese Komposition ist nicht eindeutig. Der Spezialfall $A = \text{Max}(P_1)$ entspricht der Serienkomposition, wobei P_2 keine Antikette sein muß.

Zudem sind N -freie Ordnungen genau die Partialordnungen, die *Ketten-Antiketten-vollständig* (engl.: chain-antichain complete, kurz: CAC) sind, d.h. jede maximale Kette schneidet jede maximale Antikette.

Ein Graph heißt *P_4 -erweiterbar*, wenn jede Knotenteilmenge $W \subseteq V$, die einen P_4 induziert, eine echte Erweiterung besitzt, d.h. es existiert höchstens

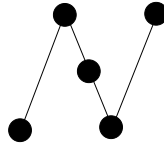


Abbildung 2.6: N -frei und nicht serien-parallel

ein Knoten $v \in V \setminus W$ der einen weiteren P_4 mit drei Knoten aus W induziert. Analog heißt eine Partialordnung N -erweiterbar, wenn der zugrundeliegende Komparabilitätsgraph P_4 -erweiterbar ist.

Die *erweiterte Spinnenkomposition* $P \bowtie_e \{A, B, C, D\}$ ist wie folgt definiert. Sei $P = (V, <)$ eine Partialordnung und A, B, C und D disjunkte Partialordnungen, von denen mindestens drei einelementig und die vierte eine Kette oder Antikette mit höchstens zwei Elementen ist. Dann ist

$$u < v \Leftrightarrow \begin{cases} u < v \text{ und } u, v \in P, \\ u \in C \text{ und } v \in P \cup B \cup D, \\ u \in P \cup A \text{ und } v \in B. \end{cases}$$

Abbildung 2.7 zeigt die erweiterten Spinnenkompositionen mit $|A| = 2$.

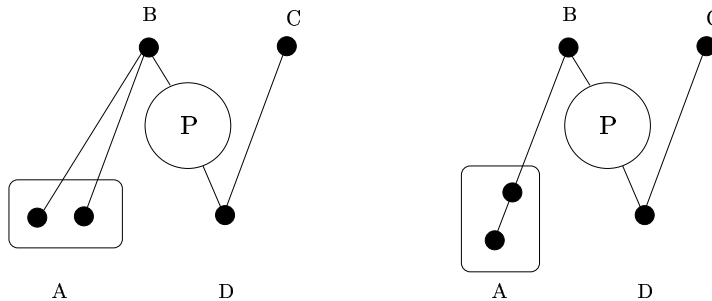
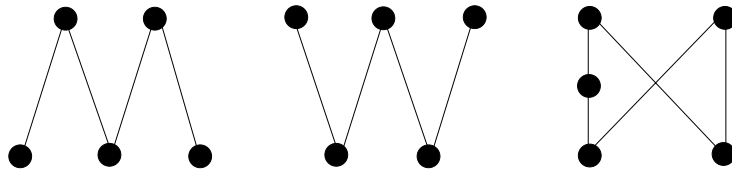


Abbildung 2.7: Erweiterte Spinnenkompositionen $P \bowtie_e \{A, B, C, D\}$

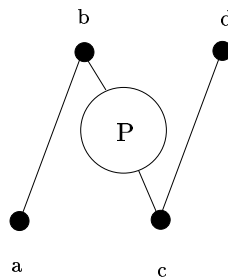
N -erweiterbare Ordnungen können aus einelementigen Partialordnungen, M 's, W 's und ungeraden Kronen OC_5 (siehe Abbildung 2.8) mittels Serien-, Parallel- und erweiterter Spinnenkomposition konstruiert werden [PW97]. Sie haben Dekompositionsweite vier, denn keines der unzerlegbaren Posets besitzt eine Antikette, die größer als vier ist.

P_4 -erweiterbare Graphen lassen sich nach Hochstättler und Schindler in linearer Zeit erkennen [HS95].

Abbildung 2.8: Spezielle N -erweiterbare Ordnungen

Der spezielle Fall, in dem alle vier Mengen A, B, C und D einelementig sind, beschreibt die „gewöhnliche“ Spinnenkomposition, welche zur Konstruktion N -dünner Posets benötigt wird. N -dünne Ordnungen sind genau die Partialordnungen, deren Komparabilitätsgraph P_4 -dünn ist. Ein Graph heißt P_4 -dünn, wenn je fünf Knoten höchstens einen P_4 induzieren. N -dünne Partialordnungen können nach Jamison und Olariu sukzessiv durch einelementige Partialordnungen mittels Serien-, Parallel- und gewöhnlicher Spinnenkompositionen – kurz: $P \bowtie_g \{a, b, c, d\}$ – aufgebaut werden [JO92] (siehe Abbildung 2.9).

Die gewöhnliche Spinnenkomposition ist eine der nichttrivialen Modulkompositionen der Weite drei. Im Kompositionsbaum $T(P)$ hat ein mit \bowtie_g markierter Knoten mindestens vier Blätter und höchstens einen inneren Knoten unter den maximal fünf Nachfolgern.

Abbildung 2.9: Gewöhnliche Spinnenkomposition $P \bowtie_g \{a, b, c, d\}$

Jamison und Olariu definieren in [JO89b] P_4 -reduzierbare Graphen, bei denen kein Knoten zu mehr als einem P_4 gehört. Für Komparabilitätsgraphen fällt diese Klasse zusammen mit den N -dünnen Ordnungen [vA93].

2.4 Das Durchlaufzeitproblem

Das Durchlaufzeitproblem, nach der Standard-Notation von Graham, Lawler, Lenstra und Rinnooy Kan [GLLRK79] kurz: $1|prec|\sum w_j C_j$, ist eines der bekanntesten und vielfach untersuchten deterministischen Ein-Maschinen-Reihenfolgeplanungsprobleme.

Gegeben ist eine Menge $J = \{j_1, j_2, \dots, j_n\}$ von Aktivitäten, auch *Jobs* genannt, die auf einer Maschine abgearbeitet werden sollen. Die Reihenfolge, in der die Aktivitäten der Maschine zugewiesen werden, muß konsistent sein mit Vorgängerrelationen, die durch eine Partialordnung $P = (J, <)$ angegeben sind. Aktivität i muß vor j abgearbeitet werden genau dann, wenn $i < j$ in P gilt. Jedem Job j ist zusätzlich eine Prozeßzeit p_j und ein Gewicht w_j zugeordnet. Allgemein bezeichnet man die Zuweisung der Jobs im zeitlichen Ablauf auf eine oder mehrere Maschinen als *Reihenfolgeplan* (engl.: schedule). Bei unserem Ein-Maschinen-Problem repräsentiert jede lineare Erweiterung $L = j_{\pi(1)} j_{\pi(2)} \dots j_{\pi(n)}$ von P einen Reihenfolgeplan. Wir setzen voraus, daß der erste Job zum Zeitpunkt $t = 0$ beginnt und es keine Leerlaufzeiten gibt. Die *Durchlaufzeit* – oder *Fertigstellungszeit* (engl.: completion time) – C_j eines Jobs j ist der Zeitpunkt an dem Job j fertiggestellt ist, d.h. seine eigene Prozeßzeit summiert mit den Prozeßzeiten aller Jobs, die vor ihm abgearbeitet wurden:

$$C_j := \sum_{\{i \mid i \leq_L j\}} p_i \quad .$$

Abbildung 2.10 verdeutlicht die Fertigstellungszeiten graphisch anhand eines sogenannten *Gantt-Charts*. Ein Gantt-Chart hat eine Dimension, die die Prozeßzeit darstellt und in dem Jobs durch Blöcke der Länge p_j repräsentiert sind.

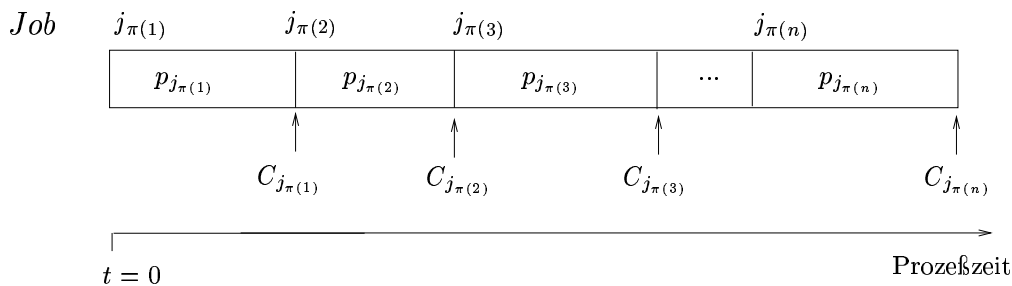


Abbildung 2.10: Durchlaufzeiten im Gantt-Chart

Da die Prozeßzeiten vorab bekannt und fest vorgegeben sind, sind die C_j wohldefiniert für jeden Reihenfolgeplan L . Wir schreiben C^L für den durch

L induzierten Fertigstellungszeitenvektor $(C_{\pi(1)}, \dots, C_{\pi(n)})$ mit $C_{\pi(1)} \leq \dots \leq C_{\pi(n)}$.

Das Ziel ist die Minimierung der Summe der gewichteten Fertigstellungszeiten $\sum w_j C_j$.

Die gewichtete Durchlaufzeit läßt sich in einem sogenannten *2D-Gantt-Chart* darstellen. Hierzu führt man eine zweite Achse ein, die den Gewichten dient. Jeder Job j ist durch ein Rechteck repräsentiert, das die Länge w_j und die Höhe p_j besitzt (siehe Figur 2.11). Die gewichtete Fertigstellungszeit $\sum w_j C_j$ entspricht der straffierten Fläche, die wir im folgenden auch einfach *Fläche* nennen. Die Minimierung von $\sum w_j C_j$ über Reihenfolgeplänen ist also äquivalent zur Minimierung der Fläche im 2D-Gantt Chart.

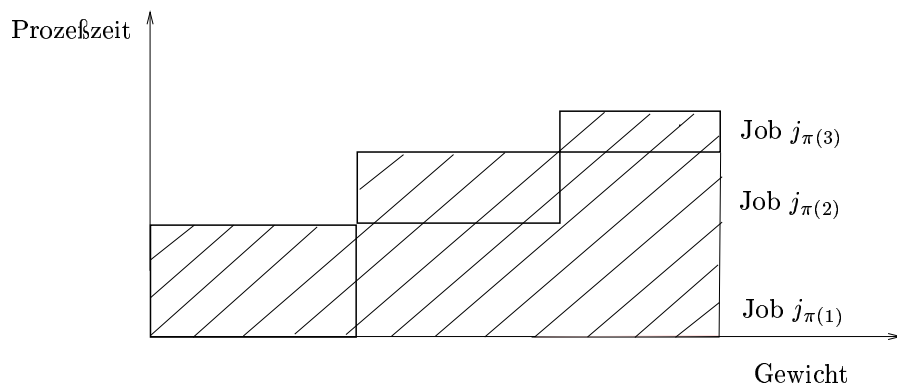


Abbildung 2.11: Gewichtete Fertigstellungszeit im 2D-Gantt-Chart

Die Geschichte des deterministischen Ein-Maschinen-Durchlaufzeit-Problems beginnt 1956 [Smi56] mit der Lösung des einfachsten Spezialfalls: dem Verzicht auf Vorgängerrelationen bzw. die Jobs bilden eine Antikette. Smith zeigt, daß das Problem dann optimal gelöst wird, wenn man die Aktivitäten nach nicht-fallenden Quotienten $\frac{p_j}{w_j}$ sortiert. Dieses Resultat ist bekannt als *Smith's rule*. Da nur sortiert werden muß, benötigt das Ermitteln der Lösung eine Laufzeit von $O(n \log n)$.

Der Fall, bei dem die Vorgängerrelationen durch parallele Ketten gegeben und alle Gewichte $w_j = 1$ sind, lösen 1967 Conway, Maxwell und Miller [CWM67].

1972 löst Horn das Problem effizient auf Partialordnungen, deren Komparabilitätsgraph ein Wurzelbaum ist [Hor72]. Adolphson und Hu geben ein Jahr später eine Implementation an, die die Laufzeit auf $O(n \log n)$ reduziert [AH73].

Die ersten Ergebnisse zum Durchlaufzeitproblem mit beliebigen Vorgängerrelationen gehen auf Sidney zurück. Wir geben hier die wichtigsten Ergebnisse aus seiner Arbeit [Sid75] an.

Sei $A \subset J$ und $\alpha = j_{\pi(1)} \dots j_{\pi(n)}$ eine lineare Erweiterung von J , dann bezeichnet $\alpha|A$ die Teilordnung, die aus allen Aktivitäten aus A besteht und die angeordnet sind wie in α . Sind $A = \{j_{\pi(1)}, j_{\pi(2)}, \dots, j_{\pi(k)}\}$ die ersten k Elemente und $B = \{j_{\pi(k+1)}, j_{\pi(k+2)}, \dots, j_{\pi(n)}\}$ die letzten $n - k$ Elemente, dann bezeichnen wir mit $\alpha = (\alpha|A, \alpha|B)$ die Konkatenation.

Für jede Teilmenge $\emptyset \neq I \subseteq V$ sei ferner

$$\rho(I) := \frac{\sum_{j \in I} p_j}{\sum_{j \in I} w_j}$$

der ρ -Wert des Ideals I . Ist jedes $w_j > 0$, dann ist $\rho(I)$ für alle $I \subseteq V$ wohldefiniert. Im Fall, daß ein $w_j = 0$ ist, wird der Job j in einem optimalen Reihenfolgeplan am Ende abgearbeitet.

Für die Durchlaufzeit gilt:

Lemma 2.6 (Sid75, Lemma 2, Adjacent string interchange lemma)

Sei $P = (J, <)$ eine Partialordnung und $A \dot{\cup} S_1 \dot{\cup} S_2 \dot{\cup} B = J$, $\alpha = (\alpha|A, \alpha|S_1, \alpha|S_2, \alpha|B)$ und $\beta = (\beta|A, \beta|S_2, \beta|S_1, \beta|B)$ zwei lineare Erweiterungen.

Dann ist die Durchlaufzeit von α genau dann geringer als die Durchlaufzeit bezüglich β , wenn $\rho(S_1) \leq \rho(S_2)$.

Ferner besitzen Module eine besondere Eigenschaft:

Theorem 2.7 (Sid75, Lemma 23) Sei $M \subseteq J$ ein Modul von $P = (J, <)$ und α eine optimale lineare Erweiterung für M . Dann existiert eine optimale Reihenfolge für J , die konsistent mit α ist (d.h. in der die Jobs in M in derselben Reihenfolge auftreten wie in α)

Eine wichtige Konsequenz aus Theorem 2.7 ist, daß das Modul M durch die berechnete optimale lineare Erweiterung α in P ersetzt werden kann. Da α eine Kette ist, bildet dann im besonderen jede Teilmenge aufeinanderfolgender Elemente in α ein Modul in P .

Zur Lösung von Modulen ist das folgende Theorem hilfreich. Wir nennen ein Ideal I^* eines Moduls M ρ -minimal, wenn $\rho(I^*) \leq \rho(I)$ für alle Ideale I aus M ist. Jedes Modul M besitzt mindestens ein ρ -minimales Ideal, möglicherweise M selbst.

Theorem 2.8 (Sid75, Lemma 3 und 5) Sei $M \subseteq J$ ein Modul von $P = (J, <)$ und I^* ein ρ -minimales Ideal von M . Dann existiert eine optimale Reihenfolge für J , in der die Jobs aus I^* hintereinander und vor allen anderen Jobs aus M stehen.

Die beiden Theoreme 2.7 und 2.8 suggerieren eine Strategie, wie das Durchlaufzeit allgemein gelöst werden kann: man identifiziert die Module einer Partialordnung im Kompositionsbaum $T(P)$, dann deren ρ -minimalen Ideale, löst die Teilprobleme und verschmelzt anschließend die berechneten Ketten. Das Identifizieren der ρ -minimalen Ideale ist effizient lösbar [Law78]. Allerdings ist das ρ -minimale Ideal eines Moduls M möglicherweise M selbst. Problematisch ist ferner das effiziente Lösen der Teilprobleme.

Sind die Vorgängerrelationen parallele Ketten, besteht das einzige Problem darin, eine Regel zum optimalen Verschmelzen zu finden. Sei M ein Modul, das nur aus zwei Ketten K_1 und K_2 besteht (siehe Abbildung 2.12). Dann existiert ein ρ -minimales Ideal I^* von M , das Ideal von nur einer der Ketten ist. Denn angenommen I^* enthält Elemente aus K_1 und K_2 . Dann ist I^* Vereinigung eines Ideals I_1 aus K_1 und I_2 aus K_2 . Da I^* ρ -minimal ist, gilt:

$$\begin{aligned} \rho(I_1 \cup I_2) \leq \rho(I_1) &\Leftrightarrow \rho(I_2) \leq \rho(I_1) \text{ und} \\ \rho(I_1 \cup I_2) \leq \rho(I_2) &\Leftrightarrow \rho(I_1) \leq \rho(I_2), \text{ d.h.} \\ \rho(I^*) &= \rho(I_1) = \rho(I_2). \end{aligned}$$

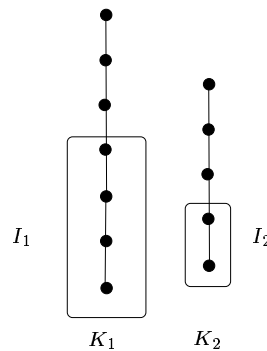


Abbildung 2.12: Parallele Ketten

Ergänzt man I^* zur optimalen Sequenz α , dann ist $M \setminus I^*$ wiederum ein Modul und man fährt fort wie oben. Dies entspricht Sidneys „parallel chain“ Algorithmus, der eine Laufzeit von $O(n^2)$ benötigt.

Lawler geht 1978 einen Schritt weiter. Er nutzt die Dekomposition serienparalleler Ordnungen und erhält durch das geschickte Zusammensetzen von Jobs einen Algorithmus mit einer Laufzeit von $O(n \log n)$. Wir erklären seine Idee aus [Law78] ausführlicher.

Lawler folgert zuerst aus Theorem 2.8, daß immer, wenn ein ρ -minimales Ideal I^* identifiziert wurde, die Jobs aus I^* ersetzt werden können durch einen einzigen *zusammengesetzten* (engl.: composite) Job. Das Gewicht und die Prozeßzeit des zusammengesetzten Jobs wird gleich $\rho(I^*)$ gesetzt. Er wird anschließend behandelt als wäre er ein einzelner Job. Genau dies nutzt er, um die gewünschte Laufzeit zu erhalten.

Der Algorithmus arbeitet den Kompositionsbaum $T(P)$ von unten nach oben ab und bestimmt eine optimale lineare Erweiterung für ein Modul M mittels vorher berechneter optimaler linearer Erweiterungen seiner Söhne M_1 und M_2 . Statt nun – wie Sidney – Ketten einer Serienkomposition zu einer einzigen Kette zu vereinigen und Ketten paralleler Kompositionen zu verschmelzen, vertauscht Lawler die Operationen Vereinigung und Verschmelzung. Der Trick besteht darin, daß die jeweils berechneten Ketten aus Jobs bestehen, manche davon gegebenenfalls zusammengesetzt, deren ρ -Werte stets nicht-fallend sind.

Konkret vereinigt man im Fall einer Parallelkomposition $M_1 \parallel M_2$ beide Mengen zu einer Kette, in der die Jobs nach nicht-fallenden Quotienten angeordnet sind. Diese Reihenfolge ist zulässig und optimal.

Bei einer Serienkomposition $M_1 \rightarrow M_2$ ist es möglich, daß einige Jobs in M_2 einen kleineren ρ -Wert haben als einige Jobs in M_1 . In diesem Fall bildet man einen zusammengesetzten Job aus einer bestimmten Anzahl von Jobs aus M_1 mit größten ρ -Werten und einer aus M_2 mit kleinsten ρ -Werten. Ziel ist es, einen zusammengesetzten Job zu bilden, dessen ρ -Wert nicht kleiner ist als der der übrigen in M_1 und nicht grösser als der der verbleibenden in M_2 . Das entscheidende Argument für dieses Vorgehen liegt begründet in der Tatsache, daß der konstruierte zusammengesetzte Job Modul und der Job selbst ρ -minimal ist.

In einem 2D-Gantt-Chart entspricht der ρ -Wert eines Jobs der Steigung der Diagonalen des Rechtecks, durch das er repräsentiert wird. Abbildung 2.13 veranschaulicht die Idee eines zusammengesetzten Jobs in einem 2D-Gantt Charts. Die Diagonale des aus Job $j_{\pi(3)}$ und Job $j_{\pi(4)}$ zusammengesetzten Jobs ist fett gestrichelt.

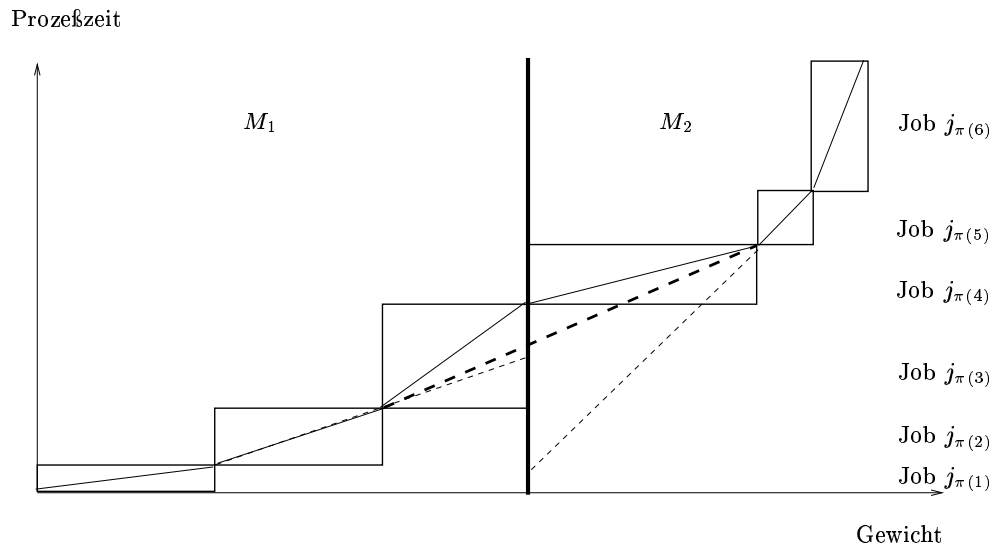


Abbildung 2.13: Zusammengesetzter Job bei Serienkomposition

Da keine der Operationen mehr als $O(n \log n)$ Zeit benötigt, erhält man

Satz 2.9 (Law78, Seite 83) *Lawlers Algorithmus löst das Durchlaufzeitproblem mit serien-parallelen Reihenfolgerestriktionen in $O(n \log n)$ Zeit.*

Goemans und Williamson geben 1998 einen alternativen Beweis zu Lawlers Algorithmus. Dieser erste dualitäts-basierte Beweis nutzt die graphische Darstellung der gewichteten Durchlaufzeit durch 2D-Gantt-Charts [GW98].

Lawler zeigt zudem, daß das Durchlaufzeitproblem mit beliebigen Reihenfolgerestriktionen NP-schwer ist, auch dann, wenn alle Gewichte w_j oder alle Prozeßzeiten p_j eins sind.

Den Prozeß, optimale lineare Erweiterungen entlang des Kompositionsbaumes zu bestimmen, verallgemeinern 1985 Möhring und Radermacher [MR85] auf Partialordnungen mit beschränktem Dekompositionsdurchmesser k . Ihr Algorithmus benötigt eine Laufzeit von $O(n^{k^2})$.

Sidney und Steiner nutzen 1986 [SS86] die Partialordnungsdekomposition in Verbindung mit dynamischer Programmierung und reduzieren so die Laufzeit von Möhring und Radermacher auf $O(n^k)$. Durch ihren Ansatz, den wir im folgenden vorstellen, können sämtliche Partialordnungen mit beschränkter Dekompositionsweite bzw- durchmesser polynomiell gelöst werden. Ihre Er-

gebnisse führen zur polynomiellen Lösbarkeit weiterer Reihenfolgeplanungsprobleme.

Dynamische Programmierung ist eine generelle Technik zur Lösung von Optimierungsproblemen. Man beginnt in der Regel mit kleinen und daher einfachen Teilaufgaben. Durch Kombination ihrer Lösungen erhält man Lösungen für Teilaufgaben zunehmender Größe bis man schließlich bei der Lösung der ursprünglichen Aufgabe ankommt. Die Effizienz der dynamischen Programmierung liegt darin begründet, daß mehrfach auftretende Teilaufgaben nur einmal ausgerechnet werden.

Bezogen auf das Durchlaufzeitproblem berechnet man rekursiv über allen Idealen $I \subseteq P$ den optimalen Zielfunktionswert $f(P)$ mittels

$$f(I) = \min\{f(I \setminus \{j\}) + w_j \sum_{i \in I} p_i \mid j \in \text{Max}(I)\} . \quad (2.1)$$

Ist K die Anzahl der Ideale in P und w die maximale Anzahl an Elementen in $\text{Max}(I)$, dann benötigt die Berechnung von $f(P)$ $O(Kw)$ Zeit. Mittels $f(P)$ wird dann die optimale Reihenfolge der Jobs generiert, in dem man mit $I = P$ beginnt und ein $i \in I$ identifiziert, für das das Minimum angenommen wird. Job i steht an letzter Stelle und man wiederholt die Berechnung mit $I := I \setminus i$. Die Prozedur endet, wenn I leer ist und benötigt eine Laufzeit von $O(nw)$.

Maßgeblich für die Laufzeit des dynamischen Programmieralgorithmus ist also die Anzahl der Ideale. Diese kann in einem Poset mit n Jobs bis zu 2^n wachsen, nämlich dann, wenn keine Vorgängerrelationen vorliegen. Tatsächlich ist K im allgemeinen wesentlich kleiner [SB78]. Interessant ist also, für welche Klasse von Posets K beschränkt durch ein Polynom in n ist.

Da das durch $\text{Max}(I)$ induzierte Poset in P eine Antikette für jedes Ideal I ist, ist w mit dem Dilworth-Theorem [Dil50] kleiner oder gleich der kleinsten Anzahl Ketten, die zur Partitionierung von P benötigt wird. Eine solche Partitionierung kann in polynomieller Zeit durch Standard-Techniken wie Netzwerkflüssen oder der Bestimmung eines maximalen Matchings gefunden werden [Law76]. Sei $P = (J, <)$ partitioniert in w Ketten mit k_1, k_2, \dots, k_w Knoten. Dann kann jedes Ideal I über die Anzahl der Knoten aus jeder Kette definiert werden, d.h.

$$\begin{aligned} K &\leq (1 + k_1)(1 + k_2) \cdots (1 + k_w) \\ &\leq \binom{n+w}{w}^w \\ &= O(n^w) \end{aligned}$$

Ist \mathcal{A}_w die Klasse aller Posets mit Weite kleiner oder gleich w , folgt

Satz 2.10 (SS86, Theorem 1) $1 \mid \mathcal{A}_w \mid w_j C_j$ ist lösbar mit einer Laufzeit von $O(n^w)$.

Dieses Ergebnis kann auf eine weitere Klasse \mathcal{B}_w von Posets verallgemeinert werden. P gehört genau dann zu \mathcal{B}_w , wenn P sukzessiv aufgebaut werden kann durch eine endliche Anzahl von Kompositionen von Posets, bei denen jeder äußerer Faktor aus \mathcal{A}_w ist, also beschränkte Weite hat. Beispielsweise gehören serien-parallele Posets zu \mathcal{B}_2 , denn wenn $P = P_0[P_1, P_2, \dots, P_m]$ für $m > 1$ eine serien-parallele Komposition ist, dann kann P auch durch $m - 1$ sukzessive Kompositionen mit je zweielementigen Ketten oder Antiketten aufgebaut werden. \mathcal{B}_w enthält natürlich \mathcal{A}_w , ist aber wesentlich grösser, denn \mathcal{B}_w ist im Sinne des folgenden Lemmas abgeschlossen.

Lemma 2.11 (SS86, Lemma 4) Sei $P_0 = (J_0, <_0) \in \mathcal{B}_w$, $|J_0| = m$ und $P_1, P_2, \dots, P_m \in \mathcal{B}_w$. Dann ist $P_0[P_1, P_2, \dots, P_m]$ auch in \mathcal{B}_w .

Für festes $w \geq 2$ enthält \mathcal{B}_w unendlich viele Posets beliebig großer Weite. Zum Beispiel hat die Partialordnung in Figur 2.14 Weite vier, aber jeder äußere Faktor, der zum sukzessiven Aufbau benötigt wird, hat nur Weite zwei, d.h. $P \in \mathcal{B}_2$.

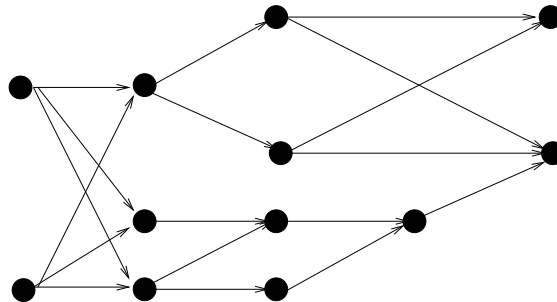


Abbildung 2.14: Poset $\in \mathcal{B}_2$

Der Kompositionsbaum $T(P)$ besitzt also nur Prim-Kompositionen aus \mathcal{A}_w . Die Berechnung dieser benötigt mit Satz 2.10 eine Laufzeit von $O(n^w)$. Serien- und Parallelkompositionen sind mit Lawler in $O(n \log n)$ zu lösen. Da der Kompositionsbaum $T(P)$ eines Posets $P = (V, <)$ höchstens $2n - 1$ Knoten besitzt [BM83], folgt:

Satz 2.12 (SS86, Theorem 5) $1 \mid \mathcal{B}_w \mid w_j C_j$ ist lösbar mit einer Laufzeit von $O(n^{w+1})$.

1987 beweisen Habib und Möhring [HM87] die NP-Vollständigkeit des Durchlaufzeitproblems auf N -freien Ordnungen. Sie zeigen auch, daß die Anzahl N -freier Ordnungen verschwindend gering ist.

Die ersten polyedrischen Beschreibungen für das Durchlaufzeitproblem auf serien-parallelen und N -dünnen Ordnungen gehen unabhängig voneinander auf von Arnim, Faigle und Schrader und Schulz [vAFS90],[vASW96], [vAS97] und Queyranne und Wang [QW91] zurück.

Queyranne und Wang optimieren über dem Polytop, dessen Ecken aus allen zulässigen Permutationen besteht, die durch Fertigstellungszeitenvektoren $C^{\pi^{-1}}$ beschrieben sind. Man nennt dieses Polytop auch *generalisiertes Permutaeder*.

Der Name *Permutaeder* geht auf Schoute 1911 [Sch11] zurück, der damit die konvexe Hülle aller Permutationen einer Menge $V = \{1, 2, \dots, n\}$ bezeichnet und diese durch *Permutationsvektoren* beschreibt. Ein Permutationsvektor zu einer Permutation α ist der Vektor $x(\alpha) := (\alpha(1), \alpha(2), \dots, \alpha(n))$. Vollständige Beschreibungen des Permutaeders wurden unabhängig von verschiedenen Autoren gegeben (vgl. [Rad52], [Bal75] und [You78]).

Von Arnim, Faigle und Schrader geben zusätzlich zur Beschreibung mittels Fertigstellungszeitenvektoren eine vollständige Beschreibung über allen zulässigen Permutationsvektoren, falls alle $p_i = 1$ oder alle $w_i = 1$.

Ein sehr guter Überblick über weitere polyedrische Ergebnisse findet sich in Queyranne und Schulz [QS95].

1993 verbessert Schulz [Sch93] das Ergebnis von Sidney und Steiner für das Durchlaufzeitproblem über N -dünnen Ordnungen, die Dekompositionsweite drei haben, auf eine Laufzeit von $O(n^2)$.

Peter und Wambach [PW97] lösen durch Modifikation des Algorithmus von Lawler das Durchlaufzeitproblem mit N -erweiterbaren Reihenfolgebedingungen in einer Laufzeit von $O(n \log n)$. Da N -erweiterbare Partialordnungen eine echte Obermenge N -dünnere Ordnungen sind und Dekompositionsweite vier haben, verbessern sie sowohl das Ergebnis von Schulz als auch das von Sidney und Steiner.

Kapitel 3

$(q, q-3)$ -Graphen

In [BO98] wird die Klasse der (q, t) -Graphen vorgestellt, die – lokal gesehen – eine geringe Anzahl von induzierten Pfaden der Länge drei besitzen. Genauer besitzt in einem (q, t) -Graph jeder induzierte Teilgraph mit höchstens q Knoten maximal t induzierte Pfade der Länge drei.

$(q, q-3)$ -Graphen sind per Definition $(q, q-4)$ -Graphen und verallgemeinern damit im besonderen Cographen, P_4 -dünne und P_4 -erweiterbare Graphen. Die Einschränkung dieser drei speziellen $(q, q-4)$ -Graphen ($q = 4, 5, 6$) auf Komparabilitätsgraphen führt ordnungstheoretisch zu serien-parallelen, N -dünnen und N -erweiterbaren Ordnungen, für die das Durchlaufzeitproblem polynomiell lösbar ist.

In diesem Kapitel sind die für uns relevanten Ergebnisse über $(q, q-3)$ -Graphen mit $q \geq 7$ und einleitend über $(q, q-4)$ -Graphen aus [Bab97] und [BO98] zusammengefaßt.

Die Basis für diese Ergebnisse bildet das Strukturtheorem 2.1, das erlaubt, sich auf die Beschreibung der p -zusammenhängenden Teilgraphen zu beschränken. Denn ist sowohl G als auch \overline{G} zusammenhängend, dann ist entweder G selbst p -zusammenhängend oder G besitzt eine echte separable p -Zusammenhangskomponente H . Im letzten Fall ist H im besonderen p -zusammenhängend.

Im nächsten Kapitel nutzen wir die Ergebnisse zur Charakterisierung von $(q, q-3)$ - bzw. $(q, q-4)$ -Partialordnungen.

3.1 $(q, q-4)$ -Graphen

Wir benötigen zuerst die Definition von *Spinnen*. Ein Graph $G = (V, E)$ heißt *Spinne*, wenn seine Knotenmenge in zwei disjunkte Mengen S und K aufgeteilt werden kann, so daß

- $|S| = |K|$, S unabhängig und K Clique ist und
- eine bijektive Abbildung $f : S \rightarrow K$ existiert, für die entweder

$$N(s) = \{f(s)\} \text{ für alle Knoten } s \in S$$

oder

$$N(s) = K \setminus \{f(s)\} \text{ für alle Knoten } s \in S \quad .$$

Die kleinste Spinne und die einzige, die sowohl dick als auch dünn ist, ist ein P_4 . Spinnen mit mehr als vier Knoten nennt man *echte Spinnen*. Wenn die erste der oben genannten Alternativen für f zutrifft, spricht man von Spinnen mit *dünnen Beinen*, im zweiten Fall von *dicken Beinen*. Offensichtlich ist das Komplement einer Spinne mit dünnen Beinen eine mit dicken und umgekehrt. Abbildung 3.1 zeigt die dicke und dünne Spinne mit acht Knoten.

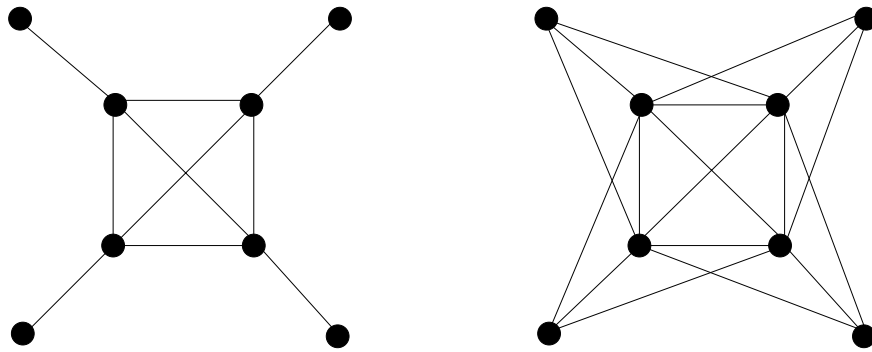


Abbildung 3.1: Eine dünne und eine dicke Spinne

Spinnen sind separabel.

Spinnen mit q Knoten besitzen genau $\frac{1}{8}q(q-2)$ P_4 s (Bab98, Beobachtung 8.1.1) Da eine dünne (dicke) Spinne mit q Knoten jede dünne (dicke) Spinne mit weniger als q Knoten als Untergraph enthält, gilt:

Lemma 3.1 (Bab98, Beweis zu Proposition 8.1.2) *In einer Spinne mit q Knoten induzieren $k < q$ Knoten $\frac{1}{8}k(k-2)$ P_4 s, wenn k gerade ist und $\frac{1}{8}(k-1)(k-3)$ P_4 s, wenn k ungerade ist.*

Spinnen sind genau die *minimal p -zusammenhängenden* Graphen, d.h das Entfernen eines beliebigen Knoten führt dazu, daß der Restgraph nicht mehr p -zusammenhängend ist. Dadurch erhält man zwei sehr hilfreiche Eigenschaften p -zusammenhängender Graphen.

Lemma 3.2 (Bab98, Beobachtung 4.2.3) *Sei G p -zusammenhängend. Dann existiert eine Ordnung $(v_n, v_{n-1}, v_{n-2}, \dots, v_1)$ der Knoten und eine Zahl $k \in \{4, 5, \dots, n\}$, so daß gilt: $G(\{v_i, v_{i-1}, \dots, v_1\})$ ist p -zusammenhängend für $i=k, \dots, n$ und eine Spinne für $i = k$.*

Dies führt zu einer unteren Schranke für die Anzahl der P_4 s.

Korollar 3.3 (Bab98, Korollar 4.2.4) *Ein p -zusammenhängender Graph $G = (V, E)$ besitzt mindestens $n - 3$ Pfade der Länge drei.*

Ein Beweis zum folgenden Theorem, das $(q, q - 4)$ -Graphen charakterisiert, findet sich auch in [JO92].

Theorem 3.4 (Bab98, Theorem 8.1.3) *Sei $G = (V, E)$ p -zusammenhängend.*

- (a) *Ist G ein $(5, 1)$ -Graph, dann ist G eine Spinne.*
- (b) *Ist G ein $(7, 3)$ -Graph, dann ist $|V| < 7$ oder G ist eine Spinne.*
- (c) *Ist G ein $(q, q - 4)$ -Graph, $q = 6$ oder $q \geq 8$, dann ist $|V| < q$.*

Nach Jamison und Olariu [JO92] besitzen P_4 -erweiterbare Graphen p -zusammenhängende Teilgraphen, die aus höchstens fünf Knoten bestehen. Da der C_5 der einzige p -zusammenhängende Graph mit fünf Knoten ist, der mehr als zwei induzierte P_4 s besitzt, ist mit Theorem 3.4 klar, daß die C_5 -freien P_4 -erweiterbaren Graphen genau die $(6, 2)$ -Graphenklasse bilden.

Theorem 3.5 (Bab98, Theorem 8.2.1) *Für festes q können $(q, q - 4)$ -Graphen in linearer Zeit erkannt werden.*

Viele klassische Optimierungsprobleme sind auf $(q, q - 4)$ -Graphen mit fest vorgegeben q in linearer Zeit lösbar: u.a Isomorphietest, Cliquenzahl, chromatische Zahl [Bab97], Dominanzprobleme und das Steiner-Baum Problem [BKK⁺01].

Sei $\mathcal{G}(q, t)$ die Menge aller (q, t) -Graphen. Dann gelten folgende Inklusionen.

Korollar 3.6 (Bab98, 8.1.4)

- (a) $\mathcal{G}(4,0) \subset \mathcal{G}(5,1)$.
- (b) $\mathcal{G}(6,2) \subset \mathcal{G}(7,3)$.
- (c) $\mathcal{G}(6,2) \subset \mathcal{G}(q,q-4) \subset \mathcal{G}(q+1, q-3)$ für $q \geq 8$.

Daß die Inklusionen strikt sind, beweisen die folgenden Beispiele:

- (a) ein P_4 ,
- (b) ein P_6 und
- (c) ein P_6 bzw. ein P_q .

Die Klassen $\mathcal{G}(5,1)$ und $\mathcal{G}(6,2)$ sind nicht vergleichbar wie der Pfad auf fünf Knoten und eine Spinne mit sechs Knoten belegen.

Ein Graph heißt P_4 -leicht, wenn jeder induzierte Teilgraph mit höchstens sechs Knoten entweder höchstens zwei induzierte P_4 s enthält oder isomorph ist zu einer Spinne [JO89a]. Da eine Spinne auf sechs Knoten drei P_4 s induziert, ist die Klasse aller P_4 -leichten Graphen eine echte Obermenge von $\mathcal{G}(5,1)$ und $\mathcal{G}(6,2)$. Auf der anderen Seite ist sie eine echte Teilmenge der Klasse der $(7,3)$ -Graphen.

Ein Graph heißt P_4 -reduzierbar, wenn jeder Knoten in maximal einem induzierten P_4 vorkommt.

Das Verhältnis der vorgestellten Graphenklassen mit wenigen P_4 s ist in Abbildung 3.2 dargestellt.

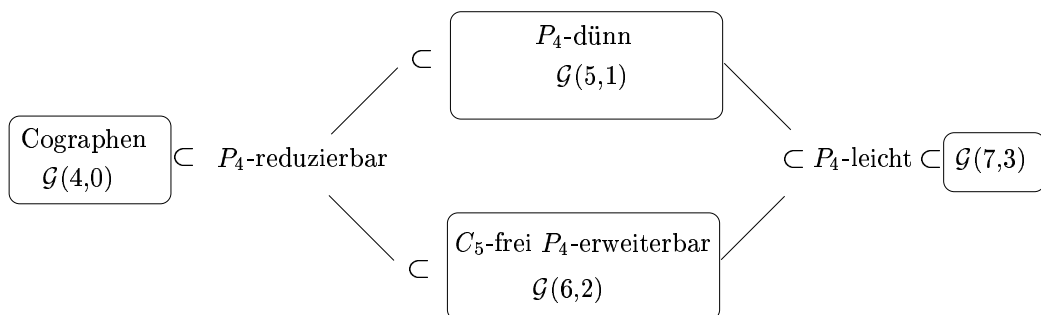


Abbildung 3.2: Graphen mit wenigen P_4 s

3.2 $(q, q-3)$ -Graphen

$(q, q-3)$ -Graphen sind eine echte Obermenge der $(q, q-4)$ -Graphen, wie ein sehnloser Weg der Länge k , $k \geq 5$, belegt. Zur Charakterisierung benötigen wir die Definitionen von p -Kreisen, p -Bäumen und *angespitzten* p -Ketten sowie einige Sätze.

Ein p -Kreis ist ein Graph, in dem jeder Knoten zu mindestens zwei P_4 s gehört und der minimal ist mit dieser Eigenschaft, d.h. jeder induzierte Teilgraph besitzt einen Knoten, der höchstens zu einem P_4 gehört. Natürlich sind p -Kreise p -zusammenhängend.

Satz 3.7 (Bab98, Theorem 7.5.5) *Ein p -Kreis mit mindestens acht Knoten ist ein Kreis C_n oder das Komplement eines Kreises \overline{C}_n .*

Im besonderen sind p -Kreise mit mindestens acht Knoten nicht separabel. Beispiele für p -Kreise mit weniger als acht Knoten sind in Abbildung 3.3 dargestellt.

Lemma 3.8 (Bab98, Propositionen 8.4.4 und 8.4.5)

- (i) *Ein p -Kreis mit mehr als q Knoten ist ein $(q, q-3)$ -Graph.*
- (ii) *Ein p -Kreis mit q Knoten ist kein $(q, q-3)$ -Graph. Die einzigen Ausnahmen sind Spinnen mit sechs Knoten.*

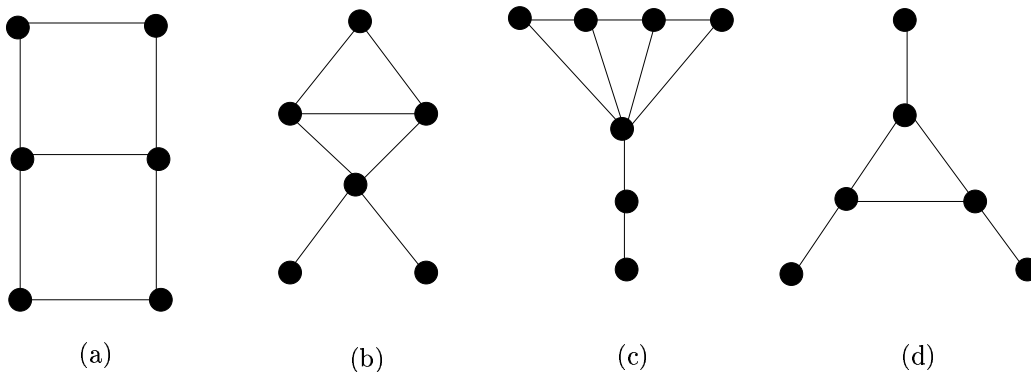


Abbildung 3.3: Kleine p -Kreise

Ein p -Baum ist ein p -zusammenhängender Graph ohne induzierte p -Kreise. Der kleinste p -Baum ist ein P_4 oder ein einzelner Knoten. Eine p -Kette der

Länge $n - 1$, die zwei Knoten u und v miteinander verbindet, ist definiert als Sequenz paarweise verschiedener Knoten (v_1, v_2, \dots, v_n) , so daß

- $u = v_1, v = v_n$ und
- $X_i := \{v_i, v_{i+1}, v_{i+2}, v_{i+3}\}$ induziert einen P_4 für $i = 1, 2, \dots, n - 3$.

Die einfachsten p -Ketten sind Pfade P_n und deren Komplemente \overline{P}_n . Eine p -Kette $(v_1, v_2, \dots, v_{n-1}, v_n)$ heißt deshalb auch *einfach*, wenn im durch die Knoten $\{v_1, v_2, \dots, v_{n-1}, v_n\}$ induzierten Graphen kein P_4 existiert, der nicht von der Form $(v_i, v_{i+1}, v_{i+2}, v_{i+3})$, $1 \leq i \leq n - 3$, ist. Mit anderen Worten, eine p -Kette der Länge $n - 1$ ist genau dann einfach, wenn die Knoten genau $n - 3$ P_4 s induzieren.

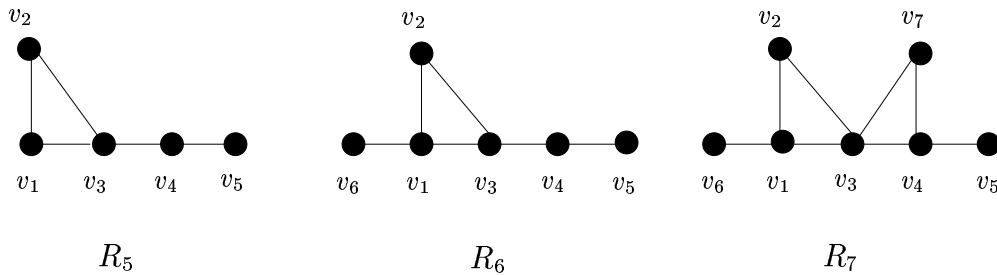


Abbildung 3.4: Kleine p -Ketten: R_5, R_6, R_7

In einem p -Baum sind alle p -Ketten einfach. Wie sich zeigt, gibt es nur wenige verschiedene einfache p -Ketten. Beispiele sind die Graphen R_5, R_6, R_7 aus Abbildung 3.4 und die Graphen Q_n (siehe Abbildung 3.5).

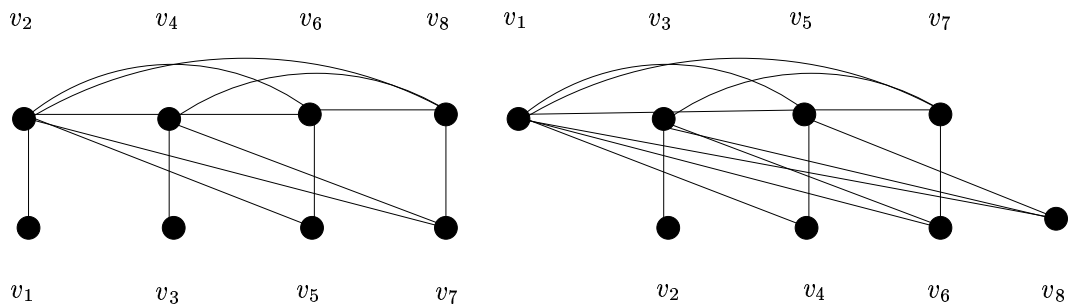


Abbildung 3.5: Q_8 und \overline{Q}_8

Seien v_1, v_2, \dots, v_n Knoten und sei $N(v_i)^+$ (bzw. $\overline{N}(v_i)^+$) die Menge aller Nachbarn (bzw. Nicht-Nachbarn) von v_i mit einem Index grösser als i . Für $i \geq 1$ ist dann Q_n definiert durch

$$N(v_{2i-1})^+ = \{v_{2i}\} \text{ und } \overline{N}(v_{2i})^+ = \{v_{2i+1}\}.$$

Wir nennen einen Q_n auch *Spinnennetz*. Im Gegensatz zu Pfaden P_n sind Q_n s für $n \geq 5$ separabel.

Theorem 3.9 (Bab98, 7.2.3) *Eine einfache p-Kette ist isomorph zu einem der Graphen P_n, Q_n ($n \geq 4$), R_5, R_6, R_7 oder den Komplementen dieser Graphen.*

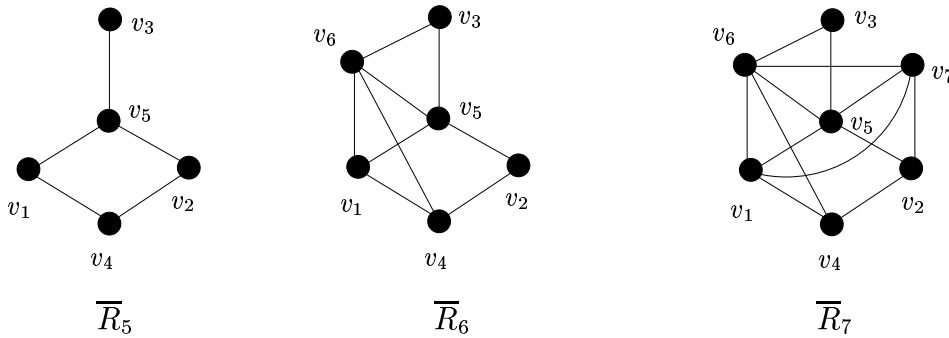


Abbildung 3.6: Kleine p -Ketten: $\overline{R}_5, \overline{R}_6, \overline{R}_7$

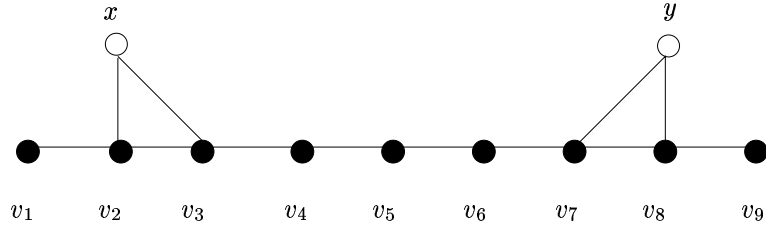
Auch p -Bäume besitzen eine handliche Charakterisierung. Wir benötigen dazu die Definition sogenannter *angespitzter p-Ketten*. Sie sind informell p -Ketten mit einigen wenigen zusätzlichen Knoten, die alle p -Endknoten sind.

Formal ist eine *angespitzte p-Kette* \vec{P}_n eine p -Kette $P_n = (v_1, v_2, \dots, v_n)$ $n \geq 6$ mit zusätzlichen Knoten x und y , so daß

- $N(x) = \{v_2, v_3\}$
- $N(y) = \{v_{n-2}, v_{n-1}\}$ und
- x und y gehören zu keinem P_4 .

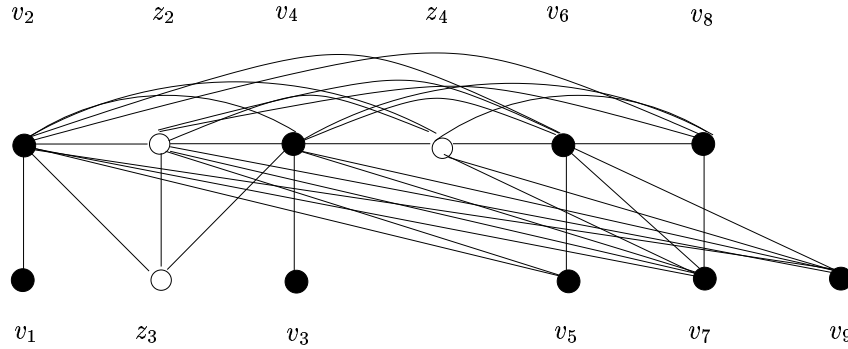
Einer oder beide Knoten x und y können fehlen (siehe Abb. 3.7). Allerdings ist in einer angespitzten P_6 -Kette nur einer der beiden Knoten erlaubt.

Eine *angespitzte p-Kette* \vec{Q}_n – oder kurz: ein *angespitztes Spinnennetz* – ist eine p -Kette $Q_n = (v_1, v_2, \dots, v_n)$, $n \geq 6$, mit zusätzlichen Knoten z_2, z_3, \dots, z_{n-5} , so daß

Abbildung 3.7: Angespitzte p -Kette \vec{P}_9

- $\overline{N}(z_i) = \{v_1, v_3, \dots, v_{i-1}, v_{i+1}\} \cup \{z_3, z_5, \dots, z_{i-1}\}$ für gerade i und
- $N(z_i) = \{v_2, v_4, \dots, v_{i-1}, v_{i+1}\} \cup \{z_2, z_4, \dots, z_{i-1}\}$ für i ungerade.

Jeder der p -Endknoten z_2, z_3, \dots, z_{n-5} kann fehlen (siehe Abbildung 3.8).

Abbildung 3.8: Angespitzte p -Kette \vec{Q}_9

Als p -Endknoten induziert jedes z_i genau einen weiteren P_4 , nämlich $z_i v_{i+1} v_{i+2} v_{i+3}$. Damit ist die $(q, q-3)$ -Eigenschaft erhalten. Anschaulich gesprochen verhält sich der Knoten z_i wie der Knoten v_i in Bezug auf den induzierten Pfad der Länge drei, bei dem v_i kleinsten Index hat.

Eine *angespitzte p -Kette* $\overline{\vec{P}}_n$ (oder $\overline{\vec{Q}}_n$) ist das Komplement einer angespitzten p -Kette \vec{P}_n (oder \vec{Q}_n).

Natürlich sind auch angespitzte p -Ketten p -Bäume. Sie bleiben auch dann p -Bäume, wenn man die p -Endknoten durch Cographen ersetzt. Es gilt sogar die Rückrichtung.

Theorem 3.10 (Bab98, Theorem 7.4.1) *Ein Graph ist genau dann ein p -Baum, wenn er entweder ein P_4 ist, bei dem ein Knoten durch einen Cograph ersetzt ist oder wenn er eine (angespitzte) p -Kette (im besonderen: R_5 , R_6 , R_7 , P_n , \vec{P}_n , Q_n , \vec{Q}_n und deren Komplemente) ist, bei der die p -Endknoten durch Cographen ersetzt sind.*

Wir kommen nun zur Charakterisierung von $(q, q - 3)$ -Graphen in Abhängigkeit ihrer Knotenanzahl.

Theorem 3.11 (Bab98, Theorem 8.4.7) *Ein p -zusammenhängender Graph G mit q Knoten, $q \geq 7$, ist $(q, q - 3)$ genau dann, wenn G ein p -Baum ist.*

Für Spinnen gilt:

Theorem 3.12 (Bab98, Theoreme 8.4.8 und 8.4.9) *Ein p -zusammenhängender Graph G mit 8 Knoten (bzw. 10 Knoten) ist ein $(7, 3)$ -Graph (bzw. ein $(9, 6)$ -Graph) genau dann, wenn G eine Spinne ist.*

Für Graphen mit mehr als q Knoten gilt:

Theorem 3.13 (Bab98, 8.4.8 und 8.4.9) *Ein p -zusammenhängender Graph G mit $n > q$ Knoten, $q = 8$ oder $q \geq 10$, ist ein $(q, q - 3)$ -Graph genau dann, wenn eine der folgenden Alternativen gilt:*

- (a) G ist p -Baum;
- (b) G ist ein Kreis C_n oder das Komplement eines Kreises \overline{C}_n .

Ein p -zusammenhängender Graph mit $k < q$ Knoten ist natürlich auch dann ein $(q, q - 3)$ -Graph, wenn er auf diesen k Knoten bis zu $q - 3$ P_4 s induziert.

Babel entwickelt basierend auf diesen Ergebnissen einen Algorithmus, der effizient testet, ob ein Graph ein $(q, q - 3)$ -Graph ist. Dazu stellt man den Graphen nach dem Strukturtheorem 2.1 als Baum dar und testet, ob die Blätter, die den p -zusammenhängenden Komponenten des Graphen entsprechen, p -Bäume, Kreise, Komplemente von Kreisen oder Spinnen sind. Ferner müssen sämtliche maximalen homogenen Mengen Cographen induzieren.

Die Identifikation maximaler homogener Mengen und das Testen, ob diese Mengen keinen induzierten P_4 enthalten, benötigt lineare Zeit in der Anzahl der Knoten (siehe [MS94],[CH94]). Anschließend schrumpft man jede maximale homogene Menge zu einem einzelnen Knoten und betrachtet so die

charakteristischen Graphen der p -zusammenhängenden Komponenten.

Hat ein Blatt weniger als q Knoten, zählt man die Anzahl der P_4 s. Da q fest vorgegeben ist, benötigt das Zählen konstante Zeit.

Besitzt ein Blatt exakt q Knoten, prüft man, ob es sich um einen p -Baum handelt.

Besitzt ein Blatt mehr als q Knoten wird getestet, ob es ein p -Baum, ein Kreis oder ein Spinne (bei genau 8 bzw. 10 Knoten) ist.

Tests auf Kreise C_n , deren Komplemente \overline{C}_n , Spinnen, angespitzte Pfade \vec{P}_n oder deren Komplemente $\overline{\vec{P}}_n$ lassen sich augenscheinlich in linearer Zeit realisieren.

Ist der charakteristische Graph des Blatts ein P_4 , muß geprüft werden, ob nur einer der Knoten ein Cograph war.

Um Spinnennetze Q_n zu erkennen, identifiziert man den Knoten v mit größtem Grad und den Nachbarn w von v , der kleinsten Grad hat, entfernt diese und wiederholt den Vorgang. v muß eindeutig sein, $d(w) = 1$ und $d(v) = k - 2$ muß gelten, wobei k der Anzahl der Knoten im jeweils betrachteten (Rest-)Graphen entspricht. Handelt es sich um einen Q_n und numeriert man die identifizierten Knoten v, w aufsteigend, erhält man die Numerierung aus der Definition der Q_n . Die Korrektheit gilt wegen $Q_n \setminus \{v_1, v_2, \dots, v_k\} \simeq Q_{n-k}$ (vgl. (**) auf Seite 60). Das Verfahren ist leicht modifiziert übertragbar auf angespitzte Spinnennetze \vec{Q}_n (siehe Details in [Bab97], Seite 78) und die Komplemente \overline{Q}_n und $\overline{\vec{Q}}_n$ (vgl. Satz 4.6). Geeignete Datenstrukturen und Sortierverfahren führen zur gewünschten Laufzeit.

Theorem 3.14 (Bab98, Theorem 8.2.12) *Für festes q können $(q, q-3)$ -Graphen in linearer Zeit erkannt werden.*

Kapitel 4

$(q, q-3)$ -Partialordnungen

Basierend auf Babels Ergebnissen wird in diesem Kapitel analysiert, welche $(q, q-3)$ -Graphen, $q \geq 7$, Komparabilitätsgraphen sind und damit Partialordnungen assoziieren.

Wegen Lemma 2.2 kann die transitive Orientierbarkeit für jede Λ^* -Klasse separat geprüft werden. Wegen des Strukturtheorems 2.1 reicht es aus, die transitive Orientierbarkeit der p -Zusammenhangskomponenten zu untersuchen. Denn im Fall separabler p -Zusammenhangskomponenten kann keine Λ^* -Klasse existieren, die sowohl Kanten aus der eindeutigen Separation H als auch Kanten aus $G \setminus H$ enthält, da die inzidenten Knoten einer Kante $uv = e \in G \setminus H$ adjazent zu allen Knoten aus H_1 und zu keinem in H_2 sind (siehe Beispiel in Abbildung 4.1).

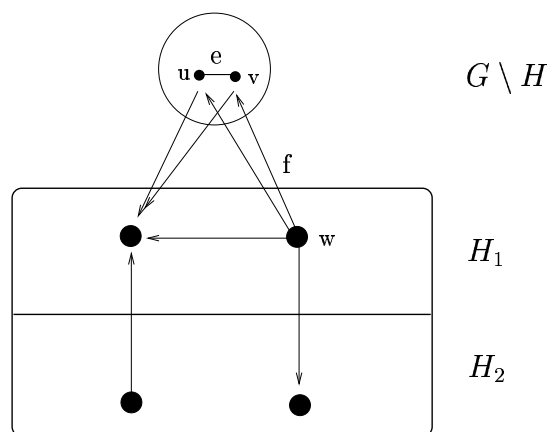


Abbildung 4.1: Separable p -Zusammenhangskomponente

Da ein einzelner Knoten in einem Graphen eine homogene Menge ist, ist ein Graph transitiv orientierbar, wenn einzelne Knoten durch Cographen (die serien-parallele Ordnungen assoziieren) ersetzt sind und sein charakteristischer Graph transitiv orientierbar ist. Bis auf P_4 s, in denen ein Graph durch einen Cograph ersetzt ist, werden in diesem Kapitel ausschließlich charakteristische Graphen betrachtet.

Wegen Satz 2.4 besitzt der charakteristische Graph eines p -zusammenhängenden Graphen entweder genau zwei transitive Orientierungen oder gar keine.

Eine weitere Konsequenz des Strukturtheorems 2.1 ist, daß es den konstruktiven Aufbau beliebiger Partialordnungen ermöglicht, sofern man sämtliche p -zusammenhängenden Komponenten kennt. Denn überträgt man das Strukturtheorem auf Komparabilitätsgraphen $G(P)$, entsprechen (i) und (ii) der bekannten seriellen und parallelen Komposition und (iii) bietet die dritte und letzte Möglichkeit assoziierte Partialordnungen P konstruktiv mittels Prim-Kompositionen aufzubauen (siehe Satz 2.3).

Die einzelnen Abschnitte dieses Kapitels widmen sich nacheinander den in Kapitel 3 vorgestellten p -Zusammenhangskomponenten mit n Knoten:

- Spinnen für $q \in \{7, 9\}$ und $n > q$,
- Kreise C_n und deren Komplemente \overline{C}_n für $n > q$,
- Pfade P_n und angespitzte Pfade \vec{P}_n für $n \geq q$,
- Komplemente von Pfaden \overline{P}_n und angespitzten Pfaden $\overline{\vec{P}}_n$ für $n \geq q$,
- P_4 s, in denen ein Knoten durch einen Cograph ersetzt ist,
- Spinnennetze Q_n und angespitzte Spinnennetze \vec{Q}_n sowie deren Komplemente $\overline{Q}_n, \overline{\vec{Q}}_n$ für $n \geq q$ und
- Graphen mit $n < q$ Knoten, im Speziellen R_5, R_6 und R_7 (falls $q > 7$) aus Abbildung 3.4 und deren Komplemente, die höchstens $q - 3$ induzierte P_4 s besitzen.

Die vier separablen p -zusammenhängenden Komponenten, die zugleich Komparabilitätsgraphen sind, werden separat in den angegebenen Abschnitten behandelt:

1. den P_4 (Abschnitt 4.1),

2. den P_4 , in dem ein Knoten durch einen Cograph ersetzt sein kann (Abschnitt 4.3),
3. Spinnennetze Q_n , angespitzte Spinnennetze \vec{Q}_n und deren Komplemente $\overline{Q}_n, \overline{\vec{Q}}_n$ (Abschnitt 4.4),
4. separable p -zusammenhängende Graphen mit beschränkter Knotenanzahl kleiner q , die maximal $q - 3$ P_4 s induzieren (Abschnitt 4.5).

Im letzten Abschnitt fassen wir die Ergebnisse zusammen und geben eine Vorschrift zum konstruktiven Aufbau von $(q, q - 3)$ -Partialordnungen und zusätzlich von der Subklasse der $(q, q - 4)$ -Partialordnungen an.

4.1 Spinnen

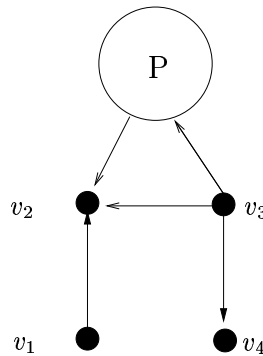
Spinnen mit mehr als sieben (bzw. mehr als neun) Knoten induzieren mit Lemma 3.1 auf sieben (bzw. neun) Knoten maximal drei (bzw. sechs) P_4 s. Allerdings sind nur Spinnen mit vier Knoten Komparabilitätsgraphen.

Lemma 4.1 *Sei $G(P)$ ein Komparabilitätsgraph. Dann besitzt $G(P)$ keine Spinne mit mehr als vier Knoten als induzierten Teilgraphen.*

Beweis. Eine dünne (dicke) Spinne mit sechs Knoten besitzt keine transitive Orientierung (siehe Seite 8 und Abbildung 2.4) und jede dünne (dicke) Spinne mit mehr als sechs Knoten besitzt eine kleinere dünne (dicke) Spinne als induzierten Teilgraph. \square

Die dicke und die dünne Spinne auf vier Knoten sind isomorph. Der P_4 ist also die einzige Spinne, die transitiv orientierbar ist.

Nun ist ein $P_4 = v_1v_2v_3v_4$ separierbar. In einem $(q, q - 3)$ -Komparabilitätsgraphen können also separable p -Zusammenhangskomponenten existieren, in dem jeder Knoten eines beliebigen $(q, q - 3)$ -Komparabilitätsgraphen adjazent ist zu den Mittelknoten v_2 und v_3 . Orientiert man die Kanten transitiv, erhält man die schon von Seite 14 bekannte gewöhnliche Spinnenkomposition $P \bowtie_g \{v_1, v_2, v_3, v_4\}$. Abbildung 4.2 veranschaulicht dies für den Fall $v_1 \rightarrow v_2$.

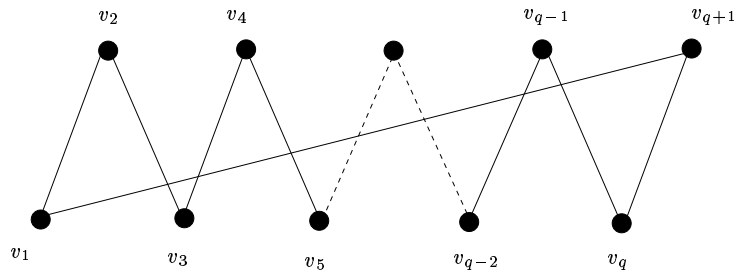
Abbildung 4.2: Separable p -Zusammenhangskomponente mit $H = P_4$

4.2 Kreise und deren Komplemente

Die p -zusammenhängenden p -Kreise in einem $(q, q-3)$ -Graphen für $q \geq 7$ sind Kreise C_n und deren Komplemente \bar{C}_n mit mindestens acht Knoten. Die Reduktion auf Komparabilitätsgraphen ist relativ streng, denn nur Kreise mit gerader Länge größer oder gleich vier sind transitiv orientierbar.

Satz 4.2 *Kreise mit gerader Länge $q+1$, $q \geq 3$ sind Komparabilitätsgraphen. Sie gehören zu den Klassen der $(q+1, q+1)$ -Graphen und der $(q, q-3)$ -Graphen.*

Beweis. Abbildung 4.3 zeigt eine der beiden transitiven Orientierungen. Jeder Knoten kann als erster Knoten eines induzierten P_4 angesehen werden. Der Kreis besitzt also auf $q+1$ Knoten genau $q+1$ P_4 s. Jeder induzierte Teilgraph auf q Knoten ist ein P_q , der nach Theorem 3.11 genau $q-3$ P_4 s besitzt. \square

Abbildung 4.3: Transitive Orientierung eines Kreises C_{q+1} gerader Länge

Das Komplement eines C_4 sind zwei parallele Ketten der Länge zwei, also ein Komparabilitätsgraph.

Satz 4.3 (Dil50,BC84) *Das Komplement \overline{C}_n eines Kreises mit Länge $n \geq 5$ ist kein Komparabilitätsgraph.*

4.3 Pfade

Eine Partialordnung, deren zugrundeliegender Komparabilitätsgraph ein Pfad P_n der Länge $n - 1$ ist, heißt in der Literatur *Zick-Zack* oder *Zaun*.

Abbildung 4.4 (bzw. 4.5) zeigt eine der beiden transitiven Orientierungen eines \vec{P}_n (bzw. $\overline{\vec{P}}_n$) gerader Länge.

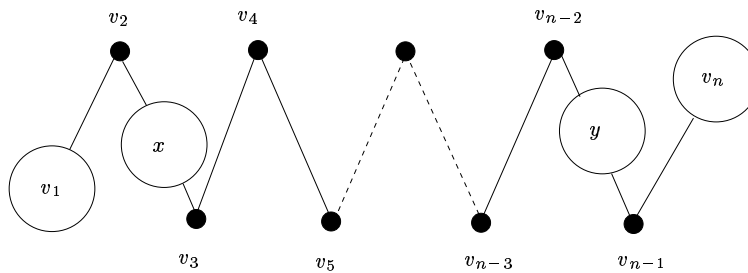
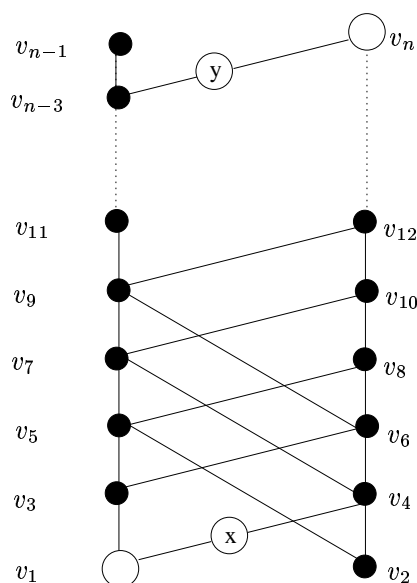


Abbildung 4.4: Hasse-Diagramm eines \vec{P}_n gerader Länge

Lemma 4.4 \vec{P}_n und $\overline{\vec{P}}_n$ sind Komparabilitätsgraphen.

Nur Pfade der Länge drei, d.h. Spinnen auf vier Knoten, sind separabel (siehe 4.1).

Der spezielle Fall, in dem in einem P_4 ein Knoten durch einen Cograph ersetzt ist, entspricht einer Verallgemeinerung der erweiterten (und gewöhnlichen \bowtie_g) Spinnenkomposition. D.h. einer der Knoten darf nicht nur durch eine zweielementige Kette oder Antikette ersetzt werden, sondern durch eine beliebige serien-parallele Partialordnung. Wir sprechen deshalb im folgenden von der *verallgemeinerten Spinnenkomposition* $P \bowtie_v \{A, B, C, D\}$.

Abbildung 4.5: Hasse-Diagramm eines \overline{P}_n

4.4 Spinnennetze

Lemma 4.5 \vec{Q}_n sind Komparabilitätsgraphen.

Beweis. (Vollständige Induktion) Ein $\vec{Q}_4 = P_4$ ist transitiv orientierbar. Da ein \vec{Q}_{n+1} einen \vec{Q}_n als induzierten Teilgraphen enthält, bleibt zu zeigen, daß die inzidenten Kanten der zwei hinzugefügten Knoten v_{n+1} und z_{n-4} transitiv orientierbar sind. Wir erinnern daran, daß im \vec{Q}_n die aufeinanderfolgenden Knoten $v_i, v_{i+1}, v_{i+2}, v_{i+3}$ und $z_{i-2}, v_{i-1}, v_i, v_{i+1}$, $i \leq n-3$, je einen P_4 induzieren. In \vec{Q}_{n+1} entstehen zwei neue P_4 s: $v_{n-2}v_{n-1}v_nv_{n+1}$ und $z_{n-4}v_{n-3}v_{n-2}v_{n-1}$. Ist n ungerade, gilt entweder $(v_{n-3}||v_{n-2}) \rightarrow v_{n-1}$ oder $v_{n-1} \rightarrow (v_{n-3}||v_{n-2})$ in \vec{Q}_n . Damit liegen weiterhin alle Kanten in \vec{Q}_{n+1} in einer Λ^* -Klasse und die inzidenten Kanten von v_{n+1} und z_{n-4} sind transitiv orientierbar. Grob gesprochen, muß v_{n+1} , falls $n+1$ gerade ist, in die Kette der geraden Knoten eingefügt werden (siehe Abbildung 4.6, die hinzugefügten Knoten sind kariert, die dazu inzidenten Kanten fett).

Ist $n+1$ ungerade, folgt mit derselben Argumentation, daß \vec{Q}_{n+1} transitiv orientierbar ist und v_{n+1} in die Antikette der ungeraden Knoten eingefügt wird. \square

Weil die geraden Knoten eines angespitzten Spinnennetzes nach Definition eine Clique und die ungeraden Knoten unabhängig sind, bilden diese Mengen

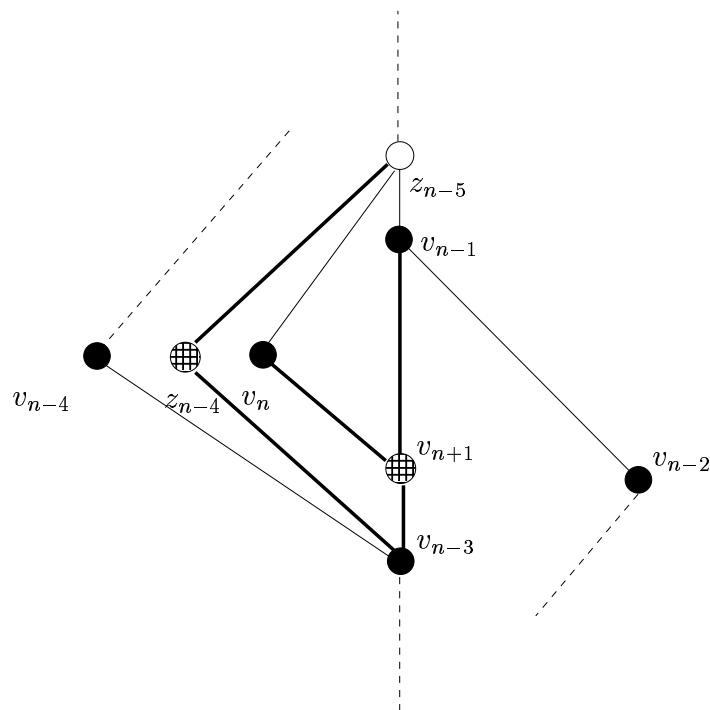
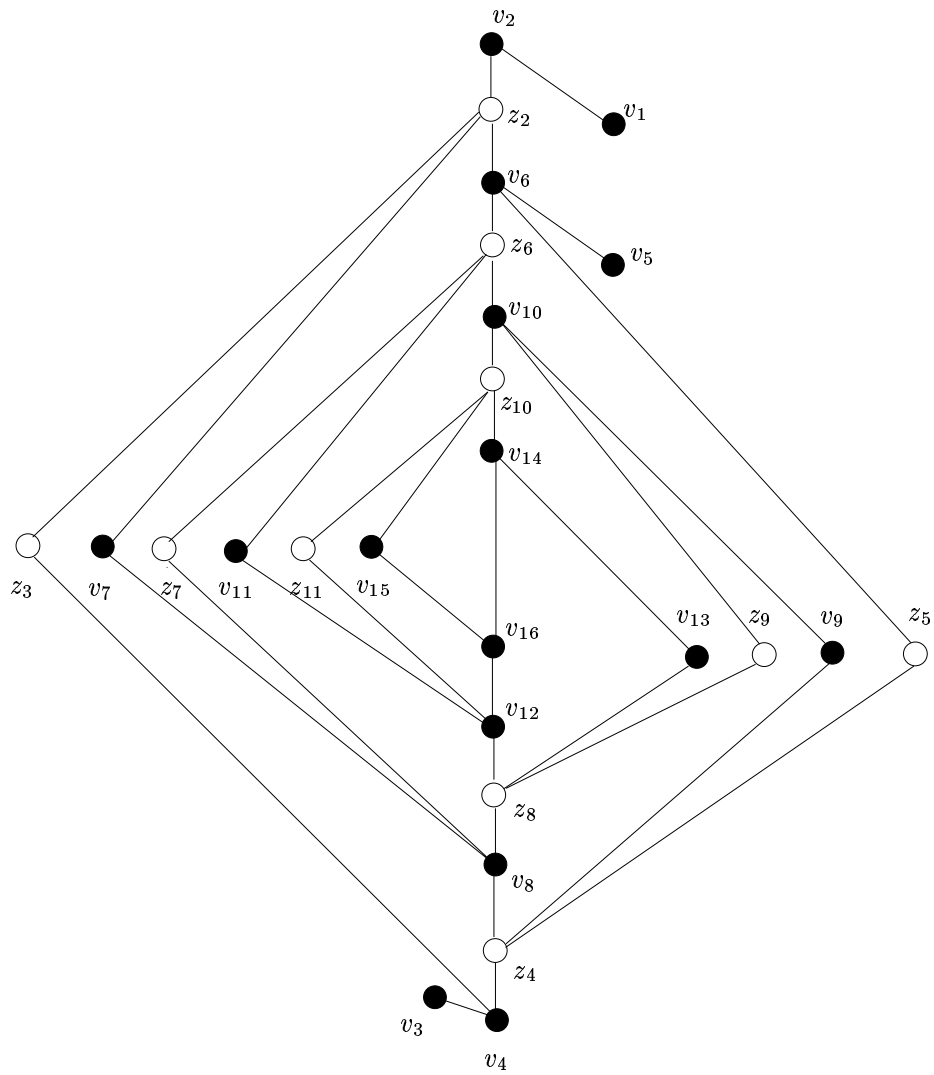


Abbildung 4.6: Induktionsschritt zu Lemma 4.5 von n auf $n+1$, $n+1$ gerade

in der assoziierten Partialordnung eine Kette bzw. Antikette (in den Komplementen umgekehrt).

Abbildung 4.7 zeigt eine der beiden transitiven Orientierungen des \vec{Q}_{16} . Die Knoten des induzierten Q_{16} sind ausgefüllt.

Abbildung 4.7: Hasse-Diagramm des \vec{Q}_{16}

Für deren Komplemente zeigt sich:

Satz 4.6 $\overline{\vec{Q}}_{n+1} \setminus \{v_1, z_2\}$ ist isomorph zu \vec{Q}_n .

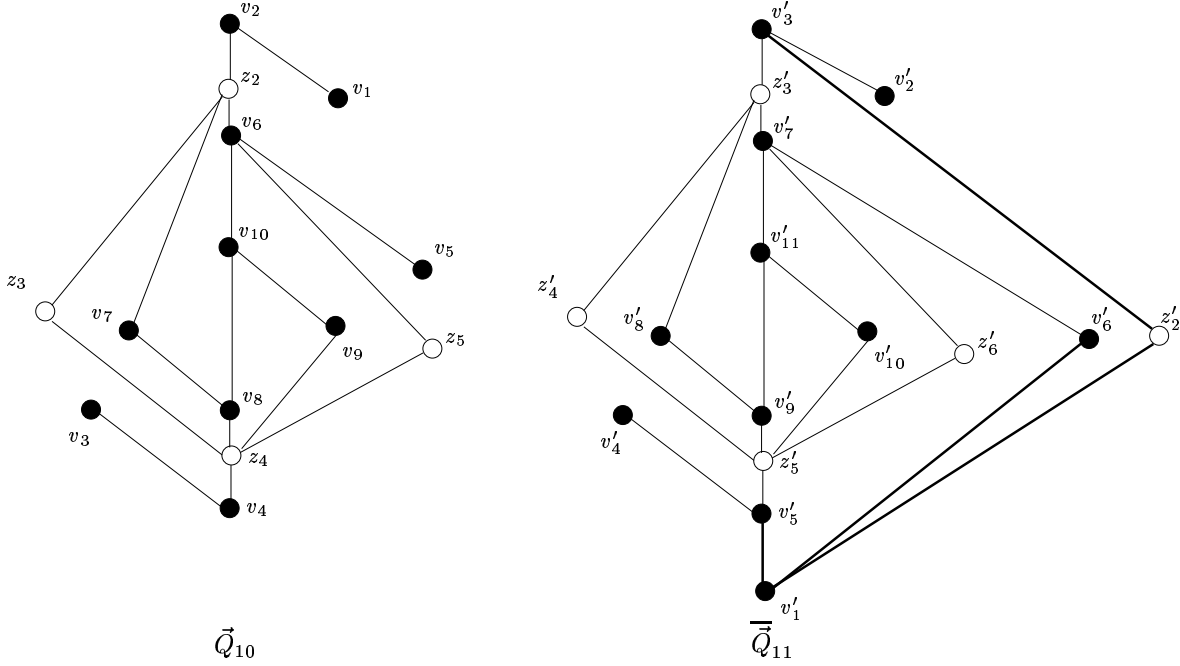


Abbildung 4.8: Q_{10} und \overline{Q}_{11}

Beweis. (vgl. Beispiel in Abbildung 4.8) Ein \vec{Q}_n ist definiert durch:

$$\begin{aligned}
 N(v_{2i-1})^+ &= \{v_{2i}\} && \text{für } i \geq 1, \\
 \overline{N}(v_{2i})^+ &= \{v_{2i+1}\} && \text{für } i \geq 1, \\
 N(z_i) &= \{v_2, v_4, \dots, v_{i+1}\} \cup \{z_2, z_4, \dots, z_{i-1}\} && \text{für } i \text{ ungerade und} \\
 \overline{N}(z_i) &= \{v_1, v_3, \dots, v_{i+1}\} \cup \{z_3, z_5, \dots, z_{i-1}\} && \text{für } i \text{ gerade.}
 \end{aligned}$$

Also ist $\overline{\vec{Q}}_{n+1}$ definiert durch:

$$\begin{aligned}
 \overline{N}(v_{2i-1})^+ &= \{v_{2i}\} && \text{für } i \geq 1, \\
 N(v_{2i})^+ &= \{v_{2i+1}\} && \text{für } i \geq 1, \\
 \overline{N}(z_i) &= \{v_2, v_4, \dots, v_{i+1}\} \cup \{z_2, z_4, \dots, z_{i-1}\} && \text{für } i \text{ ungerade und} \\
 N(z_i) &= \{v_1, v_3, \dots, v_{i+1}\} \cup \{z_3, z_5, \dots, z_{i-1}\} && \text{für } i \text{ gerade.}
 \end{aligned}$$

Sei $\vec{Q}_n := (V \cup Z, E)$ und $\overline{\vec{Q}}_{n+1} := (V' \cup Z', E')$. Dann ist

$$\begin{array}{rcl} \varphi : V \cup Z & \rightarrow & (V' \cup Z') \setminus \{v'_1\} \quad \text{mit} \\ v_i & \rightarrow & v'_{i+1} \quad \text{für } i \geq 1 \text{ und} \\ z_i & \rightarrow & z'_{i+1} \quad \text{für } 2 \leq i \leq n-5 \end{array}$$

eine Bijektion auf den Knotenmengen mit $xy \in E \Leftrightarrow \varphi(x)\varphi(y) \in E'$ für alle $x, y \in V$. \square

Nun sind angespitzte Spinnennetze \vec{Q}_n und deren Komplemente separabel. Sei $H = (H_1, H_2)$ die Separation eines angespitzten Spinnennetzes \vec{Q}_n , wobei H_1 die Menge aller geraden Knoten und H_2 die Menge aller ungeraden ist. Sei ferner $H_1 = H_1^{top} \cup H_1^{bot}$ mit

$$\begin{aligned} H_1^{top} &:= \{v_i \in \vec{Q}_n \mid i-2 \text{ ist teilbar durch } 4\} \text{ und} \\ H_1^{bot} &:= \{v_i \in \vec{Q}_n \mid i \text{ ist teilbar durch } 4\} . \end{aligned}$$

Anschaulich handelt es sich bei H_1^{top} (bzw. H_1^{bot}) um die „oberen“ (bzw. „unteren“) $\lceil \frac{n}{4} \rceil$ Elemente in der Kette der geraden Knoten im Hasse-Diagramm des \vec{Q}_n . Abbildung 4.9 veranschaulicht dies am Beispiel des Spezialfalls eines „gewöhnlichen“ Spinnennetzes. Für die andere transitive Orientierung vertauscht man H_1^{bot} und H_1^{top} .

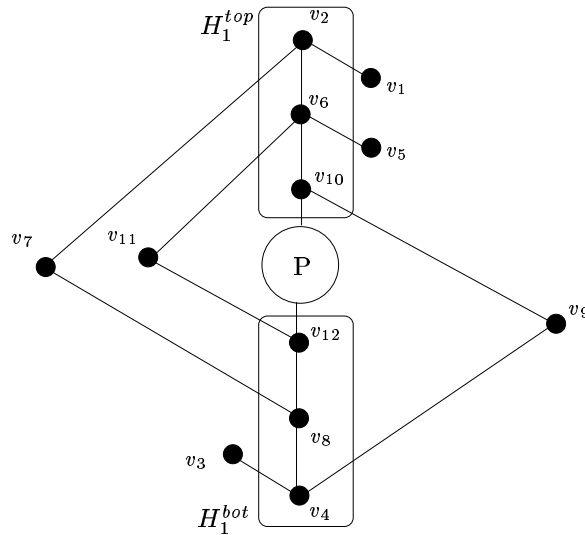


Abbildung 4.9: Spinnennetzkomposition $P \star Q_{12}$

Dann definieren wir für \vec{Q}_n und ein beliebiges $(q, q-3)$ -Poset P die *Spinnennetzkomposition* $P \star \vec{Q}_n$ durch:

$$u < v :\Leftrightarrow \begin{cases} u < v \text{ und } u, v \in P, \\ u \in P \text{ und } v \in H_1^{top}, \\ v \in P \text{ und } u \in H_1^{bot}. \end{cases}$$

Analog wird die Spinnennetzkomposition $P \star \overline{Q}_n$ für Komplemente angespitzter Spinnennetze \overline{Q}_n definiert.

4.5 Graphen beschränkter Grösse

p -zusammenhängende $(q, q - 3)$ -Graphen können ferner Graphen beschränkter Grösse sein. Sie besitzen dann maximal $q - 1$ Knoten, die höchstens $q - 3$ Pfade der Länge drei induzieren.

Die bekannten (kleinen) angespitzten p -Ketten R_5, R_6, R_7 und deren Komplemente (vgl. Seite 31) sind alle Komparabilitätsgraphen. Im R_5 (bzw. \overline{R}_5) bildet die Clique (bzw. stabile Menge) $\{v_1, v_2\}$ eine maximale homogene Menge. Der R_5 besitzt deshalb zwei Λ^* -Klassen, alle anderen eine. Je eine transitive Orientierung ist in Abbildung 4.10 dargestellt.

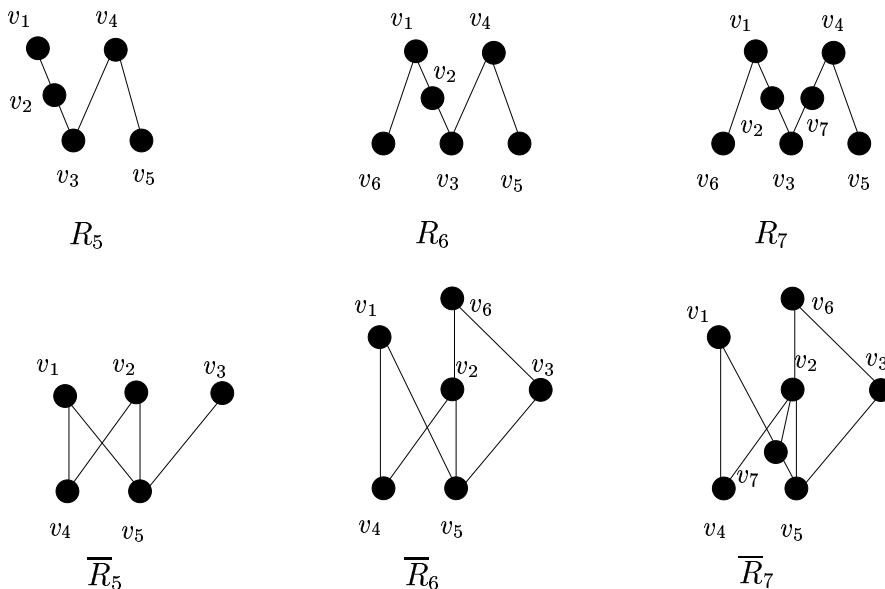


Abbildung 4.10: Hasse-Diagramme kleiner angespitzter p -Ketten

Nur der R_5 und sein Komplement \overline{R}_5 sind separabel. Sie sind schon als N -erweiterbare Partialordnungen (siehe Seite 13) bekannt.

Die ersten drei kleinen p -Kreise aus Abbildung 3.3 sind Komparabilitätsgraphen. Nur (c) ist separabel und bekannt als P_4 bei dem ein Knoten durch eine Kette der Länge vier ersetzt ist.

Natürlich können weitere p -zusammenhängende und separabel p -zusammenhängende $(q, q-3)$ -Graphen mit weniger als q Knoten existieren, die transitiv orientierbar sind.

Wir definieren die Δ -Komposition. Sei $G(H)$ der p -zusammenhängende Komparabilitätsgraph einer Partialordnung H , der $k < q$ Knoten besitzt, die höchstens $q-3$ Pfade der Länge drei induzieren und P ein beliebiges $(q, q-3)$ -Poset.

Ist $G(H)$ separabel, dann enthalte H_1 wie üblich alle Mittelknoten und H_2 alle Endknoten der Separation (H_1, H_2) . P sei nun adjazent zu allen Mittelknoten H_1 und zu keinem Knoten in H_2 . Dann ist dies eine $(q, q-3)$ -Partialordnung, denn es existiert kein P_4 , der Knoten sowohl in H und P besitzt (siehe Seite 35) und der so konstruierte Graph ist transitiv orientierbar. Abbildung 4.11 veranschaulicht die Komposition anhand einer $(6, 4)$ -Partialordnung, die im besonderen zur Klasse der $(7, 4)$ -Posets gehört.

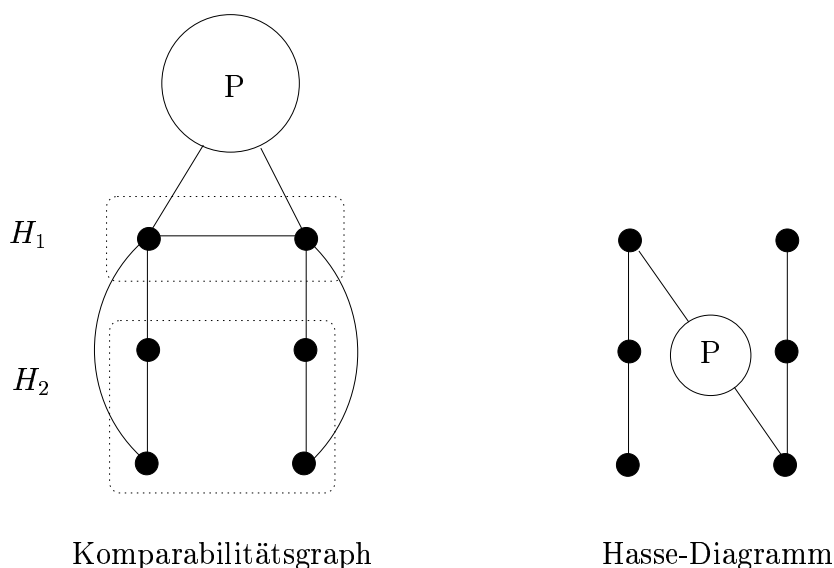
Wir schreiben abkürzend für die Komposition $P\Delta H$.

Anders formuliert: ersetzt man P in $P\Delta H$ durch einen einzelnen Knoten j , dann ist $j\Delta H$ äußerer Prim-Faktor mit höchstens q Knoten. Der zugrundeliegende Komparabilitätsgraph $G(j\Delta H)$ ist separabel p -zusammenhängend. Der Einfachheit halber fassen wir auch den nicht-separablen Fall von $G(H)$, d.h. mögliche Komponenten einer $(q, q-3)$ -Partialordnung, als „leere“ Komposition $\emptyset\Delta H$ auf.

Zwei spezielle Δ -Kompositionen sind schon bekannt.

Die gewöhnliche Spinnenkomposition $P \bowtie_g \{a, b, c, d\}$ (siehe Seite 14) entspricht $P\Delta H$ mit $G(H) := P_4$, also $|H| = 4$ und H ist eine $(q, q-3)$ -Partialordnung für alle $q \geq 4$.

Die erweiterte Spinnenkomposition $P \bowtie_e \{A, B, C, D\}$ (siehe Seite 13) entspricht $P\Delta H$, wobei $G(H)$ ein P_4 ist, in dem ein Knoten durch eine zweielementige Kette oder Antikette ersetzt ist, also $|H| = 5$ und H ist eine $(q, q-3)$ -Partialordnung für alle $q \geq 4$.

Abbildung 4.11: $P\Delta H$

Wir weisen darauf hin, daß $(q, q - 3)$ -Partialordnungen für festes $q \geq 7$ alle transitiv orientierbaren Graphen der Grösse vier und fünf beinhalten. Denn jeder Komparabilitätsgraph mit vier Knoten induziert maximal ein N und der einzige Graph, der auf fünf Knoten mehr als zwei N induziert, ist der Kreis der Länge fünf, der keine transitive Orientierung besitzt.

Aus den Ergebnissen dieses Abschnitts ist nun eine Konstruktionsvorschrift für $(q, q - 3)$ -Partialordnungen ableitbar. Aufgrund des Strukturtheorems 2.1 sind keine weiteren nicht-serien-parallelen Kompositionen möglich.

Fortan bezeichnen $\vec{P}_n, \overline{\vec{P}}_n, \vec{Q}_n, \overline{\vec{Q}}_n$ und C_n auch die mit diesen Graphen assoziierten Partialordnungen. Nicht-angespitzte Varianten (d.h. P_n, Q_n und deren Komplemente) sind Spezialfälle der angespitzten (d.h. \vec{P}_n, \vec{Q}_n und deren Komplementen).

Theorem 4.7 *Eine Partialordnung P ist eine $(q, q - 3)$ -Partialordnung mit fest vorgegebenen $q \geq 7$ genau dann, wenn sie aufgebaut werden kann aus*

- einzelnen Knoten,
- Kreisen C_n mit gerader Länge, mit $n > q$, und

- angespitzten Pfaden \vec{P}_n und deren Komplementen \overline{P}_n , $n \geq q$, in denen p -Endknoten durch serien-parallele Ordnungen ersetzt sein können

mittels

- serieller \rightarrow und
- paralleler $||$ Kompositionen,
- verallgemeinerten Spinnenkompositionen \bowtie_v ,
- Spinnennetzkompositionen \star und
- Kompositionen $P' \Delta H$, wobei H weniger als q Knoten und nicht mehr als $q - 3$ induzierte N s besitzt und P' beliebiges $(q, q - 3)$ -Poset oder leer ist.

Für $(q, q - 4)$ -Partialordnungen ergibt sich durch die Restriktion auf Komparabilitätsgraphen eine handliche Charakterisierung.

Lemma 4.8 *Ist $G = (V, E)$ ein p -zusammenhängender $(q, q - 4)$ -Komparabilitätsgraph, $q \geq 4$, dann ist $|V| < q$ oder G ist ein P_4 .*

Beweis. Satz 3.4 und Lemma 4.1. □

Sie lassen sich dementsprechend weitaus einfacher aufbauen als $(q, q - 3)$ -Posets.

Korollar 4.9 *Für festes $q \geq 4$ lassen sich $(q, q - 4)$ -Partialordnungen konstruktiv aufbauen, durch*

- einzelne Elemente

mittels

- Serienkompositionen \rightarrow ,
- Parallelkompositionen $||$ und
- Kompositionen $P' \Delta H$, wobei H weniger als q Knoten und nicht mehr als $q - 4$ induzierte N s besitzt und P' beliebiges $(q, q - 4)$ -Poset oder leer ist.

Da die gewöhnliche \bowtie_q und die erweiterte Spinnenkomposition \bowtie_e spezielle Δ -Kompositionen sind, ist deutlich, daß $(q, q - 4)$ -Partialordnungen eine Verallgemeinerung N -dünner und N -erweiterbarer Posets sind. Da ihre Dekompositionsweite stets durch q (genauer: $q - 2$, da mindestens ein N existiert) beschränkt ist, bilden sie eine Teilmenge der Klasse aller Posets mit beschränkter Dekompositionsweite. Die Relation ist echt, da sämtliche äußeren Faktoren einer $(q, q - 4)$ -Partialordnung höchstens $q - 4$ induzierte N s besitzen dürfen.

Kapitel 5

1 $|(q, q-3)| \sum w_j C_j$

In diesem Kapitel wird ein effizienter Algorithmus für das Ein-Maschinen-Durchlaufzeitproblem mit Reihenfolgebedingungen, die durch eine $(q, q-3)$ -Partialordnung mit festem $q \geq 7$ gegeben sind, entwickelt.

Durch Theorem 4.7 kann eine $(q, q-3)$ -Partialordnungen als Baum beschrieben werden, der die Dekomposition des Strukturtheorems widerspiegelt. Die Blätter entsprechen den p -zusammenhängenden Komponenten. Sie haben weniger als q Knoten, sind im besonderen einelementig, Kreise, (angespitzte) Pfade oder (angespitzte) Spinnennetze oder die Komplemente der beiden letzteren. Möglicherweise sind p -Endknoten in p -Bäumen durch serien-parallele Ordnungen ersetzt. Die inneren Knoten sind markiert mit \rightarrow , \parallel , \bowtie_v , \star oder Δ .

Wir nutzen nun Lawlers Idee und arbeiten den Baum von unten nach oben ab. Da die inneren Knoten und die Blätter Module in der Partialordnung induzieren, existiert nach Theorem 2.7 eine optimale Lösung des Gesamtproblems, die konsistent ist mit den optimalen Lösungen der induzierten Teilprobleme. Ist eine optimale Lösung für einen Knoten berechnet, ersetzt man diesen durch eine Kette. Für einen inneren Knoten, der seriell \rightarrow oder parallel \parallel markiert ist, wird Lawlers Algorithmus angewendet. Er wird auch für p -Endknoten verwendet, die durch serien-parallele Ordnungen ersetzt sind, da einzelne Knoten offensichtlich Module sind. Gesucht sind Algorithmen, die die weiteren durch die inneren Knoten und Blätter induzierten Teilprobleme effizient lösen.

Abschnitt 5.1 behandelt sämtliche pfadähnlichen $(q, q-3)$ -Partialordnungen. Wir entwickeln einen dynamischen Programmierungsalgorithmus, der als Spezialfall das Durchlaufzeitproblem über angespitzten Pfaden \vec{P}_n löst. Basis des

Verfahrens bildet die einfache Beobachtung, daß das Festhalten eines Jobs im \vec{P}_n , der keinen Nachfolger hat, die Partialordnung in zwei parallele Module, deren optimale Lösungen schon vorab berechnet wurden, zerfallen läßt.

Das Verfahren ist direkt übertragbar auf Kreise C_n und als Variante am Ende des Abschnitts auf die verallgemeinerte Spinnenkomposition \bowtie_n .

Zur Lösung des Durchlaufzeitproblems über Komplementen von angespitzten Pfaden $\overline{\vec{P}}_n$ ziehen wir den bekannten Algorithmus von Sidney und Steiner (siehe Seite 20) heran.

Im Abschnitt 5.2 widmen wir uns zunächst Spinnennetzen \vec{Q}_n . Auch dieser Algorithmus nutzt das Prinzip der dynamischen Programmierung. Die Kernidee besteht (wie beim Pfadalgorithmus) darin, daß sich größere Teilprobleme auf schon vorab berechnete kleinere Teilprobleme zurückführen lassen, in dem die Position bestimmter Elemente fixiert wird. Essentiell ist, daß diese Elemente in jeder beliebigen linearen Erweiterung nur an wenigen möglichen – genauer: zwei – Positionen stehen können.

Am Ende des Abschnitts wird das Ergebnis auf sämtliche weiteren Spinnennetze und die Spinnennetzkompositionen übertragen.

Ist das Durchlaufzeitproblem für eine Partialordnung zu lösen, deren äußerer Faktor $j\Delta H$ höchstens q Knoten und H maximal $q-3$ induzierte N s besitzt, greifen wir wiederum auf das dynamische Verfahren von Sidney und Steiner zurück. Aufgrund der Wichtigkeit der benötigten Laufzeit halten wir diese hier explizit fest.

Satz 5.1 *Sei $|j\Delta H| = q$ und $\check{n} := |P\Delta H|$. Dann ist $1|P\Delta H| \sum w_j C_j$ lösbar in einer Laufzeit von $O(\check{n}^{q-1})$, sofern eine optimale lineare Erweiterung für P bekannt ist.*

Beweis. Die Dekompositionsweite von $P\Delta H$ ist gleich der Weite von $j\Delta H$. $j\Delta H$ hat höchstens Weite $q-2$, denn in H muß mindestens ein induziertes N existieren. Die Behauptung folgt mit Satz 2.12. \square

Theorem 5.15 am Ende des Kapitels fasst die Ergebnisse zusammen.

Im gesamten Kapitel bezeichnen wir die Menge aller Jobs j_i mit $s \leq i \leq t$ als *Intervall* $[s, t]$, $1 \leq s \leq t \leq n$. Für $s > t$ sei $[s, t] := \emptyset$. Die *Intervallgröße* von $[s, t]$ ist einfach die Anzahl der Jobs in $[s, t]$.

Ist $P := (J, <)$ eine beliebige Partialordnung, dann ist $P([s, t])$ die durch die Knotenmenge $[s, t] \subseteq J$ induzierte Teilpartialordnung. $P^z[s, t]$ bezeichnet die Partialordnung $P([s, t])$ mit der zusätzlichen Einschränkung, daß $z \in [s, t]$

letzter Job ist. $Opt(P)$ sei ein optimaler Zielfunktionswert des Durchlaufzeitproblems für P und eine dazugehörige lineare Erweiterung bezeichnen wir mit $Opt_\pi(P)$.

5.1 Pfadähnliche $(q, q-3)$ -Posets

Unter pfadähnlichen $(q, q-3)$ -Partialordnungen sind angespitzte Pfade \vec{P}_n , deren Komplemente \overleftarrow{P}_n und Kreise gerader Länge C_n zusammengefasst, $n \geq 4$. Sie sind unzerlegbar und haben Weite $\lceil \frac{n}{2} \rceil$.

Wir beginnen mit angespitzten Pfaden. Statt diese direkt zu betrachten, geben wir zunächst einen polynomiellen Algorithmus für den allgemeineren Fall an, in dem in der Partialordnung P_n die beiden p -Endknoten j_1 und j_n durch serien-parallele Ordnungen ersetzt und zusätzlich $n-1$ serien-parallele Ordnungen z_1, \dots, z_{n-1} hinzugefügt sein können und zwar so, daß für $1 \leq i \leq n-1$ gilt:

$$\begin{aligned} j_i &\rightarrow z_i \rightarrow j_{i+1} \text{ für } i \text{ ungerade und} \\ j_{i+1} &\rightarrow z_i \rightarrow j_i \text{ für } i \text{ gerade (siehe Abbildung 5.1).} \end{aligned}$$

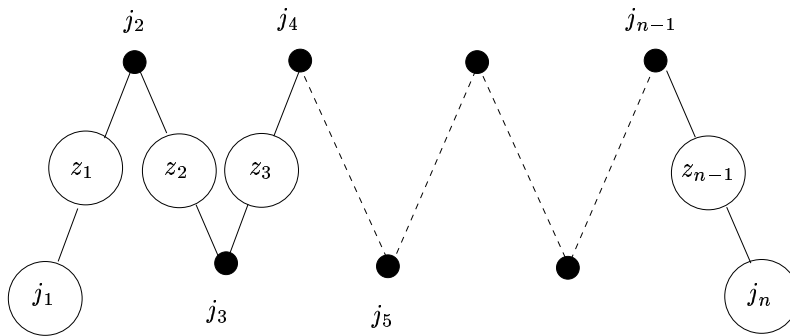


Abbildung 5.1: Hasse-Diagramm des \mathbb{P}_n ungerader Länge

Wir nennen diese Graphen *verallgemeinerte angespitzte Pfade*, kurz \mathbb{P}_n . Mit $z_0 := j_1$ und $z_n := j_n$ ist die Anzahl der Knoten $\check{n} := (n-2) + \sum_{i=0}^n |z_i|$.

$\mathcal{L}(P)$ bezeichnet wie üblich für eine Partialordnung P die Menge aller linearen Erweiterungen von P und $Max(P)$ die Menge aller Jobs aus P , die keinen Nachfolger in P besitzen.

Zunächst werden die z_i durch die mit Lawlers Algorithmus berechneten optimalen Ketten ersetzt (siehe Seite 51). Für unsere Zwecke numerieren wir anschließend die Knoten aufsteigend von j_1 bis j_n um (siehe Abbildung 5.2).

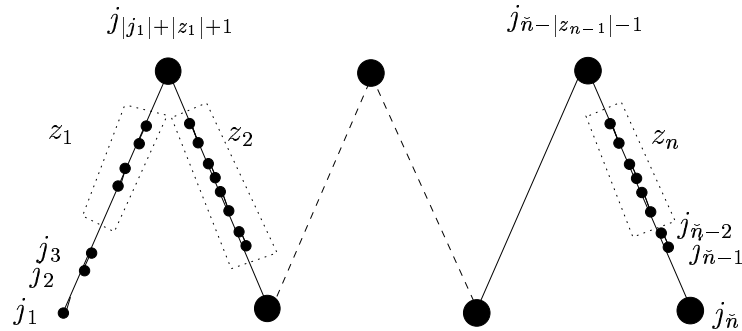


Abbildung 5.2: Umnummerierung nach Ersetzen durch optimale Ketten

Für jedes Poset P gilt trivialerweise, daß in jeder linearen Erweiterung einer der Jobs aus $Max(P)$ als letzter abgearbeitet werden muß, d.h.

$$\mathcal{L}(P) = \bigcup_{z \in Max(P)} \mathcal{L}(P) . \quad (*)$$

Zu gegebenen Prozeßzeiten $p_1, \dots, p_{\tilde{n}}$ und Gewichten $w_1, \dots, w_{\tilde{n}}$ funktioniert der Algorithmus für das Durchlaufzeitproblem mit dynamischer Programmierung wie folgt.

Angenommen wir kennen $Opt(\mathbb{P}_n([s', t']))$ für alle Intervalle $[s', t']$ mit Kardinalität kleiner $l, l \leq n$. Dann berechnet man für ein Intervall $[s, t]$ mit Intervalllänge l $Opt(\mathbb{P}_n([s, t]))$, indem man $Opt(\mathbb{P}_n^z([s, t]))$ für alle $z \in Max(\mathbb{P}_n([s, t]))$ bestimmt und den kleinsten dadurch gegebenen Zielfunktionswert wählt. Dies ist effizient möglich, da man durch das Festhalten das Problem auf das Verschmelzen zweier Ketten kleinerer Intervallgröße reduziert.

Abbildungen 5.3 und 5.4 veranschaulichen das Prinzip für das Intervall $[s, t]$, in dem einer der drei letzten Jobs j_r festgehalten ist.

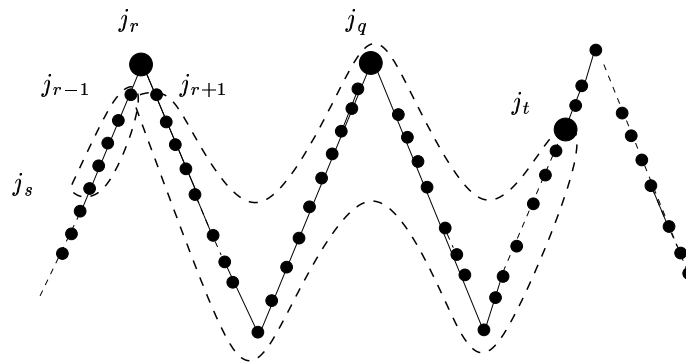


Abbildung 5.3: $\mathbb{P}_n([s, r-1])$ und $\mathbb{P}_n([r+1, t])$

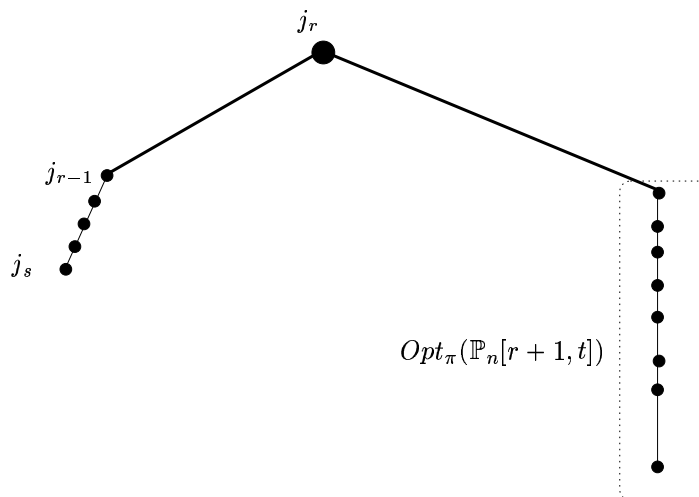


Abbildung 5.4: $\mathbb{P}_n^{j_r}([s, t])$

Lemma 5.2 Sei l die Intervallgröße von $[s, t]$ und $z \in Max(\mathbb{P}_n([s, t]))$ fest. Dann kann $Opt(\mathbb{P}_n^z([s, t]))$ in einer Laufzeit von $O(l \log l)$ berechnet werden, sofern $Opt_\pi(\mathbb{P}_n([s, z-1]))$ und $Opt_\pi(\mathbb{P}_n([z+1, t]))$ bekannt sind.

Beweis. Da $[s, z-1]$ und $[z+1, t]$ Module in $\mathbb{P}_n^z([s, z-1])$ sind, können diese mit Lemma 2.7 durch ihre nach Voraussetzung bekannten optimalen Ketten ersetzt werden. Dann ist $\mathbb{P}_n^z([s, t])$ serien-parallel. Die Behauptung folgt mit Lawlers Algorithmus nach Satz 2.9. \square

Der Algorithmus ist auf der nächsten Seite explizit angegeben.

Algorithmus 1 für $1|\mathbb{P}_n|\sum w_j C_j$ **Eingabe:** Prozeßzeiten $p_1, \dots, p_{\check{n}}$ und Gewichte $w_1, \dots, w_{\check{n}}$,**Ausgabe:** Eine optimale lineare Erweiterung, die $\sum w_j C_j$ minimiertfunction lawler(P)

bestimme mit Lawlers Algorithmus optimale
lineare Erweiterung $Opt_\pi(P)$ für $\min \sum_{j \in P} w_j C_j$;

return $Opt_\pi(P)$;function CT(π)return $\sum w_j C_j$ für die lineare Erweiterung π ;1. Ersetzung von z_0, \dots, z_n für alle z_i

{

Ersetze z_i durch lawler(z_i);Aktualisiere Numerierung der Knoten in z_i ;

}

2. Berechnung der Intervalle

für ($i = 1, i \leq \check{n}, i++$)für ($s = 1, s \leq \check{n}, s++$)

{

wenn ($s + i \leq \check{n}$)für $z \in Max(\mathbb{P}_n([s, s + i]))$

{

 $Opt_\pi(\mathbb{P}_n^z([s, s + i])) :=$ lawler($(Opt_\pi(\mathbb{P}_n([s, z-1])) \parallel Opt_\pi(\mathbb{P}_n([z+1, s+i]))) \rightarrow z$);

}

 $z_{min} := \min \left\{ CT \left(Opt_\pi(\mathbb{P}_n^z([s, s + i])) \right) \mid z \in Max(\mathbb{P}_n([s, s + i])) \right\}$;

}

 $Opt_\pi(\mathbb{P}_n([s, s + i])) := Opt_\pi(\mathbb{P}_n^{z_{min}}([s, s + i]));$

Theorem 5.3 *Algorithmus 1 löst das Durchlaufzeitproblem über verallgemeinerten Pfaden \mathbb{P}_n , $\check{n} := |\mathbb{P}_n|$, in einer Laufzeit von $O(\check{n}^4 \log \check{n})$.*

Beweis. Wegen (*) berechnet Algorithmus 1 optimale Lösungen. Das Ersetzen der z_i benötigt mit Lawler $O(n \cdot \check{n} \log \check{n})$. In 2. existieren $O(\check{n}^2)$ ($= \sum_1^{\check{n}} i$) viele Intervalle $[s, t]$ mit Intervallgröße $O(\check{n})$, für das jeweils $O(\check{n})$ mögliche Kandidaten an letzter Stelle stehen können. Mit Lemma 5.2 ergibt dies eine Gesamtlaufzeit von:

$$O(n \cdot \check{n} \log \check{n}) + O(\check{n}^2 \cdot \check{n} \cdot \check{n} \log \check{n}) = O(\check{n}^4 \log \check{n}). \quad \square$$

Da \vec{P}_n ein spezieller verallgemeinerter angespitzter Pfad \mathbb{P}_n ist, gilt:

Korollar 5.4 *Algorithmus 1 löst das Durchlaufzeitproblem über angespitzten Pfaden \vec{P}_n , $\check{n} := |\vec{P}_n|$, in einer Laufzeit von $O(\check{n}^4 \log \check{n})$.*

Das Verfahren für Kreise C_n verläuft analog zu Algorithmus 1. Da $O(n^2) = \sum_1^n n$ zyklische Intervalle existieren, folgt mit Theorem 5.3 dieselbe Größenordnung der Laufzeit wie bei verallgemeinerten Pfaden.

Korollar 5.5 *Das Durchlaufzeitproblem über Kreisen C_n gerader Länge ist lösbar in $O(n^4 \log n)$*

Wir fahren fort, das Durchlaufzeitproblem über Komplementen angespitzter Pfade $\overline{\vec{P}}_n$ zu lösen. Sind die p -Endknoten j_1, j_n, z_1 und z_2 durch ihre optimalen Ketten mit Lawlers Algorithmus ersetzt, hat $\overline{\vec{P}}_n$ höchstens Weite drei und das Problem läßt sich mit dem Algorithmus von Sidney und Steiner (siehe Seite 21) effizient lösen. Aus Satz 2.10 folgt:

Korollar 5.6 *Das Durchlaufzeitproblem über Komplementen von angespitzten Pfaden $\overline{\vec{P}}_n$, $\check{n} := |\overline{\vec{P}}_n|$, ist lösbar in $O(\check{n}^3)$.*

Zum Ende dieses Abschnitts betrachten wir die verallgemeinerte Spinnenkomposition $P \bowtie_v \{A, B, C, D\}$. Abbildung 5.5 zeigt eine solche, in der B durch eine serien-parallele Ordnung ersetzt ist.

Ist A, C oder D durch eine serien-parallele Ordnung ersetzt, löst Algorithmus 1 das Durchlaufzeitproblem effizient. Ist B ersetzt, können nur A oder C an erster Stelle stehen. Wir verwenden in diesen Fall Algorithmus 1 mit der Abänderung, statt letzter Elemente erste Elemente festzuhalten.

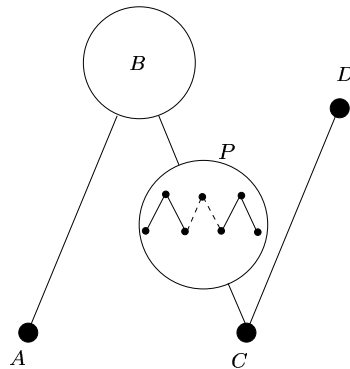


Abbildung 5.5: $P \bowtie_v \{A, B, C, D\}$

Korollar 5.7 Sei $\check{n} := |P| + |A| + |B| + |C| + |D|$.
 $1 \mid P \bowtie_v \{A, B, C, D\} \mid \sum w_j C_j$ ist mit Algorithmus 1 bzw. einer Variante lösbar in $O(\check{n} \log \check{n})$, vorausgesetzt für P ist eine optimale Lösung bekannt.

Beweis. Es müssen zwei Teilprobleme gelöst werden, die serien-parallel sind (vgl. Abbildung 5.6 für B serien-parallel). Die Behauptung folgt mit Lawlers Algorithmus. \square

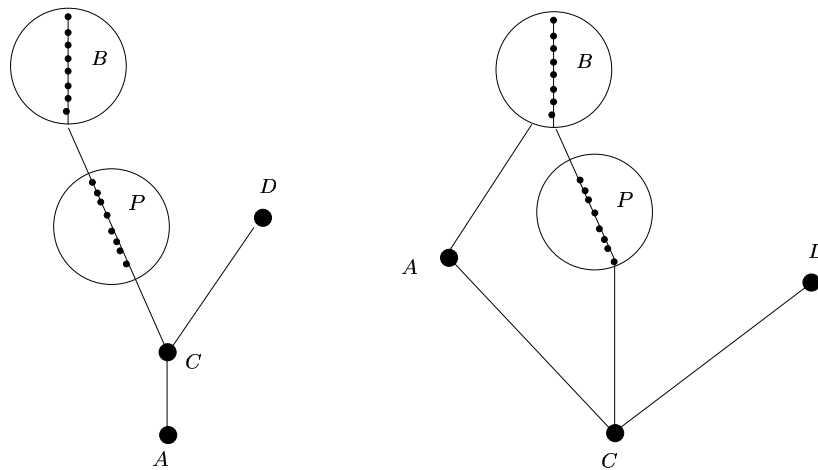


Abbildung 5.6: A oder C erstes Element

Da Partialordnungen $P \bowtie_v \{A, B, C, D\}$ Dekompositionsweite drei haben, verbessert Korollar 5.7 die Laufzeit von Sidney und Steiners Algorithmus von $O(\check{n}^4)$ aus Satz 2.12.

5.2 Spinnennetze

Wie Pfade sind auch Spinnennetze unzerlegbar und haben Dekompositionsweite $\lceil \frac{n}{2} \rceil$.

Sei o.B.d.A. $j_1 \rightarrow j_2$ in der mit \vec{Q}_n assoziierten Ordnung. Ferner sei $[z_k, z_{n-5}] := \{z_k, \dots, z_{n-5}\}$ für $2 \leq k \leq n-5$, $[z_0, z_{n-5}] := [z_2, z_{n-5}]$ und $[z_k, z_{n-5}] := \emptyset$ für $k > n-5$.

Alle z_i bzw. j_1 und j_n ersetzt man zunächst – wie bei angespitzten Pfaden – mittels Lawler durch optimale Ketten. In einer Spinnennetzkomposition $P \star \vec{Q}_n$ bzw. $P \star \overline{\vec{Q}}_n$ sei eine optimale lineare Erweiterung für P bekannt.

Anstatt direkt das Durchlaufzeitproblem mit Reihenfolgerestriktionen, die durch eine Spinnennetzkomposition gegeben sind, zu betrachten, lösen wir zuerst $1 \mid \vec{Q}_n \mid \sum w_j C_j$. Am Ende des Abschnitts übertragen wir das vorgestellte Verfahren auf die Kompositionen.

Die Idee des dynamischen Algorithmus für $1 \mid \vec{Q}_n \mid \sum w_j C_j$ basiert auf folgender einfachen Beobachtung, die das Durchlaufzeitproblem über \vec{Q}_n in zwei kleinere Teilprobleme zerlegt.

Betrachtet man ein angespitztes Spinnennetz genauer, entdeckt man, daß nur zwei Jobs j_2 und j_3 an letzter Position stehen können. Im letzten Fall muß j_2 an vorletzter Stelle stehen (siehe Abbildung 5.7), d.h.

$$Opt(\vec{Q}_n) = \min\{Opt(\vec{Q}_n^{j_2}), Opt(\vec{Q}_n^{j_3})\}.$$

Nun ist

$$\begin{aligned} \text{(a)} \quad \vec{Q}_n^{j_2} &= \vec{Q}_n \setminus \{j_2\} \rightarrow j_2 \\ &= \left(\vec{Q}_n([3, n] \cup [z_2, z_{n-5}]) \parallel j_1 \right) \rightarrow j_2 \quad \text{und} \\ \text{(b)} \quad \vec{Q}_n^{j_3} &= \vec{Q}_n \setminus \{j_2, j_3\} \rightarrow j_2 \rightarrow j_3 \\ &= \left(\vec{Q}_n([4, n] \cup [z_2, z_{n-5}]) \parallel j_1 \right) \rightarrow j_2 \rightarrow j_3 \quad . \end{aligned}$$

Angenommen wir kennen $Opt_\pi(\vec{Q}_n([3, n] \cup [z_2, z_{n-5}]))$ und $Opt_\pi(\vec{Q}_n([4, n] \cup [z_2, z_{n-5}]))$. Dann sind (a) und (b) serien-parallele Kompositionen und Lawlers Algorithmus führt zur effizienten Lösung der Teilprobleme und somit zu einer optimalen Lösung des Gesamtproblems $Opt(\vec{Q}_n)$.

Der dynamische Programmierungsalgorithmus geht wie folgt. Man berechnet nacheinander für fallendes k , $\lfloor \frac{n}{2} \rfloor - 1 \geq k \geq 1$,

- (1) $Opt_\pi \left(\vec{Q}_n([2k-1, n] \cup [z_{2k-2}, z_{n-5}]) \right)$ und
- (2) $Opt_\pi \left(\vec{Q}_n([2k, n] \cup [z_{2k-2}, z_{n-5}]) \right)$.

Diese Teilprobleme haben die Eigenschaft, für ungerades (gerades) k nur zwei Jobs j_{2k} und j_{2k+1} zu besitzen, die an letzter (erster) Position abgearbeitet werden können. Zum Beweis dieser Aussage für \vec{Q}_n , zeigen wir sie zunächst für „gewöhnliche“ Spinnennetze Q_n .

Lemma 5.8 $Q_n([2k-1, n])$ und $Q_n([2k, n])$ besitzen zwei mögliche Endelemente für k ungerade und zwei mögliche Anfangselemente für k gerade. Die beiden Jobs sind j_{2k} und j_{2k+1} , $2k+1 \leq n$.

Beweis. Wir zeigen, die Isomorphie der Komparabilitätsgraphen

$$G\left(Q_n([2k-1, n])\right) \simeq G\left(Q_{n-2k+2}\right) . \quad (**)$$

Sei $J := \{j_{2k-1}, \dots, j_n\}$ die Knotenmenge von $G\left(Q_n([2k-1, n])\right)$ und $J' := \{j'_1, \dots, j'_{n-2k+2}\}$ die Knotenmenge von $G\left(Q_{n-2k+2}\right)$. Dann ist $\varphi : J' \rightarrow J$ mit $\varphi(j'_i) = j_{2k-2+i}$ für $1 \leq i \leq n-2k+2$ eine kantenerhaltene Bijektion auf den Knotenmengen.

Analog gilt: $G\left(Q_n([2k, n])\right) \simeq G\left(Q_n([2, n-2k+2])\right)$.

Da jede mit $G\left(Q_{n-2k+2}\right)$ bzw. $G\left(Q_n([2, n-2k+2])\right)$ assoziierte Partialordnung je nach Orientierung zwei mögliche erste oder letzte Elemente j_2 und j_3 besitzt, folgt die Behauptung. \square

Satz 5.9 $\vec{Q}_n([2k-1, n] \cup [z_{2k-2}, z_{n-5}])$ und $\vec{Q}_n([2k, n] \cup [z_{2k-2}, z_{n-5}])$ besitzen zwei mögliche Endelemente für k ungerade und zwei mögliche Anfangselemente für k gerade. Die beiden Jobs sind j_{2k} und j_{2k+1} , $2k+1 \leq n$.

Beweis. Nach Definition des \vec{Q}_n sind alle $j \in [z_{2k-2}, z_{n-5}]$ überdeckt durch mindestens einen Job aus $[2k-1, n]$. Die Behauptung folgt mit Lemma 5.8. \square

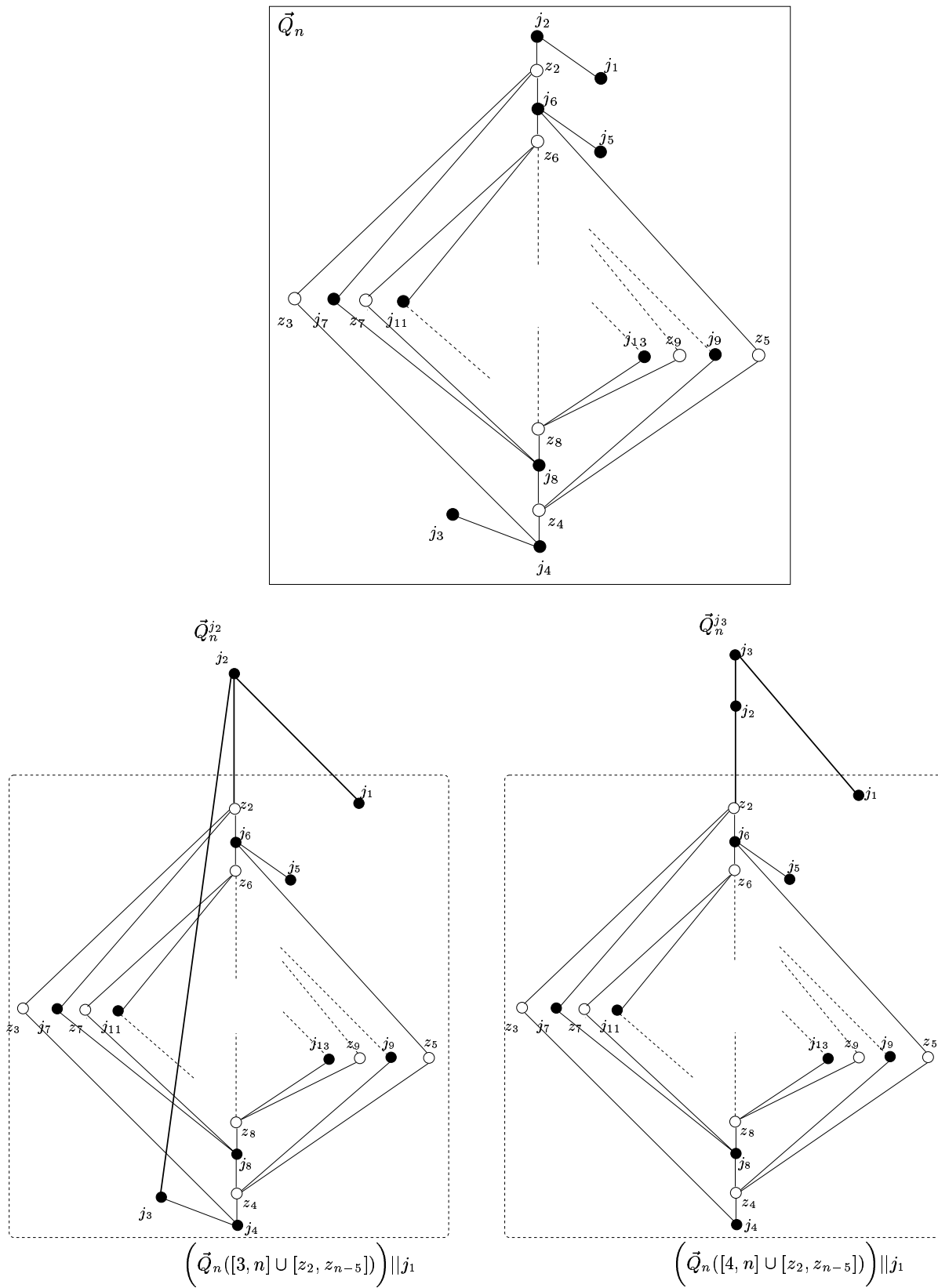


Abbildung 5.7: j_2 oder j_3 als letzte Jobs in \vec{Q}_n

Entscheidend dafür, daß sich größere Teilprobleme auf kleinere zurückführen lassen, ist zum einen, daß in beiden Fällen die Position von j_{2k} festgelegt wird. Denn steht j_{2k+1} an letzter (erster) Stelle, muß j_{2k} unmittelbarer Vorgänger (bzw. unmittelbarer Nachfolger) von j_{2k+1} sein. Zweitens steht j_{2k-1} in beiden Fällen parallel „zum Rest“, denn j_{2k-1} besitzt außer j_{2k} keine weiteren Nachbarn.

Lemma 5.10 *Die Berechnung von (1) bzw. (2) benötigt eine Laufzeit von $O(\check{n} \log \check{n})$, falls $Opt_\pi \left(\vec{Q}_n([2k+1, n] \cup [z_{2k}, z_{n-5}]) \right)$ und $Opt_\pi \left(\vec{Q}_n([2k+2, n] \cup [z_{2k}, z_{n-5}]) \right)$ bekannt sind.*

Beweis. Wir führen den Beweis für (1). Der Beweis für (2) verläuft analog. Sei k ungerade. Nach Lemma 5.9 existieren nur zwei mögliche Endelemente: j_{2k} und j_{2k+1} . Dann gilt:

$$(3) \quad \vec{Q}_n^{2k}([2k-1, n] \cup [z_{2k-2}, z_{n-5}]) = \left(\vec{Q}_n([2k+1, n] \cup [z_{2k-2}, z_{n-5}]) \parallel j_{2k-1} \right) \rightarrow j_{2k} ,$$

$$(4) \quad \vec{Q}_n^{2k+1}([2k-1, n] \cup [z_{2k-2}, z_{n-5}]) = \left(\vec{Q}_n([2k+2, n] \cup [z_{2k-2}, z_{n-5}]) \parallel j_{2k-1} \right) \rightarrow j_{2k} \rightarrow j_{2k+1} .$$

Nun muß in jeder linearen Erweiterung von $\vec{Q}_n([2k+1, n] \cup [z_{2k-2}, z_{n-5}])$ (bzw. $\vec{Q}_n([2k+2, n] \cup [z_{2k-2}, z_{n-5}])$) die gesamte Kette z_{2k-2} am Anfang stehen (denn z_{k-2} ist unmittelbarer Vorgänger der beiden einzigen ersten Elemente in $\vec{Q}_n([2k+1, n] \cup [z_{2k-4}, z_{n-5}])$). Da z_{2k-1} sonst keinen weiteren Nachbarn besitzt, gilt sogar:

$$(5) \quad \vec{Q}_n([2k+1, n] \cup [z_{2k-2}, z_{n-5}]) = z_{2k-2} \rightarrow \left(\vec{Q}_n([2k+1, n] \cup [z_{2k}, z_{n-5}]) \parallel z_{2k-1} \right),$$

$$(6) \quad \vec{Q}_n([2k+2, n] \cup [z_{2k-4}, z_{n-5}]) = z_{2k-2} \rightarrow \left(\vec{Q}_n([2k+2, n] \cup [z_{2k}, z_{n-5}]) \parallel z_{2k-1} \right).$$

Für k gerade sind j_{2k} und j_{2k+1} Vorgänger und z_{2k-2} ist Nachfolger.

Nach Voraussetzung sind $Opt_\pi \left(\vec{Q}_n([2k+1, n] \cup [z_{2k}, z_{n-5}]) \right)$ und $Opt_\pi \left(\vec{Q}_n([2k+2, n] \cup [z_{2k}, z_{n-5}]) \right)$ bekannt und damit (5) und (6) serien-parallel. Man kann Optimallösungen also mit Lawlers Algorithmus in der gewünschten Laufzeit berechnen. Dasselbe gilt anschließend für (3) und (4). Der kleinere der beiden Zielfunktionswerte entspricht dem gesuchten Optimum. \square

Wir geben den Algorithmus explizit an.

Algorithmus 2 für $1|\vec{Q}_n|\sum w_j C_j$ **Eingabe:** Prozeßzeiten $p_1, \dots, p_{\tilde{n}}$ und Gewichte $w_1, \dots, w_{\tilde{n}}$ **Ausgabe:** Eine optimale lineare Erweiterung $Opt_\pi(P)$, die $\sum w_j C_j$ minimiert

```
function lawler(P)
    bestimme mit Lawlers Algorithmus optimale
    Permutation  $Opt_\pi(P)$  für  $\min \sum_{j \in P} w_j C_j$ ;
    return  $Opt_\pi(P)$ ;
```

```
function CT( $\pi$ )
    return  $\sum w_j C_j$  für lineare Erweiterung  $\pi$ ;
```

1. Initialisierung

$$\begin{aligned} X_1 &= \infty; & /* Opt \left(\vec{Q}_n^{j_{2k}} \left([2k-1, n] \cup [z_{2k-2}, z_{n-5}] \right) \right) */ \\ X_2 &= \infty; & /* Opt \left(\vec{Q}_n^{j_{2k+1}} \left([2k-1, n] \cup [z_{2k-2}, z_{n-5}] \right) \right) */ \\ Y_1 &= \infty; & /* Opt \left(\vec{Q}_n^{j_{2k}} \left([2k, n] \cup [z_{2k-2}, z_{n-5}] \right) \right) */ \\ Y_2 &= \infty; & /* Opt \left(\vec{Q}_n^{j_{2k+1}} \left([2k, n] \cup [z_{2k-2}, z_{n-5}] \right) \right) */ \end{aligned}$$

2. Ersetzung von $z_2, \dots, z_{n-5}, j_1, j_n$
für $j \in \{z_2, \dots, z_{n-1}, j_1, j_n\}$
Ersetze j durch $lawler(j)$;

3. Berechnung der Intervalle

für $(k = \lfloor \frac{n}{2} \rfloor - 1, k \geq 1, k \text{ --})$ wenn k ungerade

$$\left\{ \begin{array}{l} X_1 = \text{CT} \left(\left(\text{Opt}_\pi \left(\vec{Q}_n([2k+1, n] \cup [z_{2k}, z_{n-5}]) \right) \parallel j_{2k-1} \right) \rightarrow j_{2k} \right); \\ X_2 = \text{CT} \left(\left(\text{Opt}_\pi \left(\vec{Q}_n([2k+2, n] \cup [z_{2k}, z_{n-5}]) \right) \parallel j_{2k-1} \right) \rightarrow j_{2k} \rightarrow j_{2k+1} \right); \\ Y_1 = \text{CT} \left(\text{Opt}_\pi \left(\vec{Q}_n([2k+1, n] \cup [z_{2k}, z_{n-5}]) \right) \rightarrow j_{2k} \right); \\ Y_2 = \text{CT} \left(\text{Opt}_\pi \left(\vec{Q}_n([2k+2, n] \cup [z_{2k}, z_{n-5}]) \right) \rightarrow j_{2k} \rightarrow j_{2k+1} \right); \end{array} \right\}$$

wenn k gerade

$$\left\{ \begin{array}{l} X_1 = \text{CT} \left(j_{2k} \rightarrow \left(\left(\text{Opt}_\pi \left(\vec{Q}_n([2k+1, n] \cup [z_{2k}, z_{n-5}]) \right) \parallel j_{2k-1} \right) \right) \right); \\ X_2 = \text{CT} \left(j_{2k+1} \rightarrow j_{2k} \rightarrow \left(\left(\text{Opt}_\pi \left(\vec{Q}_n([2k+2, n] \cup [z_{2k}, z_{n-5}]) \right) \parallel j_{2k-1} \right) \right) \right); \\ Y_1 = \text{CT} \left(j_{2k} \rightarrow \text{Opt}_\pi \left(\vec{Q}_n([2k+1, n] \cup [z_{2k}, z_{n-5}]) \right) \right); \\ Y_2 = \text{CT} \left(j_{2k+1} \rightarrow j_{2k} \rightarrow \text{Opt}_\pi \left(\vec{Q}_n([2k+2, n] \cup [z_{2k}, z_{n-5}]) \right) \right); \end{array} \right\}$$

$$\text{Opt} \left(\vec{Q}_n([2k-1, n] \cup [z_{2k-2}, z_{n-5}]) \right) := \min(X_1, X_2);$$

$$\text{Opt} \left(\vec{Q}_n([2k, n] \cup [z_{2k-2}, z_{n-5}]) \right) := \min(Y_1, Y_2);$$

Theorem 5.11 *Algorithmus 2 löst das Durchlaufzeitproblem über \vec{Q}_n , $\check{n} := |\vec{Q}_n|$, in einer Laufzeit von $O(\check{n}^2 \log \check{n})$.*

Beweis. Schritt 2 benötigt mit Lawler eine Laufzeit von $O(n \log \check{n})$. Es existieren für $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$ $O(n)$ Teilprobleme $\vec{Q}_n([2k-1, n] \cup [z_{2k-1}, z_{n-5}])$ und $\vec{Q}_n([2k, n] \cup [z_{2k-1}, z_{n-5}])$ der Intervallgröße $O(\check{n})$. Jedes Teilproblem benötigt nach Lemma 5.10 also $O(\check{n} \log \check{n})$ und die Gesamtlaufzeit ist

$$O(n \cdot \check{n} \log \check{n}) = O(\check{n}^2 \log \check{n}) \quad . \quad \square$$

Die Lösung des Durchlaufzeitproblems für die Spinnennetzkomposition $P \star \vec{Q}_n$ verläuft analog zu Algorithmus 2.

Sei o.B.d.A. n teilbar durch vier. Dann gilt $j_n \rightarrow P \rightarrow j_{n-2}$. $Opt_\pi(P)$ wurde schon vorab im Dekompositionsbaum berechnet. Dann unterscheiden sich die Berechnungen bei der Spinnennetzkomposition zu der des angespitzten Spinnennetzes nur bei den Teilproblemen $Opt_\pi(\vec{Q}_n[n-1, n])$ und $Opt_\pi(\vec{Q}_n[n-2, n])$. Danach werden diese Reihenfolgen beibehalten und dementsprechend wird P nicht mehr explizit betrachtet. Mit $\check{n} := |P \star \vec{Q}_n|$ ergibt sich:

Korollar 5.12 $1 \mid P \star \vec{Q}_n \mid \sum w_j C_j$ ist lösbar in $O(\check{n}^2 \log \check{n})$, sofern eine optimale lineare Erweiterung für P bekannt ist.

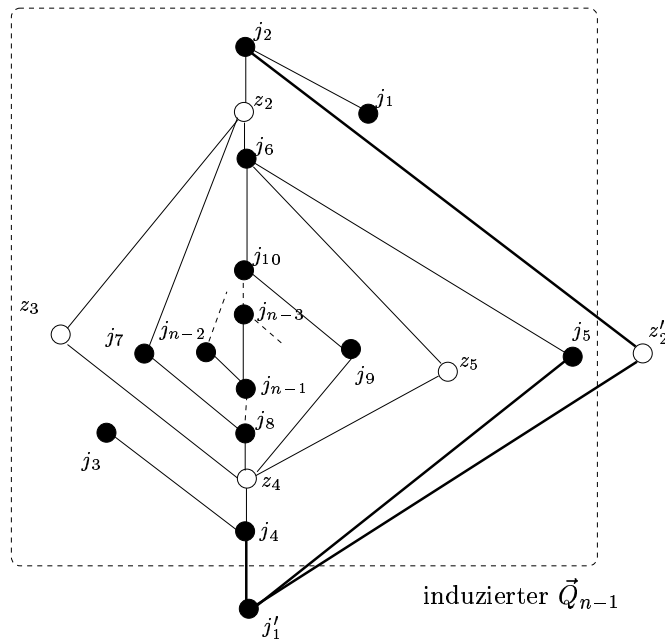
Im Fall von Komplementen von angespitzten Spinnennetzen $\overline{\vec{Q}}_n$ mit Knotenmenge $\{j'_1, j_1, \dots, j_{n-1}\} \cup \{z'_2, z_2, \dots, z_{n-6}\}$ löst man zunächst das nach Satz 4.6 ohne die Knoten j'_1, z'_2 isomorphe Problem auf \vec{Q}_{n-1} mit Knotenmenge $\{j_1, \dots, j_{n-1}\} \cup \{z_2, \dots, z_{n-6}\}$. Es sei o.B.d.A. $j_1 \rightarrow j_2$ im induzierten \vec{Q}_{n-1} (siehe Abb. 5.8).

Dann sind im besonderen Optimallösungen für die Teilprobleme auf den Intervallen $([3, n] \cup [z_2, z_{n-6}])$ und $([4, n] \cup [z_2, z_{n-6}])$ bekannt. Nun können in $\overline{\vec{Q}}_n$ nur j_1 oder j'_1 an erster Stelle stehen und j_2 oder j_3 an letzter. D.h.

$$Opt(\overline{\vec{Q}}_n) = Min\{(7), (8), (9), (10)\} \text{ mit}$$

$$\begin{array}{l} (7) \quad Opt \left(\begin{array}{l} j'_1 \rightarrow \left(\overline{\vec{Q}}_n([3, n] \cup [z_2, z_{n-6}]) \mid j_1 \quad \mid z'_2 \right) \rightarrow j_2 \end{array} \right) , \\ (8) \quad Opt \left(\begin{array}{l} j'_1 \rightarrow \left(\overline{\vec{Q}}_n([4, n] \cup [z_2, z_{n-6}]) \mid j_1 \quad \mid z'_2 \right) \rightarrow j_2 \rightarrow j_3 \end{array} \right) , \\ (9) \quad Opt \left(\begin{array}{l} j_1 \rightarrow j'_1 \rightarrow \left(\overline{\vec{Q}}_n([3, n] \cup [z_2, z_{n-6}]) \quad \mid z'_2 \right) \rightarrow j_2 \end{array} \right) , \\ (10) \quad Opt \left(\begin{array}{l} j_1 \rightarrow j'_1 \rightarrow \left(\overline{\vec{Q}}_n([4, n] \cup [z_2, z_{n-6}]) \quad \mid z'_2 \right) \rightarrow j_2 \rightarrow j_3 \end{array} \right) . \end{array}$$

series-parallel

Abbildung 5.8: \overline{Q}_n

Die vier Teilprobleme sind serien-parallel und man erhält auch im Falle, daß die z_i durch serien-parallele Ordnungen ersetzt sind:

Korollar 5.13 Eine Variante von Algorithmus 2 löst $1 | \overline{Q}_n | \sum w_j C_j$ mit $\check{n} := |\overline{Q}_n|$ in $O(\check{n}^2 \log \check{n})$ Zeit.

Die Übertragung auf die Spinnennetzkomposition $P \star \overline{Q}_n$ ist offensichtlich.

Korollar 5.14 Eine Variante von Algorithmus 2 löst $1 | P \star \overline{Q}_n | \sum w_j C_j$ mit $\check{n} := |P \star \overline{Q}_n|$ in $O(\check{n}^2 \log \check{n})$ Zeit, sofern eine optimale lineare Erweiterung für P bekannt ist.

Die wichtigsten Ergebnisse dieses Abschnitts sind in Tabelle 5.1 zusammengefaßt.

$(q, q - 3)$ -Klasse, $q \geq 7$, P Kette	Laufzeit in Abhängigkeit der Knotenanzahl	
$P\Delta H$	$O(\check{n}^{q-1})$,	Satz 5.1
\vec{P}_n	$O(\check{n}^4 \log \check{n})$,	Korollar 5.4
C_n, n gerade	$O(n^4 \log n)$,	Korollar 5.5
\overline{P}_n	$O(\check{n}^3)$,	Korollar 5.6
$P \bowtie_v \{A, B, C, D\}$	$O(\check{n} \log \check{n})$,	Korollar 5.7
$P \star \vec{Q}_n$ bzw. \vec{Q}_n	$O(\check{n}^2 \log \check{n})$,	Korollar 5.12 bzw. Theorem 5.11
$P \star \overline{Q}_n$ bzw. \overline{Q}_n	$O(\check{n}^2 \log \check{n})$,	Korollar 5.14 bzw. Korollar 5.13

Tabelle 5.1: Laufzeiten

Da die Laufzeit dominiert wird durch Kompositionen $P\Delta H$, erhalten wir zusammenfassend den Hauptsatz dieser Arbeit.

Theorem 5.15 $1 \mid (q, q - 3) \mid \sum w_j C_j, q \geq 7$, ist lösbar in einer Laufzeit von $O(n^{q-1})$, wobei n die Knotenanzahl ist.

Als Folgerung – wie schon von Sidney und Steiner bekannt – gilt diesselbe Laufzeit für das Durchlaufzeitproblem über $(q, q - 4)$ -Partialordnungen.

Kapitel 6

Zusammenfassung und Ausblick

In dieser Arbeit wird das Ein-Maschinen-Durchlaufzeitproblem mit Reihenfolgerrestriktionen, die durch eine $(q, q - 3)$ -Partialordnungen mit festem q , $q \geq 7$, gegeben sind, effizient gelöst.

$(q, q - 3)$ -Graphen wurden von Babel eingeführt. Sie sind eine echte Obermenge von Cographen, P_4 -dünnen und P_4 -erweiterbaren Graphen. Ihre p -zusammenhängenden Komponenten sind im wesentlichen Spinnen, Pfade, Spinnennetze, Kreise und Graphen beschränkter Größe.

Wir haben $(q, q - 3)$ -Graphen auf Komparabilitätsgraphen eingeschränkt und basierend auf dem Strukturtheorem eine Vorschrift zum konstruktiven Aufbau von $(q, q - 3)$ -Partialordnungen hergeleitet. Als Nebenresultat ergibt sich eine Konstruktionsvorschrift für $(q, q - 4)$ -Partialordnungen, die eine echte Subklasse der $(q, q - 3)$ -Partialordnungen und eine echte Subklasse der Partialordnungen mit beschränkter Dekompositionsweite sind.

Anschließend haben wir nacheinander für die einzelnen Komponenten und Kompositionen der Aufbauvorschrift polynomielle Algorithmen entwickelt. Für einige davon haben wir schon bekannte polynomielle Verfahren herangezogen. Für die restlichen offenen Klassen bilden zwei von uns entwickelte dynamische Programmierungsverfahren die Basis der effizienten Lösbarkeit des Durchlaufzeitproblems auf sämtlichen $(q, q - 3)$ -Partialordnungen.

Der erste dynamische Programmierungsalgorithmus löst das Problem über Pfaden in einer Laufzeit von $O(n^4 \log n)$, der zweite Spinnennetze in $O(n^2 \log n)$. Die Kernidee beider besteht darin, daß sich größere Teilprobleme auf schon vorab berechnete kleinere Teilprobleme zurückführen lassen, indem die Position bestimmter Elemente fixiert wird. Essentiell ist, daß diese

Elemente in jeder beliebigen linearen Erweiterung nur an wenigen möglichen Positionen stehen können.

Leichte Abänderungen des Algorithmus für Pfade lösen das Durchlaufzeitproblem mit sonstigen pfadähnlichen $(q, q - 3)$ -Reihenfolgerestriktionen. Dazu zählen Kreise und verallgemeinerte angespitzte Pfade, die eine echte Obermenge von angespitzten Pfaden sind.

Die Laufzeit für die gesamte Klasse der $1|(q, q - 3)| \sum w_j C_j$ Probleme mit fest vorgegebenem $q \geq 7$ wird dominiert durch Partialordnungen, deren äußerer Faktor höchstens q Knoten und maximal $q - 3$ induzierte N s besitzt. Die Berechnung benötigt dann eine Laufzeit von $O(n^{q-1})$. Die Aussage, daß das Durchlaufzeitproblem über sämtlichen $(q, q - 3)$ -Partialordnungen in dieser Laufzeit optimal zu lösen ist, bildet den Hauptsatz der vorliegenden Arbeit.

Vorab war bekannt, daß das Ein-Maschinen-Durchlaufzeitproblem mit serienparallelen, N -dünnen und N -erweiterbaren Reihenfolgerestriktionen, allgemein solchen, die beschränkte Dekompositionsweite haben, effizient lösbar ist. Die ersten drei genannten sind spezielle $(q, q - 4)$ -, im Besonderen also $(q, q - 3)$ -Partialordnungen und die vorgestellten Ergebnisse bilden eine Verallgemeinerung.

Abbildung 6.1 gliedert die Ergebnisse der Arbeit in die vorab bestehenden ein. Sie sind gestrichelt umrahmt.

Pfade und angespitzte Spinnennetze sind – zu unserem besten Wissen – die ersten nicht-trivialen Module mit unbeschränkter Weite bzw. Dekompositionsweite, für die ein polynomieller Algorithmus für das Durchlaufzeitproblem existiert. Damit wird die seit langem offene Frage nach der Existenz effizient zu lösender Poset-Klassen unbeschränkter Weite für das Ein-Maschinen-Durchlaufzeitproblem positiv beantwortet.

Die Klasse der verallgemeinerten angespitzten Pfade als echte Oberklasse angespitzter Pfade, gibt Anlaß zu weiteren effizient zu lösenden nicht-trivialen unbeschränkten Modulen.

Zudem liefert die Anwendung des Strukturtheorems auf Komparabilitätsgraphen $G(P)$ die Einsicht, daß Prim-Kompositionen des assoziierten Posets P stets separabel p -zusammenhängende Komparabilitätsgraphen zugrundeliegen. Diese neue Erkenntnis unterstreicht die Wichtigkeit, allgemein separabel p -zusammenhängende Graphen auf ihre Komparabilitätsgrapheneigenschaft zu untersuchen.

Abschließend ergänzen wir den Kenntnisstand für $(q, q - 3)$ -Partialordnungen mit $q \in \{4, 5, 6\}$.

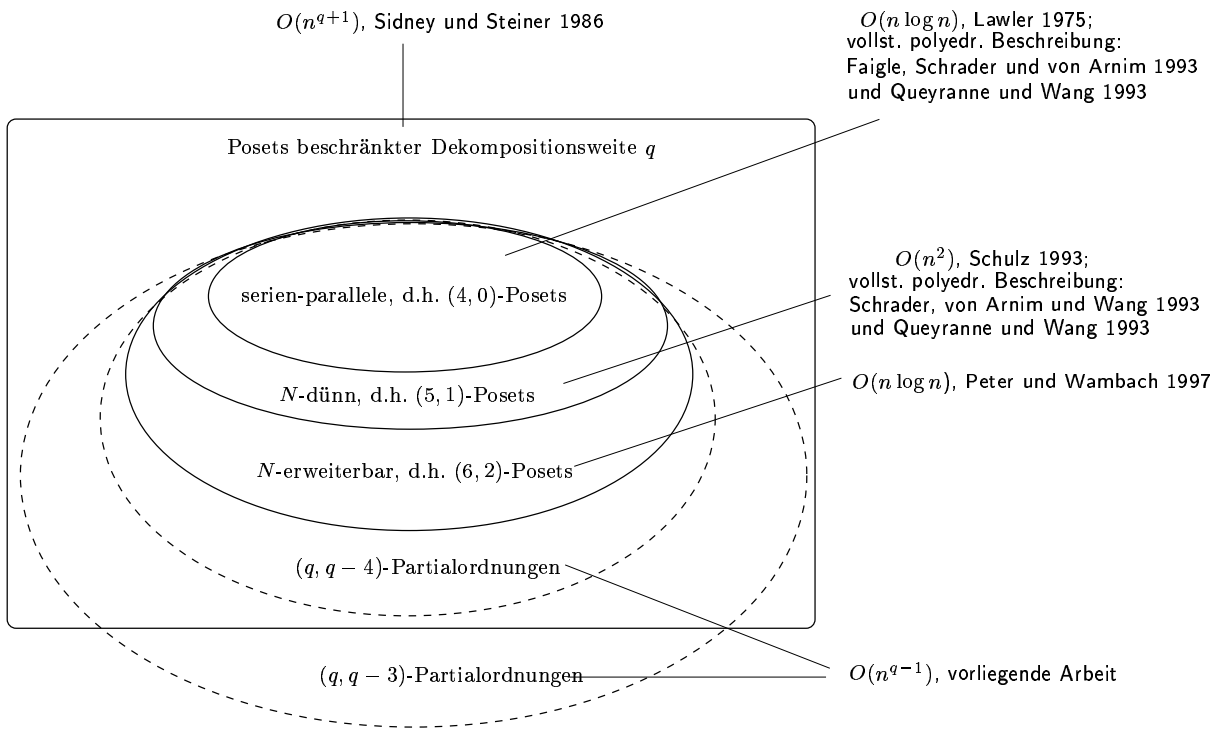


Abbildung 6.1: Polynomielle Subklassen für $1 \mid prec \mid \sum w_j C_j$

Das Durchlaufzeitproblem ist trivialerweise NP-schwer für $(4, 1)$ -Partialordnungen, denn dies sind alle Partialordnungen. $(5, 2)$ -Posets umfassen auch sämtliche Partialordnungen, denn nur der induzierte Kreis der Länge fünf, der kein Komparabilitätsgraph ist, besitzt auf fünf Knoten mehr als zwei Pfade der Länge drei.

Offen bleibt die Frage, ob das Durchlaufzeitproblem auch polynomiell lösbar ist für $(6, 3)$ -Partialordnungen.

Literaturverzeichnis

- [AH73] D. Adolphson und T.C. Hu. *Optimal linear ordering*. SIAM Journal of Applied Mathematics 25, 403–423, 1973.
- [Bab97] L. Babel. *On the P_4 -Structure of graphs*. Habilitationsschrift, Technische Universität München, 1997.
- [Bal75] E. Balas. *A linear characterization of permutation vectors*. Management science research report 264, Carnegie-Mellon University, Pittsburgh, PA, 1975.
- [Bau96] S. Baumann. *A linear algorithm for the homogeneous decomposition of graphs*. Technical Report M-9615, Technische Universität München, 1996.
- [BC84] C. Berge (Ed.) und V. Chvatal (Ed.). *Topics on perfect graphs*. Annals of Discrete Mathematics, 21. North-Holland Mathematics Studies, 88. Amsterdam - New York - Oxford: North-Holland. XIV, 369 p., 1984.
- [BKK⁺01] L. Babel, T. Kloks, J Kratochvil, D. Kratsch, H. Müller und S. Olariu. *Efficient algorithms for graphs with few P_4 s*. Discrete Mathematics 235, 29–51, 2001.
- [BM76] J.A Bondy und U.S.R. Murty. *Graph Theory with Applications*. North-Holland, Amsterdam, 1976.
- [BM83] H. Buer und R.H. Möhring. *A fast algorithm for the decomposition of graphs and posets*. Mathematics of Operation Research 8, No.2, 170–184, 1983.
- [BO98] L. Babel und S. Olariu. *On the structure of graphs with few P_4 s*. Discrete Applied Mathematics, 1–13, 1998.
- [Bru95] P. Brucker. *Scheduling Algorithms*. Springer, Berlin, 1995.

- [CH94] A. Cournier und M. Habib. *A new linear time algorithm for modular decomposition*. In: *Trees in Algebra and Programming*, Lecture Notes in Computer Science 787, 68–84. Springer-Verlag, 1994.
- [CWM67] R.W. Maxwell, W.L. Conway und L.W. Miller. *Theory of Scheduling*. Addison-Wesley, Reading, MA, 1967.
- [DGM96] E. Dahlhaus, J. Gustedt und R.M. McConnell. *Efficient and practical modular decomposition*. Technical Report 524/1996, Technische Universität Berlin, 1996.
- [Die96] R. Diestel. *Graphentheorie*. Springer Verlag, Berlin, 1996.
- [Dil50] R.P. Dilworth. *A decomposition theorem for partially ordered sets*. *Annals of Mathematics* 2, 161–166, 1950.
- [FG84] U. Faigle und G. Gierz. *A construction for strongly greedy ordered sets*. In G. Hammer und D. Pallschke, editors, *Selected topics in operations Research and mathematical Economics*, 307–324. Springer Lecture Notes in Economics and Mathematical Systems 226, 1984.
- [GLLRK79] R.L. Graham, E.L. Lawler, J.K. Lenstra und A.H.G. Rinnooy Kan. *Optimization and approximation in deterministic sequencing and scheduling: a survey*. *Annals of Discrete Mathematics* 5, 287–326, 1979.
- [Gol80] M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, Inc., 1980.
- [GW98] M.X. Goemans und D.P. Williamson. *Two-dimensional gantt-charts and a scheduling algorithm of Lawler*. *SIAM Journal of Discrete Mathematics* 13, 281–294, 1998.
- [HJ85] R.H. Habib und R. Jegou. *N-free posets as generalizations of series-parallel posets*. *Discrete Applied Mathematics* 12, 279–291, 1985.
- [HM87] M. Habib und R.H. Möhring. *On some complexity properties of N-free posets and posets with bounded decomposition diameter*. *Discrete Mathematics* 63, 157–182, 1987.

- [Hor72] W.A. Horn. *Single machine job sequencing with treelike precedence ordering and linear delay penalties*. SIAM Journal of Applied Mathematics 23, 189–202, 1972.
- [HS95] W. Hochstättler und H. Schindler. *Recognizing P_4 -extendible graphs in linear time*. Technical Report 95-188, Universität zu Köln, ZPR, 1995.
- [JO89a] B. Jamison und S. Olariu. *A new class of brittle graphs*. Studies in Applied Mathematics, 89–92, 1989.
- [JO89b] B. Jamison und S. Olariu. *P_4 -reducible graphs - a class of uniquely tree-representable graphs*. Studies Applied Mathematics 81, 79–87, 1989.
- [JO92] B. Jamison und S. Olariu. *On a unique tree representation for P_4 sparse graphs*. Discrete Applied Mathematics 35, 115–129, 1992.
- [JO95a] B. Jamison und S. Olariu. *Linear time optimization for P_4 -sparse graphs*. Discrete Applied Mathematics 61, 155–175, 1995.
- [JO95b] B. Jamison und S. Olariu. *p -components and the homogeneous decomposition of graphs*. SIAM Journal of Discrete Mathematics 8, 448–463, 1995.
- [KSW98] D. Karger, C. Stein und J. Wein. *Scheduling algorithms*. In M.J. Atallah, editor, *Algorithms and Theory of Computation Handbook*. CRC Press, Washington D.C., USA, 1998.
- [Law76] E.L. Lawler. *Combinatorial Optimization, Networks and Matroids*. Holt, Rinehart & Winston, New York, 1976.
- [Law78] E.L. Lawler. *Sequencing jobs to minimize total weighted completion time subject to precedence constraints*. Annals of Discrete Mathematics 2, 75–90, 1978.
- [Möh85] R.H. Möhring. *Algorithmic aspects of comparability graphs and interval graphs*. In I. Rival, editor, *Graphs and Order*, 41–101. Reidel, Dordrecht, 1985.
- [Möh89] R.H. Möhring. *Computationally tractable classes of ordered sets*. In I. Rival, editor, *Algorithms and Order*, 105–193. Kluwer Academic Publishers, Dordrecht, 1989.

- [MR84] R.H. Möhring und F.J. Radermacher. *Substitution decomposition for discrete structures and connections with combinatorial optimization*. Annals of Discrete Mathematics 19, 257–356, 1984.
- [MR85] R.H. Möhring und F.J. Radermacher. *Generalized results on the polynomiality of certain weighted sum scheduling problems*. Methods of Operations Research 49, 405–417, 1985.
- [MS94] R.M. McConnell und J.P. Spinrad. *Linear-time modular decomposition and efficient transitive orientation of comparability graphs*. In: Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms, Virginia, 536–544, 1994.
- [MS99] R.M. McConnell und J.P. Spinrad. *Modular decomposition and transitive orientation*. Discrete Mathematics (201), 189–241, 1999.
- [Par95] R.G. Parker. *Deterministic Scheduling Theory*. Chapman & Hall, London, 1995.
- [Pin95] M. Pinedo. *Scheduling: theory, algorithms and systems*. Prentice Hall, New Jersey, 1995.
- [PW97] M. Peter und G. Wambach. *N-extendible posets and how to minimize total weighted completion time*. Discrete Applied Mathematics 99, 157–167, 1997.
- [QS95] M. Queyranne und A.S. Schulz. *Polyhedral approaches to machine scheduling*. Technical Report 474/1995, Technische Universität Berlin, 1995.
- [QW91] M. Queyranne und Y. Wang. *Single-machine scheduling polyhedra with precedence constraints*. Mathematics of Operations Research 16, 1–29, 1991.
- [Rad52] R. Rado. *An inequality*. Journal of London Math. Society 2, 1–6, 1952.
- [Riv82] I. Rival, editor. *Ordered sets*. Reidel, Dordrecht, 1982.
- [Riv85] I. Rival, editor. *Graphs and order: the role of graphs in the theory of ordered sets and its applications*. Reidel, Dordrecht, 1985.

- [SB78] L. Schrage und K.R. Baker. *Dynamic programming solution for sequencing problems with precedence constraints*. Operations Research 26, 444–449, 1978.
- [Sch11] P.H. Schoute. *Analytical treatment of the polytopes regularly derived from regular polytopes*. Johannes Müller, 1911.
- [Sch93] A.S. Schulz. *Single machine scheduling subject to n -sparse precedence constraints*. In *3rd twente Workshop on Graphs and Combinatorial Optimization, Memorandum No. 1132*, 166–169. Faculty of Applied Mathematics, University of Twente, The Netherlands, 1993.
- [Sid75] J.B. Sidney. *Decomposition algorithms for single-machine sequencing with precedence relations and deferral costs*. Operations Research Vol. 23, No.2, 283–298, 1975.
- [Smi56] W.E. Smith. *Various optimizers for single-stage production*. Naval Research Logis. Quarterly. 3, 59–66, 1956.
- [SS86] J.B. Sidney und G. Steiner. *Optimal sequencing by modular decomposition: Polynomial algorithms*. Operations Research Vol. 34, No. 4, 606–612, 1986.
- [vA93] A. von Arnim. *Permutaeder von Ordnungen*. PhD thesis, Universität zu Köln, 1993.
- [vAFS90] A. von Arnim, U. Faigle und R. Schrader. *The permutahedron of series-parallel posets*. Discrete Applied Mathematics 28, 3–9, 1990.
- [vAS97] A. von Arnim und A.S. Schulz. *Facets of the generalized permutahedron of a poset*. Discrete Applied Mathematics 72, 179–192, 1997.
- [vASW96] A. von Arnim, R. Schrader und Y. Wang. *The permutahedron of P_4 -sparse posets*. Mathematical Programming 75, 1–18, 1996.
- [You78] H.P. Young. *On permutations and permutation polytopes*. Mathematical Programming Study 8, 128–140, 1978.

Index

- (4, 0)-Graph, 1
- (5, 1)-Graph, 1
- (6, 2)-Graph, 1
- $(q, q - 4)$ -Graph, 25
- (q, t) -Graph, 1, 25
- C^L , 16
- C_n , 4, 47
- $G(P)$, 7
- $I(w)$, 9
- I^* , 17
- $Max(P)$, 9
- $Min(P)$, 9
- N -dünn, 14
- N -erweiterbar, 13
- N -frei, 12
- $N(v)$, 3
- $N(v_i)^+$, 31
- OC_5 , 13
- $Opt(P)$, 53
- $Opt_\pi(P)$, 53
- $P([s, t])$, 52
- $P^z([s, t])$, 52
- P^{-1} , 9
- P_4 -Übergang, 4
- P_4 -dünn, 1
- P_4 -erweiterbar, 1, 12, 27
- P_4 -leicht, 28
- P_4 -reduzierbar, 14, 28
- Q_n , 30, 31
- R_5, R_6, R_7 , 30
- $T(P)$, 10
- $[s, t]$, 52
- \bowtie_e , 13
- \bowtie_g , 14, 39
- \bowtie_v , 39
- Λ -Relation, 7
- Λ^* -Klasse, 7
- \mathcal{A}_w , 22
- \mathcal{B}_w , 22
- $\mathcal{G}(q, t)$, 27
- \prec , 9
- \mathbb{P}_n , 53
- \oplus , 12
- \overline{G} , 4
- \overline{P}_n , 47
- \overline{Q}_n , 47
- ρ -Wert, 17
- ρ -minimal, 17
- \rightarrow , 10
- \star , 44
- \triangle , 46
- \vec{P}_n , 31, 47
- \vec{Q}_n , 31, 47
- \parallel , 10
- $j\Delta H$, 46, 52
- Aktivität, 1
- angespitzt, 31
- Antikette, 6
- AU-äußerer Faktor, 10
- CAC, 12
- Clique, 3
- Cograph, 1, 4, 11
- Cozusammen
 - hangskomponente, 4
- cozusammen

- hängend, 4
- DAG, 7
- Dekomposition
 - sdurchmesser, 10
 - weite, 10
 - homogene, 11
 - lexikographische, 9
 - modulare, 11
 - Substitutions-, 9
- direkter
 - Nachfolger, 9
 - Vorgänger, 9
- Durchlaufzeit, 1, 15
- Dynamische Programmierung, 21
- Endknoten, 4
- erweiterte Spinnenkomposition, 13
- Fertigstellungszeit, 15
- Fertigstellungszeitenvektor, 16
- Filter, 9
 - echter, 9
 - Haupt-, 9
- Fläche, 16
- Gantt-Chart, 15
 - 2D-, 16
- Graph
 - charakteristischer, 4
- Hasse-Diagramm, 9
- Hauptideal, 9
- homogen, 3
- Ideal, 9
 - echtes, 9
 - Haupt-, 9
- induzierter Teilgraph, 3
- innerer Faktor, 10
- Intervall, 52
- Intervallgröße, 52
- irreduzibel, 10
- isomorph, 4
- Job, 15
 - zusammengesetzter, 19
- Kante, 3
- Kette, 6
- Ketten-Antiketten-vollständig, 12
- Komparabilitätsgraph, 7
- kompatibel, 9
- Komplement, 4
- Komposition
 - Δ , 46
 - erweiterte Spinnen-, 13
 - gewöhnliche Spinnen-, 14
 - parallel, 10
 - Prim-, 10
 - quasi-serielle, 12
 - seriell, 10
 - Spinnennetz-, 44
- Kompositionsbaum, 10
- konvexe Menge, 9
- Kreis, 4
- Länge
 - eines Pfades, 4
- lineare Erweiterung, 9
- Mittelknoten, 4
- Modul, 10
- modulare Dekomposition, 9
- n, 3
- Nachbar, 3
- p-Baum, 29, 32
- p-Endknoten, 4
- p-Kette, 30
 - angespitzte, 31
- p-Kreis, 29
 - klein, 29
- p-zusammenhängend, 1, 4
 - minimal, 27

- parallel, 6
- Parallelkomposition, 10
- Partialordnung
 - N -dünn, 14
 - N -frei, 12
 - N -reduzierbar, 14
 - assoziiert mit, 7
 - Definition, 6
 - Intree-, 12
 - Outtree-, 12
 - schwache, 12
 - serien-parallel, 11
 - Threshold-, 12
- partiell geordnete Menge, 6
- Permutaeder, 23
- Permutationsvektor, 23
- Pfad, 4
 - angespitzt, 31
 - Länge, 4
 - verallgemeinerter angespitzter, 53
- Poset, 6
- prim, 10
- Prim-Komposition, 10

- Reihenfolgeplan, 15

- separabel, 5
- Separation, 5
- separierbar, 5
- serien-parallel, 11
- Serienkomposition, 10
- Smith's rule, 16
- Spinne, 26
 - echt, 26
- Spinnenkomposition, 14
 - erweiterte, 13
 - gewöhnliche, 39
 - verallgemeinerte, 39
- Spinnennetz, 31
 - komposition, 44
 - angespitztes, 31
 - stabil, 3
 - Strukturtheorem, 5

 - Totalordnung, 6
 - transitive Orientierung, 7
 - forcieren, 7
 - transitiver Abschluß, 7

 - UB-Überdeckungsgraph, 9
 - UB-überdeckt, 9
 - ungerade Krone, 13
 - Untergraph, 3
 - unzerlegbar, 10

 - vergleichbar, 6

 - Weg, 4
 - Weite, 6
 - Wurzelbaum, 6

 - Zaun, 39
 - zerlegbar, 10
 - Zick-Zack, 39
 - zulässig, 9

Abstract

Suppose n jobs are to be sequenced for processing by a single machine, with the objective to minimize weighted completion time. It is shown, that the problem is solvable by dynamic programming in $O(n^{q-1})$ time, if precedence constraints are given by a $(q, q-3)$ -poset with q fixed and greater or equal seven.

$(q, q-3)$ -graphs have been introduced by Babel as one of several generalizations of cographs. Generally a graph is called a (q, t) -graph if no set of at most q vertices induces more than t distinct P_4 s. In this sense, cographs are precisely $(4, 0)$ -graphs, P_4 -sparse graphs coincide with $(5, 1)$ -graphs and P_4 -extendible graphs result in $(6, 2)$ -graphs without cycles of length five. $(q, q-3)$ -graphs are a proper superset of $(q, q-4)$ -graphs, as an induced path of length k , $k \geq 5$, proves.

A $(q, q-3)$ -poset is a partial order where the underlying comparability graph is a $(q, q-3)$ -graph.

Our result generalizes previous results for the special cases of $(q, q-4)$ -posets. To the best of our knowledge $(q, q-3)$ -posets are the first tractable class for the weighted completion time problem with unbounded decomposition width.

Deutsche Kurzzusammenfassung

In dieser Arbeit wird gezeigt, daß das Ein-Maschinen-Durchlaufzeitproblem mit Reihenfolgerestriktionen, die durch eine $(q, q - 3)$ -Partialordnungen mit festem q , $q \geq 7$, gegeben sind, durch dynamische Programmierung in einer Laufzeit von $O(n^{q-1})$ lösbar ist.

Allgemein heißt ein Graph (q, t) -Graph, wenn q Knoten höchstens t verschiedene Pfade der Länge drei induzieren. In diesem Sinne sind Cographen $(4, 0)$ -Graphen, P_4 -dünne Graphen sind per Definition $(5, 1)$ -Graphen und P_4 -erweiterbare Graphen ohne induzierte Kreise der Länge fünf entsprechen $(6, 2)$ -Graphen.

Eine $(q, q - 3)$ -Partialordnung ist eine Partialordnung, deren zugrundeliegender Komparabilitätsgraphen ein $(q, q - 3)$ -Partialordnungen ist.

Vorab bekannt war, daß das Ein-Maschinen-Durchlaufzeitproblems mit serienparallelen, N -dünnen und N -erweiterbaren Reihenfolgerestriktionen, sowie solchen, die beschränkte Dekompositionsweite haben, effizient lösbar ist. Die ersten drei genannten sind spezielle $(q, q - 4)$ -, im Besonderen also $(q, q - 3)$ -Partialordnungen und die vorgestellten Ergebnisse bilden eine Verallgemeinerung.

$(q, q - 3)$ -Graphen sind – zu unserem besten Wissen – die ersten nicht-trivialen Module mit unbeschränkter Weite bzw. Dekompositionsweite, für die ein polynomieller Algorithmus für das Durchlaufzeitproblem existiert. Damit wird die seit langem offene Frage nach der Existenz effizient zu lösender Poset-Klassen unbeschränkter Weite für das Ein-Maschinen-Durchlaufzeitproblem positiv beantwortet.

Erklärung

Ich versichere, daß ich die von mir vorgelegte Dissertation vollständig und ohne unzulässige Hilfe angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; daß diese Dissertation noch keiner anderen Fakultät zur Prüfung vorgelegen hat; daß sie abgesehen von unten angegebenen Teilpublikationen noch nicht veröffentlicht worden ist sowie, daß ich eine solche Veröffentlichung vor Abschluß des Promotionsverfahrens nicht vornehmen werde. Die Bestimmungen der Promotionsordnung sind mir bekannt. Die von mir vorgelegte Promotion ist von Professor Dr. Rainer Schrader betreut worden.

V. Kaibel, R. Schrader, V. Stausberg, *The weighted completion time problem on posets with few N s*. Zur Einreichung bei Discrete Applied Mathematics fertiggestellt.

Lebenslauf

Name	Vanessa A. Stausberg
Geburtsdatum	17. Juli 1969
Geburtsort	Leverkusen-Wiesdorf
Nationalität	deutsch
Familienstand	ledig, eine Tochter

Schulausbildung

1975 - 1979	Waldschule, Leverkusen
1979 - 1988	Freiherr-vom-Stein Gymnasium, Leverkusen
06/1988	Abitur

Hochschulausbildung

1988 - 1989	Studium der Mathematik, Psychologie und Übersetzungen am University College London, London, England
1989 - 1991	Studium der Mathematik an der Heinrich Heine Universität in Düsseldorf
1991 - 1996	Fortsetzung des Mathematikstudiums an der Universität zu Köln
1992	Vordiplom in Mathematik (Köln)
1996	Diplom in Mathematik (Köln)
1996 - 1998	Stipendiatin der Universität zu Köln
1998 - 2000	Wissenschaftliche Mitarbeiterin am Institut für Informatik der Universität zu Köln (Lehrstuhl Prof. Dr. Rainer Schrader)
seit 03/2000	Erziehungsurlaub

Druck: Copy & Star, Köln