Vorwort

Es handelt sich hierbei um einen vorläufigen Aufschrieb zu Flüssen in planaren Graphen. Die Darstellung basiert auf dem Artikel "Maximum Flows and Parametric Shortest Paths in Planar Graphs" von Jeff Erickson, erschienen in SODA'10.

Illustrationen und die noch fehlenden Beweise werden im Laufe der Zeit ergänzt. Anregungen und Feedback bitte per Mail an `rutter@kit.edu`.

1 Flüsse in Planaren Graphen

Sei $G = (V, E)$ ein gerichteter planarer Graph. Jede Kante $e \in E$ verbindet zwei Knoten, den Startknoten $\text{tail}(e)$ und den Zielknoten $\text{head}(e)$. Außerdem trennt e zwei Facetten $\text{left}(e)$ und $\text{right}(e)$ links und rechts von e. Wir schreiben $u \to v$ für die Kante mit Startknoten u and Zielknoten v. Außerdem definieren wir $\text{rev}(u \to v) := v \to u$. Wir nehmen an, dass e und $\text{rev}(e)$ durch dieselbe Kurve dargestellt werden. Für einen Subgraphen H von G bezeichnet $\text{rev}(H)$ den Subgraphen, den man durch Umdrehen aller Kanten in H erhält.

Sie $e = u \to v$ eine Kante und seien f und g die Facette links bzw. rechts von e. Der gerichtete Dualgraph ist definiert durch $(u \to v)^\star := f \to g$.

Ein s–t-Schnitt ist eine Teilmenge C von Kanten in G, sodass jeder Pfad von s nach t mindestens eine Kante aus C enthält. Ein s–t-Fluss ist eine Abbildung $\Phi \colon E \to \mathbb{R}$, die zwei Bedingungen erfüllt. Zum einen muss Φ antysymmetrisch sein, das heißt, $\Phi(e) = -\Phi(\text{rev}(e))$ für alle Kanten $e \in E$. Zweitens muss Φ die Flusserhaltungsbedingung erfüllen: $\sum_w \Phi(v \to w) = 0$ für alle Knoten v außer s und t. (Zur Vereinfachung der Notation setzen wir $\Phi(v \to w) = 0$ wenn $v \to w$ keine Kante in G ist.

Sei nun $c \colon E \to \mathbb{R}$ eine Abbildung, die jeder Kante eine nicht-negative Kapazität zuordnet. Ein Fluss Φ ist gültig, wenn $\Phi(e) \leq c(e)$ für alle Kanten $e \in E$ gilt. Ein Fluss Φ saturiert eine Kante e wenn $\Phi(e) = c(e)$. Die Kapazität eine Schnittes C ist $c(C) := \sum_{e \in C} c(e)$. Das Mincut-Maxflow-Theorem besagt, dass der maximale Wert eines gültigen s–t Flusses gerade die Kapazität eines minimalen s–t-Schnittes ist. Insbesondere gilt, dass ein Fluss Φ, der alle Kanten eines Schnittes C saturiert, maximal ist, und C folglich ein minimaler Schnitt ist.

1.1 Flüssen in Planaren Graphen als Kürzeste-Wege-Problem

In diesem Abschnitt entwickeln wir ein Kriterium dafür, ob ein gegebener Graph G einen gültigen s–t-Fluß mit Wert λ besitzt oder nicht. Dieses Kriterium entspricht im wesentlichen einem Kürzeste-Wege-Problem im Dualgraphen $G^\star$ von G.

Sei P ein beliebiger einfacher s–t-Pfad in G und bezeichnet $\pi \colon E \to \mathbb{R}$ den Einheitsfluss von s nach t entlang P:

$$\pi(e) := \begin{cases} 1 & \text{falls } e \in P \\ -1 & \text{falls } \text{rev}(e) \in P \\ 0 & \text{sonst} \end{cases}$$

Für eine Kantenmenge $E' \subseteq E$, sei $\pi(E') = \sum_{e \in E'} \pi(E)$. Ein Subgraph C von G heißt *Kozykel* wenn der entsprechende duale Subgraph $C^\star$ ein einfacher gerichteter Kreis in $G^\star$ ist. Für einen Kreis $C^\star$ in $G^\star$, nennen wir $\pi(C)$ die *Kreuzungszahl* von $C^\star$. Die Kreuzungszahl von $C^\star$ ist genau die Zahl der Kreuzungen in denen P den Kreis $C^\star$ von links nach rechts kreuzt, minus die Zahl der Kreuzungen in denen P den Kreis $C^\star$ von rechts nach links kreuzt.

Lemma 1. *$\pi(C) \in \{-1, 0, 1\}$ für jeden Kozykel C. Es gilt $\pi(C) = 1$ genau dann wenn C ein s–t-Schnitt ist.*

Beweis. Jede Kante in einem Kozykel C kreuzt seinen dualen Kreis $C^\star$ von links nach rechts; jede Kante in $\text{rev}(C)$ kreuzt $C^\star$ von rechts nach links. Gemäß dem Jordanschen Kurvensatz partitioniert die Kurve $C^\star$ die Ebene in zwei Bereiche $\text{left}(C^\star)$ und $\text{right}(C^\star)$. Wir betrachten die drei möglichen Fälle:

Fall 1: s und t liegen auf verschiedenen Seiten von $C^\star$. Dann kreuzt P die Kurve $C^\star$ in beiden Richtungen gleich oft. Dies ist äquivalent dazu, dass P die gleiche Anzahl von Kanten in C und $\text{rev}(C)$ enthält. Es folgt also $\pi(C) = 0$.

Fall 2: $s \in \text{left}(C^\star)$ und $s \in \text{right}(C^\star)$. Dann kreuzt P die Kurve $C^\star$ einmal mehr von links nach rechts als von rechts nach links. Dies ist äquivalent dazu, dass P eine Kante mehr aus C als aus $\text{rev}(C)$ enthält, also $\pi(C) = 1$.

Fall 3: $s \in \text{right}(C^\star)$ und $s \in \text{left}(C^\star)$. Dieser Fall ist symmetrisch zum vorigen Fall, es folgt $\pi(C) = -1$.

Es bleibt zu zeigen, dass ein Kozykel genau dann ein s–t-Schnitt ist, wenn $\pi(C) = 1$. Dies gilt, da $\pi(C) = 1$ genau dann, wenn Fall 2 vorliegt. Dies ist genau dann der Fall wenn C ein s–t-Schnitt ist. □

Wir betrachten nun den Fluss $\lambda \cdot \pi$, der jeder Kante $e \in C$ einen Fluss von $\lambda \cdot \pi(e)$ zuweist. Sei $G_\lambda := G_{\lambda\pi}$ das Residualnetzwerk bezüglich dieses Flusses. Das heißt, G_λ ist der Graph G mit der residualen Kapazitätsfunktion $c(\lambda, e) := c(e) - \lambda \cdot \pi(e)$. Bezeichne $G^\star_\lambda$ das duale Residualnetzwerk, also das gerichtet dual $G^\star$ von G mit der Kostenfunktion $c(\lambda, e^\star) := c(\lambda, e)$

Lemma 2. *In G existiert genau dann ein gültiger s–t-Fluß mit Wert λ wenn das Residualnetzwerk $G^\star_\lambda$ keinen negativen Kreis enthält.*

Beweis. Wir betrachten zunächst den Fall, dass $G^\star_\lambda$ einen negativen Kreis $C^\star$ enthält. Dann lassen sich die Kosten dieses Kreises wie folgt zerlegen:

$$0 > c(\lambda, C^\star) = \sum_{e \in C} c(\lambda, e) = \sum_{e \in C} c(e) - \lambda\pi(C) \ .$$

Da $\lambda \geq 0$ und $\sum_{e \in C} c(e) \geq 0$ folgt $\pi(C) > 0$. Nach Lemma 1 gilt also $\pi(C) = 1$ und C ist ein s–t-Schnitt. Einsetzen liefert von $\pi(C) = 1$ liefert zudem

$$\sum_{e \in C} c(e) < \lambda \ .$$

Damit ist C ein s–t-Schnitt mit $c(C) < \lambda$, es kann also keinen gültigen Fluss mit Wert λ geben.

Nehmen wir nun umgekehrt an, dass $G^\star_\lambda$ keine negativen Kreise enthält, kürzeste Wege in $G^\star_\lambda$ also wohldefiniert sind. Wähle einen beliebigen Knoten o in $G^\star_\lambda$ als Ursprung. Für jeden Dualknoten p bezeichne $\text{dist}(\lambda, p)$ die Länge des kürzesten Weges in $G^\star_\lambda$ von o zu p. Wir definieren die Funktion

$$\Phi(\lambda, e) := \text{dist}(\lambda, \text{head}(e^\star)) - \text{dist}(\lambda, \text{tail}(e^\star)) + \lambda \cdot \pi(e) \ .$$

Wir zeigen nun, dass Φ ein gültiger s–t-Fluß mit Wert λ ist. Für jeden Knoten v gilt $\sum_w \Phi(v \to w) = \sum_w \lambda \cdot \pi(v \to w)$, da die aus v ausgehenden Kanten einen gerichteten Kreis im Dualgraphen bilden und sich die Distanz-Terme $\text{dist}(\lambda, \cdot)$ daher aufheben. Damit folgt direkt, dass Φ ein Fluss mit Wert λ ist.

Es bleibt zu zeigen, dass Φ gültig ist, also $\Phi(\lambda, e) \leq c(e)$ für alle $e \in E$. Hierzu definieren wir für jede Kante e den Schlupf $\text{slack}(\lambda, e^\star) := \text{dist}(\lambda, \text{tail}(e^\star)) + c(\lambda, e) - \text{dist}(\lambda, \text{head}(e^\star))$. Eine leichte Rechnung zeigt, dass $\text{slack}(\lambda, e) = c(e) - \Phi(\lambda, e)$. Die Bedingung, $\Phi(\lambda, e) \leq c(e)$ lässt sich damit äquivalent als $\text{slack}(\lambda, e) \geq 0$. Wäre nun $\text{slack}(\lambda, e) < 0$, so folgt $\text{dist}(\lambda, \text{head}(e^\star)) > \mathtt{dist}(\lambda, \text{tail}(e^\star)) + c(\lambda, e^\star)$. Dies widerspricht aber der Definition der Distanzen über kürzeste Wege. $\square$

1.2 Flüsse als Parametrisches Kürzeste-Wege-Problem

Um einen maximalen gültigen s–t-Fluss zu finden, genügt es also den maximalen Wert λ_{max} zu bestimmen, für den $G^\star_\lambda$ keinen negativen Kreis enthält. Kennen wir zudem die zugehörigen Werte $\text{dist}(\lambda_{\text{max}}, \cdot)$ in $G^\star_\lambda$, so lässt sich hieraus in $O(n)$ Zeit ein zugehöriger gültiger s–t-Fluss $\Phi(\lambda_{\text{max}}, \cdot)$ mit Wert λ bestimmen.

Das Problem in einem Graphen, dessen Kantengewichte lineare Funktionen abhängig von einem Parameter λ sind, den maximalen Wert λ_{max} zu bestimmen, sodass der Graph keine negativen Kreise enthält, heißt auch *parametrisches Kürzeste-Wege-Problem*. Hier ist die Situation insofern speziell als sowohl der Graph $G^\star$ und die Menge $P^\star$ der Kanten, die von λ abhängen eine besondere Struktur haben. Zudem hat die parametrische Kostenfunktion $c(\lambda, \cdot)$ nur drei verschiedene λ-Koeffizienten, nämlich $-1, 0$ und 1.

Für einen festen Wert λ, für den $G^\star_\lambda$ keinen negativen Kreis enthält, bezeichne T_λ den Kürzeste-Wege-Baum in $G^\star_\lambda$ mit Wurzel o. Da die Eingabekapazitäten nicht-negativ sind, sind kürzeste Wege in $G^\star_0$, und somit auch T_0 wohldefiniert. Zudem nehmen wir an, dass unsere Kapazitäten insofern *generisch* sind, als dass T_λ eindeutig definiert ist für alle Werte von 0 bis λ_{max}, außer für eine endliche Menge von *kritischen Werte*. Unser allgemeines Vorgehen sieht nun wie folgt aus.

<u>PLANARMAXFLOW(G,c,s,t):</u>

1. Berechne T_0.
2. Verwalte T_λ währen λ kontinuierlich von 0 bis λ_{max} erhöht wird.
3. Berechne $\Phi(\lambda_{\text{max}}, \cdot)$ aus $T_{\lambda_{\text{max}}}$.

In T_λ hat jeder Knoten $p \neq o$ einen eindeutigen Vorgänger $\text{pred}(\lambda, p)$. Wir sagen eine duale Kante $e^\star$ ist *straff bei* λ wenn $\text{slack}(\lambda, e^\star) = 0$. Außer bei kritischen λ-Werten ist eine Kante straff bei λ genau dann wenn sie zu T_λ gehört. Bei jedem kritischen λ-Wert wird eine duale Kante $p \rightarrow q$ straff und tritt in T_λ ein. Dabei ersetzt sie die vorherige Kante $\text{pred}(\lambda, p) \rightarrow q$ (außer bei $p = o$). Wir bezeichnen ein solches Ereignis als *Pivot-Schritt*.

Wiederum stellt unsere Annahme über die Generizität der Kapazitätsfunktion sicher, dass für jeden kritischen λ-Wert genau eine Nicht-Baumkante straff wird.

Für jeden dualen Knoten p bezeichne $\text{path}(\lambda, p)$ den eindeutigen Pfad von o zu p in T_λ. Für eine duale Kante $p \rightarrow q$ bezeichne $\text{cycle}(p \rightarrow q)$ den Kreis, den gerichteten Kreis, der von $p \rightarrow q$ zusammen mit T_λ induziert wird. Das heißt, $\text{cycle}(p \rightarrow q)$ ist die Konkatenation von $\text{path}(\lambda, p)$, $p \rightarrow q$ und $\text{rev}(\text{path}(\lambda, q))$. Wir charakterisieren zunächst den gesuchten Wert λ_{max}.

Lemma 3. *λ_{max} ist der erste kritische Wert von λ dessen Pivot-Ereignis einen gerichteten Kreis in T_λ erzeugt.*

Beweis. $\square$ Wird noch ergänzt

Diese Charakterisierung lässt sich äquivalent im (ungerichteten) primalen Graphen ausdrücken. Wir bezeichnen eine (ungerichtete) primale Kante e als *locker* bei λ wenn weder $e^\star$ noch $\text{rev}(e^\star)$ bei λ straff sind. Sei L_λ der Graph, der aus allen lockeren Kanten von G besteht. Außer bei kritischen Werten von λ, ist L_λ als Komplement eines Spannbaums im Dualgraphen ein Spannbaum von G (siehe Übung!). Zusammen bilden T_λ und L_λ eine sogenannte Baum-Kobaum-Zerlegung.

Enthält nun T_λ einen Kreis, so entspricht dies einem Schnitt in G, das heißt L_λ ist dann nicht mehr zusammenhängend. Daher lässt sich das obige Kriterium äquivalent umschreiben.

Lemma 4. *$\lambda_{\max}$ ist der kleinste kritische λ-Wert dessen Pivot-Ereignis L_λ unzusammenhängend macht.*

Als nächstes müssen wir, um λ Schrittweise zu erhöhen ohne kritische Werte zu verpassen, genauer herausfinden welche Kanten ein Pivot-Ereignis auslösen können und wann sie dies tun. Es ist klar, dass nur eine Kante, deren Schlupf mit steigendem λ abnimmt straff werden kann und dann einen Pivot-Schritt auslösen kann. Wir nennen solche Kanten *aktiv*. Der primale Spannbaum L_λ enthält einen eindeutigen ungerichteten s–t-Pfad LP_λ. Wir zeigen, dass die aktiven Kanten genau die Kanten von LP_λ sind.

Lemma 5. *Eine duale Kante $e^\star$ ist aktiv bei λ genau dann wenn e eine Kante von LP_λ ist.*

Beweis. □

Wird noch ergänzt.

Hieraus ergibt sich direkt ein Algorithmus, der λ, beginnend mit $\lambda = 0$, jeweils zum nächsten kritischen λ-Wert erhöht. Hierzu suchen wir einfach in jedem Schritt die Kante von LP_λ mit minimalem Schlupf Δ. Erhöht man λ nun genau um Δ, so wird keine andere Kante zuvor straff (sonst müßte sie aktiv sein, läge damit auf LP_λ und hätte damit geringeren Schlupf als Δ). Es wird also kein kritischer λ-Wert verpasst. Der Wert $\lambda + \Delta$ ist kritisch, und wir passen dementsprechende T_λ und L_λ an. Werden dabei s und t getrennt, so ist der Wert $\lambda_{\max}$ erreicht. Anderenfalls wird das Vorgehen fortgesetzt..

Pseudocode wird noch ergänzt

1.3 Details der Implementierung

Eine naive Implementierung des Algorithms benötigt eine deutlich höhere Laufzeit als $O(n \log n)$. Wir zeigen, dass sich durch Verwendung geeigneter Datenstrukturen jeder Schleifendurchlauf der while-Schleife mit Laufzeit $O(\log n)$ implementieren lässt. Hierzu verwaltet der Algorithmus folgende Datenstrukturen.

1. Den Spannbaum L_λ der lockeren Kanten.
2. Den Vorgänger $\mathrm{pred}(\lambda, q)$ für jeden dualen Knoten q.
3. Den Schlupf-Wert $\mathrm{slack}(\lambda, e^\star)$ jeder dualen Kante $e^\star$.

Den Baum L_λ, zusammen mit den Schlupf-Werten der entsprechenden Dualkanten, verwalten wir in einem sogenannten top-tree. Ein top-tree verwaltet einen gewichteten Wald und bietet die folgenden Operationen jeweils mit einer amortisierten Laufzeit von $O(\log n)$ an.

A) Kanten entfernen.
B) Kanten einfügen.
C) Abfragen ob sich zwei gegebene Knoten in derselben Komponente befinden.
D) “Darstellen” eines Pfades zwischen zwei gegebenen Knoten derselben Komponente.
E) Auffinden der Kante mit minimalem Gewicht auf dem dargestellten Pfad.
F) Addieren eines Wertes auf alle Kantengewichte des dargestellten Pfades.

Die Vorgänger-Informationen $\mathrm{pred}(\lambda, \cdot)$ werden als einfache Zeiger gespeichert. Wir wählen weder den initialen Pfad P noch verwalten wir explizit den Wert von λ. Letzteres wird implizit durch Absenken der Schlupf-Werte repräsentiert. Es ist nicht schwer zu sehen, dass sich jeder Schleifendurchlauf der while-Schleife des mittels top-trees eine Laufzeit von $O(\log n)$ benötigt. Die Initialisierung der Datenstrukturen kann mittels Dijkstras Algorithmus in $O(n \log n)$ Zeit erfolgen. Die abschließende Berechnung von $\Phi(\cdot)$ erfolgt dann in $O(n)$ Zeit. Insgesamt ergibt sich daher eine

Laufzeit von $O((n + N) \log n)$, wobei N die Anzahl der Pivot-Ereignisse, also der kritischen λ-Werte, bezeichnet. Um auf eine Gesamtlaufzeit von $O(n \log n)$ zu kommen, genügt es zu zeigen, dass $N \in O(n)$ gilt.

1.4 Anzahl der Pivot-Ereignisse

Um zu zeigen, dass $N \in O(n)$ machen wir eine weitere Annahme, nämlichj dass t inzident zur äußeren Facette $o^\star$ ist. Für diesen Fall zeigen wir, dass tatsächliche jede duale Kanten höchstens ein Pivot-Ereignis auslöst, also höchstens einmal in den Baum T_λ aufgenommen wird.

Sei nun Π ein beliebiger Pfad von o zu einem Knoten p des Dualgraphen $G^\star$. Wir bezeichnen $\pi(\Pi) = \sum_{e^\star \in \Pi} \pi(e)$ als *Kreuzungszahl* von Π.

Jedes Mal wenn sich ein Pfad $\mathrm{path}(\lambda, p)$ ändert, geschieht dies dadurch, dass eine neue Kante, deren Gewicht mit λ absinkt, in den Pfad aufgenommen wurde. Dass diese Kante aufgenommen wurde, bedeutet aber genau, dass $\mathrm{path}(\lambda, p)$ den Pfad P ein weiteres Mal kreuzt.

Lemma 6. *Jede Mal wenn sich* $\mathrm{path}(\lambda, p)$ *ändert, erhöht sich seine Kreuzungszahl um 1.*

Wir betrachten also für jeden Knoten p Pfade $\mathrm{path}(\lambda, p)$ mit wachsender Kreuzungszahl. Für einen beliebigen dualen Knoten p und eine ganze (nicht notwendigerweise positive) Zahl i bezeichne $\mathrm{path}_i(p)$ den kürzesten Weg von o zu p in $G^\star = G_0^\star$. Ist $\mathrm{path}(\lambda, p) = \mathrm{path}_i(\lambda, p)$, so wird, gemäß Lemma 6, bei der nächsten Änderung von $\mathrm{path}(\lambda, p)$ dieser Pfad durch $\mathrm{path}_{i+1}(\lambda, p)$ ersetzt. Es ist daher zweckmäßig die doppelt unendliche Sequenz der Pfad $\mathrm{path}_i(\lambda, p)$ für $i \in \mathbb{Z}$ zu betrachten. Um stattdessen normale kürzeste Wege betrachten zu können, konstruieren wir einen geeigneten (unendlichen) planaren Graphen $\bar{G}^\star$, in dem diese Wege zu normalen kürzesten Wegen korrespondieren. Dies geht wie folgt. Wir definieren $\bar{G}^\star = (\bar{V}^\star, \bar{E}^\star)$ mit

$$\begin{aligned}
\bar{V}^\star &:= \{p_i \mid p \in V^\star, i \in \mathbb{Z}\} \\
\bar{E}^\star &:= \{p_i \to q_{i+\pi(p\to q)} \mid p \to q \in E^\star\}
\end{aligned}$$

Informell besteht $\bar{G}^\star$ aus einer doppelt unendlichen Sequenz $\dots, G_{-1}^\star, G_0^\star, G_1^\star, G_2^\star, \dots$ von Kopien von $G^\star$. Für einen dualen Knoten p bezeichnet p_i die entsprechende Kopie von p in $G_i^\star$. Für jede Kante $p \to q$ in $P^\star$ ersetzen wir nun $p_i \to q_i$ durch $p_i \to q_{i+1}$ und symmetrisch $q_i \to p_i$ durch $q_i \to p_{i-1}$. Das heißt, wir wechseln von einem Graphen zum nächsten wenn wir dabei den Pfad P kreuzen. Die Kantengewichte von $\bar{G}^\star$ definieren wir durch $c(p_i \to q_j) = c(p \to q)$.

Es gibt eine natürliche Projektion $\bar{\omega}\colon \bar{G}^\star \to G^\star$: $\bar{\omega}(p_i) = p$ und $\bar{\omega}(p_i \to q_j) = p \to q$. Das Urbild $\bar{\omega}^{-1}(\Pi)$ eines Pfades Π in $G^\star$ ist dann eine doppelt unendliche Menge von Pfaden in $\bar{G}^\star$, die sogenannten *Lifts* von Π. Ist p ein Pfad von p nach q, so gibt es für jedes $i \in \mathbb{Z}$ einen Lift von Pi, der bei p_i startet und bei $q_{i+\pi(\Pi)}$ endet. Die Facetten $s^\star$ und $t^\star$ werden auf diese Weise zu zwei unbeschränkten Facetten geliftet, jede andere Facette wird zu einer doppelt unendlichen Sequenz von Facetten in $\bar{G}^\star$ geliftet.

Bezeichne nun $\bar{\mathrm{path}}(p_i, q_j)$ den kürzesten Weg in $\bar{G}^\star$ von p_i nach q_j. Nach Konstruktion von $\bar{G}^\star$ ist für jeden Knoten p und jedes $j \in \mathbb{Z}$ der Pfad $\mathrm{path}_j(p)$ die Projektion von $\bar{\mathrm{path}}(o_0, p_j)$. Die kürzesten Wege in $G^\star$ mit Kreuzungszahl j entsprechen also kürzesten Wegen in $\bar{G}^\star$ von o_0 zu p_j. Natürlich lässt sich der Index des Urbildes verschieben, das heißt, auch $\bar{\mathrm{path}}(o_i, p_{i+j})$ mit $i \in \mathbb{Z}$ ist ein solches Urbild. Folglich ist jeder kürzeste Weg $\mathrm{path}(\lambda, p)$ die Projektion eines kürzesten Weges in $\bar{G}^\star$.

Tritt nun eine Kante $p \to q$ mehrmals in den Baum T_λ ein, so bedeutet dies, dass es zwei Indizes i und j mit $i < j$ gibt, sodass $\mathrm{path}_i(q)$ und $\mathrm{path}_j(q)$ mit $p \to q$ enden, aber $\mathrm{path}_{i-1}(q)$ und $\mathrm{path}_{j-1}(q)$ nicht. Betrachten wir die Urbilder der Pfade $\mathrm{path}_{i-1}(p)$ und $\mathrm{path}_{j-1}(p)$ in $\bar{G}^\star$, so verschoben, dass beide in p_0 enden, so entsteht, zusammen mit einer Kante von o_{-j} zu o_{-i} ein Kreis Γ. Wir zeigen, dass der Knoten q_0 weder innerhalb, noch außerhalb, noch auf Γ liegen kann. Dies ist ein Widerspruch dazu dass eine Kante mehrfach in den Kürzeste-Wege-Baum eintritt.

Lemma 7. *Jede duale Kante* $\mathsf{p} \to \mathsf{q}$ *löst höchstens ein Pivot-Ereignis aus.*

Beweis. □

Beweis wird noch ergänzt!