

RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT BONN
INSTITUT FÜR INFORMATIK I



- Diplomarbeit -
Kompetitive Routenplanung
bei ausfallenden Kanten

Sebastian Jacobi & Manuel Wedemeier

Betreuer: Prof. Dr. Rolf Klein

Eidesstattliche Erklärungen

Hiermit versichere ich, dass ich die in der Übersicht auf Seite 155 angegebenen Bestandteile dieser Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, habe ich als solche gekennzeichnet. Weiter erkläre ich, die Diplomarbeit in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt zu haben.

Bonn, 17.12.2007

(Sebastian Jacobi)

Hiermit versichere ich, dass ich die in der Übersicht auf Seite 155 angegebenen Bestandteile dieser Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, habe ich als solche gekennzeichnet. Weiter erkläre ich, die Diplomarbeit in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt zu haben.

Bonn, 17.12.2007

(Manuel Wedemeier)

Inhaltsverzeichnis

1	Einleitung	7
1.1	Motivation und Problemstellung	7
1.2	Inhalt der Arbeit	9
1.3	Variationen des Canadian Traveller Problem (CTP)	11
1.3.1	Recoverable-Varianten des CTP	11
	Stochastisches Recoverable CTP (sr-CTP)	12
	Deterministisches Recoverable k -CTP (dr- k -CTP)	13
1.3.2	Das k -Canadian Traveller Problem (k -CTP)	15
1.3.3	Das k -Vital Edges Problem (k -VEP)	16
1.4	Grundlagen	16
2	Komplexitätsanalyse des CTP	19
2.1	Reduktion von DVG auf QSAT	21
2.2	Transformation des DVG zum CTP	28
3	Die Strategien Backtrack und Greedy	37
3.1	Darstellung der Strategien	37
3.1.1	Die Strategie Backtrack	38
3.1.2	Die Strategie Greedy	39
3.2	Backtrack und Greedy auf allgemeinen Graphen	40
3.2.1	Die Strategie Backtrack	41
3.2.2	Die Strategie Greedy	43
3.3	Backtrack und Greedy auf speziellen Graphklassen	46
3.3.1	Vollständiger Graph	46
3.3.2	Vollständiger (m,n) -Gittergraph	46
	Die Strategie Backtrack	47
	Die Strategie Greedy	50
3.3.3	Allgemeiner (m,n) -Gittergraph	61
	Die Strategie Backtrack	61
	Die Strategie Greedy	62
3.4	Backtrack und Greedy auf Spannern	65
3.4.1	t -Spanner	65
	Die Strategie Backtrack	65

	Die Strategie Greedy	70
	Backtrack oder Greedy - welche Strategie ist im worst-case schneller?	91
3.4.2	(t, l) -Self-Spanner	94
	Die Strategie Backtrack	94
	Die Strategie Greedy	98
3.4.3	t -Self-Spanner	105
	Die Strategie Backtrack	106
	Die Strategie Backtrack (auf schlichten Graphen)	111
	Die Strategie Greedy	113
4	Untere Schranken deterministischer Strategien	125
4.1	Allgemeiner Graph	125
4.2	Vollständiger Graph	126
4.3	Vollständiger (m, n) -Gittergraph	126
4.4	Allgemeiner (m, n) -Gittergraph	131
4.5	t -Spanner	135
4.6	(t, l) -Self-Spanner	143
4.7	t -Self-Spanner	144
5	Untere Schranke randomisierter Strategien	147
6	Ergebnisse und Ausblick	149
6.1	Zusammenfassung der Ergebnisse	149
6.2	Bewertung der Ergebnisse	150
6.3	Ausblick	153
	Aufteilung der Diplomarbeit	155
	Literaturverzeichnis	157
	Index	159

Kapitel 1

Einleitung

1.1 Motivation und Problemstellung

In den 60er-Jahren haben sich die Anfänge des heutigen Internets herausgebildet. Zu Beginn war es das U.S. Militär, in enger Zusammenarbeit mit dem MIT (Massachusetts Institute of Technology), das zu Zeiten instabiler Telefonleitungen Anstrengungen unternommen hat, einen neuen Ansatz zur Rechnerkoppelung zu entwickeln, der über ein höheres Maß an Robustheit verfügt und so die technischen Mängel der Telefonverbindungen kompensieren konnte. In diesem Zeitraum wurde das so genannte packet-switching entworfen, bei dem größere Informationsmengen in kleinen Paketen auf verschiedenen Wegen über ein Computer-Netzwerk versandt werden können. Dies war die Geburtsstunde des Advanced Research Projects Agency Network (ARPRANET), welches unter der Leitung des Department of Defense Advance Research Project Agency (ARPA) entstand (vgl. [IN01] und [IN02]). Aus diesem Netzwerk ging später das heute bekannte Internet hervor. Auf diese Weise sollten die sehr knapp bemessenen Rechenleistungen der einzelnen Forschungseinrichtungen kombiniert werden, um so einen Leistungszuwachs zu erzielen. Im Jahr 1969 bildeten die vier Knotenpunkte der University of California at Los Angeles, Stanford Research Institute, University of California at Santa Barbara und die University of Utah, die mit dem Netzwerk der DARPA verbunden wurden, den Anfang. Ein Meilenstein in der Geschichte der Computertechnologie war geboren. Das Verschicken von Daten auf diesem dezentralen Netzwerk und der dadurch auftretende Netzwerkverkehr war nicht vorhersehbar, da die Datenauslastung auf den einzelnen Teilstrecken inhomogen auftrat. Dies führte zur Notwendigkeit, Strategien zu entwickeln, mit denen die Nachrichtenpakete schnellstmöglich zu ihrem Bestimmungsort gelangten.

Zur Lösung eines verwandten Problems haben gegen Ende der 80er-Jahre C.H. Papadimitriou und M. Yannakakis [PY89] in ihrer Arbeit verschiedene Varianten des Kürzeste-Wege-Problems ohne vollständige Karte untersucht. Um die Sprache dieses Problems aufzugreifen, stellen wir uns dabei vor, dass ein Reisender mittels einer Landkarte eine Route für einen Weg zwischen zwei Orten im verschneiten Winter des nördlichen Kanada pla-

nen will. Da die Wetterbedingungen unbeständig sind, kann es immer wieder vorkommen, dass Straßenabschnitte zugeschnitten und somit unpassierbar sind. Die Information darüber erhält der Reisende jedoch immer erst an einer zur Straße inzidenten Kreuzung. Die automatisierte Routenplanung, also das Finden einer Strategie, die einen möglichst kurzen passierbaren Weg zwischen den beiden Orten berechnet, wird als Canadian Traveller Problem bezeichnet. Unter einer Strategie verstehen wir in diesem Zusammenhang eine Abfolge von Explorationsschritten, die jedes lösbare Problem in endlicher Zeit löst. Handelt es sich um eine deterministische Strategie so dürfen Unterschiede im Lösungsweg nur vorkommen, wenn die Probleminstanzen verschieden sind. Um das Problem zu formalisieren, definieren wir zuerst, was in diesem Zusammenhang unter einer Strategie zu verstehen ist.

Definition 1.1. Gegeben sei ein ungerichteter Graph $G = (V, E)$ sowie ein Startknoten $s \in V$ und ein Zielknoten $t \in V$. Eine **Graph-Strategie** ist eine Funktion S_G , die jeder Kantenteilmenge $E' \subseteq E$ einen in s startenden und in t , mit $s \neq t$, endenden endlichen Weg zuordnet, der nur Kanten aus $E \setminus E'$ traversiert. Die Wege einer Strategie zu zwei unterschiedlichen Ausfallmengen E_1, E_2 dürfen sich erst ab einem Knoten unterscheiden, der inzident zu einer Kante aus $E_1 \cup E_2$ ist.

Mit dieser Definition einer Strategie können wir nun das Problem formal beschreiben.

Definition 1.2. Gegeben sei ein ungerichteter Graph $G = (V, E)$ sowie eine Gewichtsfunktion $w : E \rightarrow [0, \infty)$. Jeder Kante $e \in E$ wird mittels w ein Kostenmaß zugeordnet. Eine Kante kann sich jedoch bei Erreichen eines ihrer inzidenten Knoten als blockiert herausstellen. Es sei $G_i = (V, E \setminus \{e_1, \dots, e_i\})$ der Graph, in dem die ersten i entdeckten blockierten Kanten $e_1, \dots, e_i$ entfernt wurden. Es wird dabei garantiert, dass auch im Restgraphen G_k ein s - t -Pfad mit endlichen Kosten existiert. Das **Canadian Traveller Problem (CTP)** besteht darin, eine Strategie zu finden, die von einem festen Startknoten $s \in V$ zu einem festen Zielknoten $t \in V$ einen Pfad in dem ihr zum jeweiligen Zeitpunkt bekannten Graphen G_i berechnet und den worst-case-Quotienten aus Wegkosten und Kosten eines optimalen Pfades von s zu t in G_k minimiert.

Diese Definition lässt sich zu folgendem Entscheidungsproblem erweitern:

Definition 1.3. Gegeben sei ein ungerichteter Graph $G = (V, E)$, eine Gewichtsfunktion $w : E \rightarrow [0, \infty)$ sowie ein vorgegebener Quotient $r \geq 1$, der das Verhältnis der durch eine Strategie verursachten Kosten zu denen einer optimalen Lösung darstellt. Jeder Kante $e \in E$ wird mittels w ein Kostenmaß zugeordnet. Das **CTP-Entscheidungsproblem** ist die Frage, ob es eine Strategie gibt, die das CTP auf G löst und dabei den Faktor r bezüglich eines kürzesten Weges in G' gemäß Definition 1.2 einhält.

Um den Zusammenhang zu verdeutlichen übertragen wir das Problem auf unsere ursprüngliche Thematik. Dabei stellen die einzelnen Server der Forschungseinrichtungen die Knoten (Straßenkreuzungen) zwischen den diese Universitäten verbindenden Kanten (Netzwerkleitungen) dar. Eine Strategie, die nach einem möglichst unkomplizierten Verfahren einen stets schnellen Weg zwischen zwei Knoten ermittelt, garantiert gleichermaßen einen

schnellen Informationsaustausch zwischen den Bestimmungsorten. Wie schnell können solche Informationen im schlimmsten Fall, d.h. bei Ausfall wichtiger Netzwerkleitungen, versandt werden. Gibt es bestimmte Strukturen eines Netzwerkes, die den Datenaustausch begünstigen? Hilft Randomisierung, um eine Strategie zu verbessern? An dieser Stelle verspricht ein genaueres Verständnis des Canadian Traveller Problem hilfreich zu sein.

1.2 Inhalt der Arbeit

In Kapitel 1.3 stellen wir die Ergebnisse von Bar-Noy und Schieber (vgl. [BS91]) vor, die sich mit Variationen des Canadian Traveller Problem beschäftigt haben. Komplexität des Canadian Traveller Problem wird in Kapitel 2 detailliert untersucht. Es wird sich zeigen, dass bereits das CTP-Entscheidungsproblem *PSPACE*-vollständig ist. Die weiteren Kapitel dieser Arbeit stützen sich auf ein Grundgerüst an Definitionen, das im letzten Teil dieses ersten Kapitels zu finden ist. Werden Definitionen nicht kapitelübergreifend verwendet, so sind diese einleitend zu den entsprechenden Kapiteln platziert. In Kapitel 3 werden wir einen konservativen sowie ein progressiven Ansatz zur Lösung des k -Canadian Traveller Problem untersuchen, das wie folgt definiert ist:

Definition 1.4. *Gegeben sei ein ungerichteter Graph $G = (V, E)$, eine Gewichtsfunktion $w : E \rightarrow [0, \infty)$ sowie eine Konstante $k \in \mathbb{N}$. Jeder Kante $e \in E$ wird mittels w ein Kostenmaß zugeordnet. Maximal k Kanten können sich jedoch bei Erreichen einer ihrer inzidenten Knoten als blockiert herausstellen. Es sei $G_i = (V, E \setminus \{e_1, \dots, e_i\})$ der Graph, in dem die ersten i entdeckten blockierten Kanten $e_1, \dots, e_i$ entfernt wurden. Jedoch ist sichergestellt, dass auch in G_k noch ein Pfad zwischen s und t existiert. Das **k -Canadian Traveller Problem (k -CTP)** besteht darin, eine Strategie zu finden, die von einem festen Startknoten $s \in V$ zu einem festen Zielknoten $t \in V$ einen Pfad in dem ihr zum jeweiligen Zeitpunkt bekannten Graphen G_i berechnet und den worst-case-Quotienten aus Wegkosten und Kosten eines optimalen Pfades von s zu t in G_k minimiert.*

Bei diesem Problem stellt eine vorab bekannte Konstante k die Anzahl maximal ausfallender Kanten dar, welche von essentieller Wichtigkeit für die Konstruktion geeigneter Szenarien ist. Um die Güte einer Strategie erfassen zu können, benötigen wir diese Information, da die Güte von der maximalen Anzahl blockierter Kanten abhängt und nur hierdurch eine Vergleichsmöglichkeit zwischen unterschiedlichen Szenarien möglich ist.

Wir betrachten zunächst folgende Definition:

Definition 1.5. *Eine **Strategie** S ist eine Menge von Graph-Strategien, die zu jeder Instanz $P \in \Pi$ eines Problems Π mindestens eine Graph-Strategie $S_G \in S$ enthält, die P korrekt löst.*

Wir führen nun die beiden Strategien *Backtrack* und *Greedy* ein, die die Grundlage von Kapitel 3 darstellen. Beide Strategien verfolgen das Ziel, in einem Graphen $G_k :=$

$(V, E \setminus \{e_1, \dots, e_k\})$ von einem Startknoten $s \in V$ zu einem Zielknoten $t \in V$ zu gelangen und dabei relativ zur optimalen Lösung eine möglichst kurze Strecke zurückzulegen. Zu Beginn der Berechnungen kennen beide Strategien jedoch nur den Graphen $G = (V, E)$ und erfahren von einer blockierten Kante $e \in \{e_1, \dots, e_k\}$ erst, wenn sie im Sinne einer Online-Exploration an einem ihrer inzidenten Knoten ankommen. Von s aus berechnen *Backtrack* und *Greedy* einen kürzesten Pfad $\pi_G^{\min}(s, t)$ zu t in G und folgen diesem, bis sie t erreichen oder zu einem Knoten $v_1 \in V$ kommen, der inzident zu einer blockierten Kante $e \in \{e_1, \dots, e_k\}$ des Pfades $\pi_G^{\min}(s, t)$ ist. *Backtrack* folgt von v_1 aus dem bisher zurückgelegten Teil von $\pi_G^{\min}(s, t)$ zurück zum Startknoten s und berechnet anschließend einen kürzesten Pfad $\pi_{G_1}^{\min}(s, t)$ zu t in $G_1 = (V, E \setminus \{e_1\})$. Diesem folgt die *Backtrack*-Strategie bis zum Zielknoten t oder bis zu einem Knoten v_2 , der inzident zu einer blockierten Kante $e \in \{e_1, \dots, e_k\}$ des Pfades $\pi_{G_1}^{\min}(s, t)$ ist. Auf diese Weise führt die Strategie ihre Berechnung fort. *Greedy* berechnet von v_1 aus direkt einen kürzesten Pfad $\pi_{G_1}^{\min}(v_1, t)$ zu t in $G_1 = (V, E \setminus \{e_1\})$ und folgt diesem bis zum Zielknoten t oder bis zu einem Knoten v_2 , der inzident zu einer blockierten Kante $e \in \{e_1, \dots, e_k\}$ des Pfades $\pi_{G_1}^{\min}(v_1, t)$ ist und so weiter. Dieser Vorgang kann sich bei beiden Strategien im schlimmsten Fall k mal wiederholen, bis bei der $(k+1)$ -ten Iteration schließlich der gesamte Graph G_k bekannt ist und ein kürzester Weg zu t berechnet wird, der keine weiteren blockierten Kanten mehr enthalten kann.

Da in einem Graphen die von *Backtrack* und *Greedy* berechneten kürzesten Pfade nicht zwingend eindeutig sind, handelt es sich, wie die folgende Definition zeigt, um *nicht eindeutige* Strategien:

Definition 1.6. Eine **nicht eindeutige Strategie** S ist eine Menge von Graph-Strategien, die zu jeder Instanz $P \in \Pi$ eines Problems Π mindestens zwei unterschiedliche Graph-Strategien $S_G, S'_G \in S$ enthält, die P korrekt lösen.

Neben dem allgemeinen Graphen werden auch spezielle Graphklassen betrachtet und dargelegt, ob bzw. inwieweit, sich die Güte der Strategien verbessern. Wir haben uns hierbei für Graphklassen entschieden, die einen praktischen Bezug zum eingangs besprochenen Nachrichtenversand über ein Computer-Netzwerk herstellen. Die Wahl fiel auf den vollständigen Graphen (Kapitel 3.3.1), den vollständigen (m, n) -Gittergraphen (Kapitel 3.3.2), den allgemeinen (m, n) -Gittergraphen (Kapitel 3.3.3), die Klasse der t -Spanner (Kapitel 3.4.1), die der (t, l) -Self-Spanner (Kapitel 3.4.2) sowie abschließend die Teilklasse der t -Self-Spanner (Kapitel 3.4.3).

Um die Ergebnisse besser in einen Gesamtzusammenhang einordnen zu können, werden in Kapitel 4 Ergebnisse für eine bestmögliche Strategie bezüglich der in Kapitel 3 behandelten Graphklassen untersucht. Auf diese Art können wir deutlich machen, wie gut oder wie schlecht die in Kapitel 3 untersuchten Strategien einzuschätzen sind. Im Zusammenhang mit Online-Algorithmen stellt sich auch immer die Frage, inwieweit Randomisierung das Problem verändert. Haben Bar-Noy und Schieber in [BS91] untersucht, wie sich die

Komplexität des Problems bei Randomisierung der Kantenausfälle darstellt, so untersuchte Stephan Westphal in [SW07] die Güte von randomisierten Strategien. Diesem Thema und damit der Vorstellung von Westphals Ergebnis wenden wir uns in Kapitel 5 zu. Eine bewertende Stellungnahme in Kapitel 6 bildet den Abschluss unserer Arbeit. Diese wird durch einen Ausblick auf weitere, aus unserer Sicht sinnvolle Denkanstöße im Hinblick auf nachfolgende Untersuchungen komplettiert.

1.3 Variationen des Canadian Traveller Problem (CTP)

Anfang der 90er-Jahre griffen Bar-Noy und Schieber das in Abschnitt 1.1 definierte Canadian Traveller Problem (CTP) in ihrer Arbeit [BS91] wieder auf, indem sie zahlreiche Variationen dieses Problems untersuchten. Dabei schränkten sie verschiedene Freiheitsgrade ein und versuchten so, die Komplexität zu reduzieren.

Algorithmen zum Lösen des CTP sowie seiner Variationen müssen ohne vollständige (Vorab-) Informationen auskommen, da im Vorfeld nicht bekannt ist, welche Kanten des gegebenen Graphen blockiert sind. Zur Lösung dieser so genannten Online-Probleme werden Online-Algorithmen eingesetzt. Im Gegensatz dazu stehen bei Offline-Problemen alle Informationen zum Berechnungszeitpunkt zur Verfügung. Um die Güte von Online-Algorithmen abzuschätzen, wird deren Lösung mit der optimalen (Offline-) Lösung des gegebenen Problems verglichen. Für diesen Vergleich wird der Begriff der Kompetitivität verwendet, der wie folgt definiert ist:

Definition 1.7. *Sei Π ein Problem und S eine Strategie, die jede Instanz $P \in \Pi$ korrekt löst. Seien $S(P)$ die Kosten, die S verursacht, und $\text{OPT}(P)$ die Kosten einer optimalen Lösung von P . Dann heißt S **kompetitiv** mit Faktor c (oder kurz c -kompetitiv), falls es ein $\alpha > 0$ gibt, so dass für alle $P \in \Pi$ gilt:*

$$S(P) \leq c \cdot \text{OPT}(P) + \alpha.$$

Die additive Konstante α wird oft für initiale Kosten benötigt.

Für nachfolgende Betrachtungen sei $G = (V, E)$ ein ungerichteter Graph mit $|V| = n$, $|E| = m$ und Gewichtsfunktion $w : E \rightarrow [0, \infty)$. Je nach Problemstellung ordnet die Gewichtsfunktion w jeder Kante ihre Länge bzw. die zur Traversierung benötigte Zeit zu. Darüber hinaus seien Start- und Zielknoten durch $s, t \in V$ gegeben, wobei $s \neq t$ gelte.

1.3.1 Recoverable-Varianten des CTP

Der Unterschied der Recoverable-Varianten des CTP zum ursprünglichen CTP liegt darin, dass einmal blockierte Kanten wieder freigegeben werden können. Neben der bereits

bekannten Gewichtsfunktion w existiert zusätzlich eine Recoverfunktion

$$r : E \rightarrow [0, \infty).$$

Diese ordnet jeder Kante $e \in E$ eine Wiederherstellungszeit $r(e)$ zu, die angibt, wie lange eine Kante blockiert ist. Diese Information beeinflusst natürlich auch die Gesamtkosten des von einer Strategie berechneten Pfades von s nach t . Auch für alle Knoten $v \in V$ wird eine Wiederherstellungszeit $r_{\text{vertex}}(v)$ festgelegt. Für diese gilt:

$$r_{\text{vertex}}(v) \geq \max\{r(e) \mid \text{für alle } e = \{v, v'\} \in E\}.$$

Die Wiederherstellungszeit $r_{\text{vertex}}(v)$ gibt also die maximale Ausfallzeit der zu v inzidenten Kanten an. Ein Knoten selbst fällt jedoch zu keinem Zeitpunkt aus. Bei Erreichen eines Knotens v bestehen nun die Möglichkeiten, eine freie zu v inzidente Kante zu passieren oder $r_{\text{vertex}}(v)$ abzuwarten, bis alle zu v inzidenten Kanten wieder verfügbar sind (sofern diese nicht unmittelbar nach ihrer Recoverzeit wieder blockiert werden). Das CTP stellt im Prinzip nur einen Spezialfall des Recoverable-CTP dar, in dem alle Wiederherstellungszeiten unendlich groß sind. Die von Bar-Noy und Schieber vorgestellten Algorithmen setzen dagegen Wiederherstellungszeiten voraus, für die gilt:

$$\forall e, e' \in E : r(e) \leq w(e').$$

Diese Annahme garantiert, dass nach Passieren einer Kante alle bereits entdeckten Blockaden wiederhergestellt sind. Es sei darauf hingewiesen, dass dies eine starke Einschränkung darstellt und die Problemstellung somit eher theoretischer Natur ist. Dennoch lohnt die Betrachtung der von Bar-Noy und Schieber vorgeschlagenen Algorithmen, da diese nicht nur Lösungen der Recoverable-Varianten des CTP darstellen, sondern auch als Basis für weitere Problemlösungen dienen. Es werden zwei Varianten des Recoverable-CTP unterschieden:

- Stochastisches Recoverable Canadian Traveller Problem (sr-CTP)
- Deterministisches Recoverable k -Canadian Traveller Problem (dr- k -CTP)

Stochastisches Recoverable CTP (sr-CTP)

Veranschaulichend lässt sich das stochastische Recoverable Canadian Traveller Problem (sr-CTP) als Modell für eine Straßenkarte mit Stauprognosen für einzelne Straßen betrachten. Dazu wird jeder Kante $e \in E$ durch eine Blockadefunktion

$$p : E \rightarrow [0, 1]$$

eine Wahrscheinlichkeit $p(e)$ zugeordnet, mit der sie blockiert ist. Da der Algorithmus von Bar-Noy und Schieber die Blockadewahrscheinlichkeit bei der Berechnung der Lösung mit

einbezieht, liefert er als Ergebnis einen Lösungsweg mit erwarteter Reisezeit $E(s, t)$ vom Startknoten s zum Zielknoten t . Dieser Ansatz berücksichtigt also den Kompromiss zwischen kürzestem Weg und geringster Blockadewahrscheinlichkeit. Der Algorithmus basiert auf Dijkstras Kürzeste-Wege-Algorithmus und berechnet zu jedem Knoten $v \in V$ eine Prioritätsliste $P(v)$. Diese Listen werden durch den Algorithmus sukzessive aufgebaut und enthalten in aufsteigend sortierter Reihenfolge Kosteninformationen der zu v inzidenten Kanten. In diese Informationen fließen neben den Traversierungskosten auch die Blockadewahrscheinlichkeiten ein. Über die Prioritätslisten wird jedem Knoten v eine erwartete Reisezeit $E(v, t)$ zum Zielknoten t zugeordnet. Zu Beginn gilt für die Knoten:

- $\forall v \in V, v \neq t : E(v, t) = \infty$
- $E(t, t) = 0$

Außer $E(t, t) = 0$ sind alle Erwartungswerte nur vorläufige Bewertungen, die schrittweise aktualisiert werden, bis $E(s, t)$ fest steht. Die Ausgabe des Algorithmus ist eine Reisestrategie von s nach t mit erwarteter Reisezeit $E(s, t)$. Diese besteht aus allen Prioritätslisten und kann wie folgt auf eine gegebene Problem Instanz mit tatsächlichen Kantenblockaden angewandt werden: Der Reisende startet in s und traversiert an jedem Knoten v seiner Route stets die erste Kante der Prioritätsliste $P(v)$. Sollte diese blockiert sein, prüft der Reisende die nächste Kante der Liste. Sollten an einem Knoten v alle Kanten aus $P(v)$ blockiert sein, so wartet der Reisende $r_{vertex}(v)$ und beginnt anschließend wieder mit der ersten Kante aus $P(v)$. Diese Strategie setzt er solange fort, bis er t erreicht. Die Laufzeit des Algorithmus liegt in $O(m \log n)$. Bar-Noy und Schieber zeigen in ihrer Arbeit, dass die erwartete Reisezeit von einem Knoten v nach t tatsächlich dem Wert $E(v, t)$ entspricht, den der Algorithmus berechnet. Neben der Korrektheit wird auch die Optimalität der Werte $E(v, t)$ gezeigt, woraus folgt, dass der Algorithmus die minimale erwartete Reisezeit korrekt ermittelt. Wird die berechnete Reisestrategie jedoch auf eine konkrete Instanz des sr-CTP angewandt, so ist der Algorithmus dennoch nicht kompetitiv. Die Wahrscheinlichkeit einer für den Algorithmus sehr ungünstigen Eingabe ist stochastisch gesehen zwar gering, muss zur Bestimmung des kompetitiven Faktors jedoch berücksichtigt werden. Da eine Problem Instanz die vom Algorithmus berechnete Strategie im schlimmsten Fall zu beliebig großen relativen Kosten veranlassen kann, lässt sich kein kompetitiver Faktor angeben.

Deterministisches Recoverable k -CTP (dr- k -CTP)

Das deterministische Recoverable k -Canadian Traveller Problem (dr- k -CTP) lässt sich durch die Vorstellung einer gewissen Anzahl von Baustellen innerhalb eines Gebietes veranschaulichen. Dieses Gebiet wird dabei durch den Graphen G repräsentiert. Wir nehmen an, dass höchstens k verschiedene Straßen blockiert sein können, wobei die Zeit der Blockade wieder durch den Wert $r(e)$ der Recoverfunktion für die betreffende Kante $e \in E$ bestimmt ist. Über den Ort der Blockaden in G ist auch weiterhin a priori nichts bekannt. Der von

Bar-Noy und Schieber vorgestellte Algorithmus zu diesem Problem berechnet einen Weg von s nach t in G unter Berücksichtigung k möglicher Blockaden mit optimaler worst-case-Reisezeit. Auch dieser Algorithmus basiert auf einer Weiterentwicklung von Dijkstras Kürzeste-Wege-Algorithmus. Wie beim sr-CTP liegt auch hier die Lösung des Problems nicht in der Ausgabe eines eindeutigen Pfades von s nach t , sondern in der Ausgabe von $k+1$ Reisestrategien $S^0, \dots, S^k$. Jede einzelne Reisestrategie S^h , mit $0 \leq h \leq k$, berücksichtigt hierbei die Möglichkeit, dass noch genau h Kanten in G blockiert sein können. Eine Instanz des dr- k -CTP kann dann unter Verwendung dieser Strategien gelöst werden. Erreicht ein Reisender einen Knoten $v \in V$ oder startet von dort, so ist wie folgt die geeignete Strategie zu wählen:

- Hat der Reisende *vor* Ankunft am Knoten v bereits $b \in \{0, \dots, k\}$ blockierte Kanten entdeckt, wählt er Strategie S^h , mit $h = k - b$.

Nachdem die Auswahl von S^h getroffen wurde, wird mittels dieser Strategie die nächste zu traversierende Kante berechnet. Jede einzelne Strategie S^h hat folgende wesentlichen Bestandteile:

- i Listen $\text{Alt}_i^h(v)$ alternativer Kanten für jeden Knoten $v \in V$, $0 \leq i \leq h$
- eine Primärkante $\text{Prim}^h(v)$ für jeden Knoten $v \in V$

$\text{Alt}_i^h(v)$ bezeichnet hierbei die alternative Liste des Knotens v für die Strategie S^h , wobei an v aktuell i lokale Blockaden festgestellt werden. Sie enthält geeignete zu v inzidente alternative Kanten in der Reihenfolge ihrer Priorität. Besteht eine Liste $\text{Alt}_i^h(v)$ aus weniger als $i + 1$ Kanten, wird diese mit Kanten $\{v, v\}$ bis zur Länge $i + 1$ ergänzt, wodurch eine ausreichende Anzahl an Alternativen an jedem Knoten v sichergestellt ist. Die Listen werden benötigt, um an einem bestimmten Knoten die nächste zu traversierende Kante zu ermitteln. Für den Fall, dass am Knoten v die Anzahl der lokalen Blockaden $i = 0$ ist, besteht die Liste alternativer Kanten $\text{Alt}_i^h(v)$ nur aus der sogenannten Primärkante $\text{Prim}^h(v) = \text{Alt}_0^h(v) = \{v, u\}$. Diese Kante wird über eine Kostenfunktion so gewählt, dass die Reisezeit für den restlichen Pfad von u nach t im Fall der ungünstigsten Blockaden, die noch auftreten können, minimiert wird. Die an einem Knoten v in Abhängigkeit von der Anzahl bereits entdeckter Blockaden gewählte Strategie S^h geht wie folgt vor:

- Sind an v keine Kanten blockiert, wählt der Reisende die Primärkante $\text{Prim}^h(v)$ von S^h .
- Sind an v i Kanten blockiert, wählt der Reisende die erste nicht blockierte Kante in der i -ten Liste alternativer Kanten $\text{Alt}_i^h(v)$ von S^h . Handelt es sich hierbei um die Kante $\{v, v\}$, entspricht das dem Abwarten an Knoten v .

Die Laufzeit des Algorithmus liegt in $O(k^2m + kn \log n)$. Bar-Noy und Schieber zeigen, dass der Algorithmus $k + 1$ Strategien zur Berechnung eines Pfades von s nach t in G unter Berücksichtigung k möglicher Blockaden mit optimaler worst-case-Reisezeit korrekt ermittelt. Das bedeutet nicht, dass der berechnete Weg auch gleichzeitig der kürzeste Weg

zwischen diesen beiden Knoten ist. Die Reisezeit ist im worst-case optimal unter der Annahme, dass der Reisende die Blockaden im Vorfeld nicht kennt und „online“ darauf reagieren muss. Der Algorithmus löst also das *Online-Problem* dr- k -CTP worst-case-optimal.

1.3.2 Das k -Canadian Traveller Problem (k -CTP)

Geben wir für das Recoverable k -Canadian Traveller Problem (dr- k -CTP) die Eigenschaft der wiederherstellbaren Kanten auf, erhalten wir das k -Canadian Traveller Problem (k -CTP), vergleiche Definition 1.4. Der Unterschied zum ursprünglichen Canadian Traveller Problem (siehe Definition 1.2) besteht also lediglich darin, dass a priori eine obere Schranke k auftretender Blockaden bekannt ist. In ihrer Arbeit stellen Bar-Noy und Schieber zunächst einen effizienten Algorithmus für $k = 1$ vor, der sich auch auf beliebiges k verallgemeinern lässt. Dieser berechnet mit einer leicht modifizierten Version von Dijkstras Kürzeste-Wege-Algorithmus zwei Strategien S^0 und S^1 . Über diese Strategien lässt sich, analog zum dr- k -CTP, ein Weg von s nach t in G bestimmen, der unter Berücksichtigung der möglichen Blockade eine optimale worst-case-Reisezeit garantiert. Auch hier bedeutet das nicht, dass der ermittelte Weg gleichzeitig dem kürzesten Weg zwischen s und t entspricht, sondern dass die Reisezeit worst-case-optimal ist unter der Annahme, dass der Reisende die Blockade im Vorfeld nicht kennt und „online“ darauf reagieren muss. Der Algorithmus berechnet also einen s - t -Pfad, dessen maximale Länge im schlimmsten Fall am kürzesten ist. Bei der Auswahl der geeigneten Strategie wird ähnlich wie beim dr- k -CTP zwischen folgenden Fällen unterschieden:

- Hat der Reisende noch keine blockierte Kante entdeckt, wählt er Strategie S^1 .
- Hat der Reisende *vor* Erreichen des aktuellen Knotens bereits eine blockierte Kante entdeckt, wählt er Strategie S^0 .

Die Strategie S^0 besteht für alle $v \in V$ lediglich aus einer Primärkante $\text{Prim}(v)$. Die Strategie S^1 besteht zusätzlich aus je einer alternativen Kante $\text{Alt}(v)$, für jedes $v \in V$. Die an einem Knoten v gewählte Strategie wird von einem Reisenden wie folgt benutzt:

- Ist *an* v keine Kante blockiert, wählt der Reisende die Primärkante $\text{Prim}(v)$ der aktuellen Strategie.
- Ist *an* v die Primärkante blockiert, wählt der Reisende die alternative Kante $\text{Alt}(v)$ der aktuellen Strategie.

Die Primärkanten der Knoten sind gerade die Kanten, die die Länge des kürzesten Pfades von s nach t im ungünstigsten Fall minimieren. Dabei wird der ungünstigste Fall einer blockierten Kante unter allen $m = |E|$ Möglichkeiten ermittelt, eine Kante aus G zu entfernen. Es sei

$$\pi(s, t) := (s, v_1, v_2, \dots, t)$$

der Weg von s nach t bestehend aus Primärkanten. Dann ist die maximale Länge dieses Pfades in allen Graphen $G_i := (V, E \setminus \{e_i\})$ die kürzeste. Hierbei bezeichnet $e_i \in E$ die

beliebige Ausfallkante. Da in S^0 keine weiteren Blockaden auftreten können, entspricht dort die Länge von s nach t im ungünstigsten Fall gerade der Länge des kürzesten Pfades zwischen diesen beiden Knoten. Die alternative Kante $\text{Alt}(v)$ am Knoten v ist diejenige auf dem kürzesten Pfad von v nach t *ohne* die Primärkante von v , denn es gilt:

- Am Knoten v wird nur dann eine alternative Kante gewählt, wenn dessen Primärkante blockiert ist.
- Ist die Primärkante von v blockiert, so ist bereits eine Kante in G gesperrt, und es kann wegen $k = 1$ keine weiteren Blockaden mehr geben.

Insgesamt liegt die Laufzeit dieses Algorithmus in $O(m + n \log n)$. Um den Algorithmus auf beliebige (aber feste) k zu erweitern, würde im Prinzip eine k -malige Wiederholung ausreichen. Aus diesem Grund ist das k -CTP mit einer Erweiterung des obigen Algorithmus in polynomieller Zeit lösbar. Weil jedoch die Berechnungskomplexität der Strategien $S^1, \dots, S^k$ exponentiell in der Größe der Eingabe wächst, ist eine Erweiterung des Algorithmus auf beliebige *unbeschränkte* k nicht effizient. In diesem Fall betrachten wir das CTP, dass, wie wir in Kapitel 2 genauer sehen werden, nicht effizient lösbar ist.

1.3.3 Das k -Vital Edges Problem (k -VEP)

Das k -Vital Edges Problem (k -VEP) kann als zum k -CTP duales Problem angesehen werden. Gegeben sind ein gewichteter Graph $G = (V, E)$, zwei Knoten $s, t \in V$ sowie ein Parameter k . Gesucht sind k Kanten, deren Entfernen aus dem Graphen G den kürzesten Pfad zwischen s und t maximiert. Bei dem zum k -VEP korrespondierenden Entscheidungsproblem ist zusätzlich ein Parameter b gegeben. Die Frage lautet hier: Können k Kanten gefunden werden, deren Entfernen aus dem Graphen G zu einer Verlängerung des kürzesten Pfades zwischen s und t um mehr als b führt? Bar-Noy und Schieber geben an, dass das k -VEP NP -hart ist und zeigen die NP -Vollständigkeit für das korrespondierende Entscheidungsproblem.

1.4 Grundlagen

In diesem Abschnitt behandeln wir Bezeichnungen und Definitionen, die für das Verständnis dieser Arbeit notwendig sind. Wir haben uns bemüht, Definitionen so intuitiv wie möglich zu formulieren, und auf die in der Literatur am häufigsten verwendeten Bezeichnungen und Symbole zurückgegriffen.

Definition 1.8. Ein **Pfad** $\pi(v_0, v_m)$ ist eine Knotenfolge $(v_0, v_1, \dots, v_m)$, mit $v_i \in V$, für alle $i \in \{0, \dots, m\}$, wobei $\{v_j, v_{j+1}\} \in E$, für alle $j \in \{0, \dots, m-1\}$. Ein solcher Pfad heißt Pfad von u nach v , falls $u = v_0$ und $v = v_m$ gilt. Die Anzahl seiner Kanten ist m .

Definition 1.9. Ein solcher Pfad heißt **einfacher Pfad**, wenn seine Knoten paarweise verschieden sind.

Definition 1.10. Ein Pfad $(v_0, v_1, \dots, v_m)$ heißt **Kreis** oder **Zyklus**, falls $v_0 = v_m$ gilt.

Definition 1.11. Ein Pfad $(v_0, v_1, \dots, v_m)$ heißt **einfacher Kreis**, falls $v_0 = v_m$ gilt und die Knoten paarweise verschieden sind.

Definition 1.12. Gegeben sei ein ungerichteter Graph $G = (V, E)$. Eine **Gewichtsfunktion** $w : E \rightarrow [0, \infty)$ belegt jede Kante $e \in E$ mit nicht-negativen Kosten (z. B. Länge oder Reisedauer).

Definition 1.13. Gegeben sei ein ungerichteter Graph $G = (V, E)$ mit Gewichtsfunktion $w : E \rightarrow [0, \infty)$. Die Länge eines Pfades $\pi(v_0, v_k) = (v_0, v_1, \dots, v_k)$ ist gegeben durch

$$|\pi(v_0, v_k)| := \sum_{i=0}^{k-1} w(\{v_i, v_{i+1}\}).$$

Die **Länge des kürzesten Pfades** $d_G(u, v)$ von $u \in V$ nach $v \in V$ in G ist gegeben durch das Minimum der Längen aller Pfade π von u nach v in G . Es gilt:

$$d_G(u, v) := \min\{|\pi(u, v)| \mid \pi(u, v) \text{ ist ein Pfad von } u \text{ nach } v \text{ in } G\}.$$

Das Minimum der leeren Menge definieren wir dabei der Konvention folgend als ∞ .

Definition 1.14. Gegeben sei ein ungerichteter Graph $G = (V, E)$. Ein **kürzester Pfad** von $u \in V$ nach $v \in V$ in G wird mit $\pi_G^{\min}(u, v)$ bezeichnet. Es gilt $|\pi_G^{\min}(u, v)| = d_G(u, v)$, für alle $u, v \in V$.

Im Gegensatz zu der auf Seite 10 definierten *nicht eindeutigen* Strategie ist unter einer *eindeutigen* Strategie Folgendes zu verstehen:

Definition 1.15. Eine Strategie S zu einem Problem Π heißt **eindeutige Strategie**, wenn zu jeder Instanz $P \in \Pi$ genau eine Graph-Strategie $S_G \in S$ existiert, die P korrekt löst.

Auf die an dieser Stelle eingeführten Definitionen werden wir an verschiedenen Stellen der Diplomarbeit zurückgreifen. Alle Definitionen, die im weiteren Verlauf der Arbeit eingeführt werden, sind nur für die betreffenden Kapitel relevant.

Kapitel 2

Komplexitätsanalyse des CTP

In diesem Kapitel stellen wir den Komplexitätsbeweis zum Canadian Traveller Problem aus [PY89] in einer aufbereiteten und ausführlicheren Version vor. Bei diesem Problem haben wir im Gegensatz zum k -CTP keine Kenntnis über die Anzahl ausfallender Kanten. Dabei betrachten wir den Quotienten aus den Kosten einer deterministischen Online-Strategie, die das CTP löst, zu denen einer optimalen Offline-Lösung. Unter einer optimalen Offline-Lösung verstehen wir dabei die Berechnung eines kürzesten Pfades vom Start zum Zielknoten, wobei die Kantenausfallmenge a priori vollständig bekannt ist und zur Berechnung verwendet werden darf. Papadimitriou und Yannakakis haben in ihrer Arbeit [PY89] Folgendes bewiesen: Das Finden eines kleinsten Quotienten $r \geq 1$, für den eine Online-Strategie existiert, die das CTP löst, ist *PSPACE*-hart.

Um dies zu zeigen nutzt man Folgendes aus: Eine Probleminstanz $\mathcal{P}$ sei durch einen Graphen $G = (V, E)$ und einen festen Quotienten $r \geq 1$ gegeben. Die Entscheidung, ob für $\mathcal{P}$ eine Graph-Strategie existiert, die die r -fachen Wegkosten gegenüber einer optimalen Offline-Lösung nicht überschreitet, ist *PSPACE*-vollständig. Diese Fragestellung ist das CTP-Entscheidungsproblem. Aus der *PSPACE*-Vollständigkeit des CTP-Entscheidungsproblems folgt die *PSPACE*-Härte für das Finden eines minimalen Quotienten $r \geq 1$ zu einer gegebenen Probleminstanz $\mathcal{P}$.

Definition 2.1. *Gegeben sei eine wohldefinierte **quantifizierte Boolesche Formel (QBF)** $F = (Q_1x_1)(Q_2x_2)\dots(Q_nx_n) : \varphi$, wobei φ eine aussagenlogische Boolesche Formel unter Verwendung der Variablen $x_1, x_2, \dots, x_n$ und jedes Q_i einer der Quantoren „ $\exists$ “ oder „ $\forall$ “ ist und dabei eine Variable x_i bindet. Gefragt ist nun, ob φ erfüllt ist.*

Direkt aus dem Problem QBF ableitbar ist das zugehörige Erfüllbarkeitsproblem.

Definition 2.2. *Gegeben sei eine quantifizierte Boolesche Formel F . Das **quantifizierte Erfüllbarkeitsproblem (QSAT)** fragt, ob es eine Belegung gibt, so dass F erfüllt wird.*

Diese mit Quantoren versehene Boolesche Formel F kann im Falle eines alternierenden Vorkommen der Quantoren $\exists$ und $\forall$ als Spiel zwischen zwei Personen angesehen werden.

Definition 2.3. Es sei F eine quantifizierte Boolesche Formel mit alternierenden Quantoren Q_i , wobei $Q_{2i-1} := \{\exists\}$ und $Q_{2i} := \{\forall\}$, für alle $i \in \{1, \dots, \lfloor \frac{n}{2} \rfloor\}$. Unter einem **QSAT-Spiel** verstehen wir ein abwechselndes Belegen der Variablen x_i . Der Spieler versucht die Formel dabei die Formel φ zu erfüllen, indem er die Variablen x_i mit ungeradem Index i belegt. Gehindert wird er daran von einem Gegenspieler, der versucht φ unerfüllbar zu machen und dazu die Variablen x_i mit geradem Index i belegt.

Dadurch entsprechen die Existenzquantoren jeweils den Entscheidungen eines Spielers, während die Allquantoren jeweils die Entscheidungen eines Gegenspielers repräsentieren. In die Sprache der Online-Algorithmen übertragen, hieße demnach obige Fragestellung, ob Spieler 1 eine Graph-Strategie kennt, mit der er immer (also gegen jeden Gegenspieler) gewinnt.

Viele, auf deren Komplexität untersuchte Spiele, bei denen zwei Spieler abwechselnd Entscheidungen treffen müssen, wie Go, Schach (auf ein beliebig großes Feld der Größe $n \times n$ verallgemeinert) oder GEOGRAPHIE (siehe [TS78]) sind *PSPACE*-vollständig.

Definition 2.4. Das **GEOGRAPHIE-Spiel** ist dabei wie folgt definiert: Gegeben ist ein gerichteter Graph $G = (V, E)$ sowie ein Startknoten $s \in V$. Es spielen zwei Spieler gegeneinander. Die Teilnehmer markieren nun nach folgenden Regeln abwechselnd Knoten in G :

1. Im ersten Zug markiert Spieler 1 s .
2. Im t -ten Zug, mit $t > 0$, markiert der jeweilige Spieler der am Zug ist einen direkten Nachfolgerknoten des Knotens, welcher im $t - 1$ -ten Zug markiert wurde.
3. Ein Knoten darf nicht mehrmals markiert werden.
4. Es gewinnt der Spieler, welcher den letzten Zug ausführt.

QSAT ist das kanonische Problem der Komplexitätsklasse *PSPACE*-vollständigen Probleme. Es stellt gewissermaßen das Pendant zu SAT für die Klasse der *NP*-vollständigen dar. Gelingt die Reduktion eines Problems $\mathcal{P}$ auf das als *PSPACE*-vollständige bekannte Problem QSAT, so ist gezeigt, dass $\mathcal{P}$ *PSPACE*-hart sein muss. Für die *PSPACE*-Vollständigkeit ist dann nur noch $\mathcal{P} \in PSPACE$ zu zeigen. Für nähere Hintergrundinformationen verweisen wir den Leser auf die einschlägige Literatur, insbesondere auf [SM73] und [CP83].

Die Reduktion des Canadian Traveller Problem auf das $PSPACE$ -vollständige Problem QSAT gelingt dabei über einen kleinen Umweg. Wir definieren hierfür zunächst, was wir unter dem Problem Double Valued Graph verstehen:

Definition 2.5. Eine Instanz des Problems **Double Valued Graph (DVG)** ist gegeben durch einen gerichteten Graphen $G = (V, E)$ sowie zwei Gewichtsfunktionen $w_1 : E \rightarrow \mathbb{R}_{\geq 0}$ und $w_2 : E \rightarrow \mathbb{R}_{\geq 0}$ gegeben, wobei das tatsächliche Gewicht einer Kante $e = (u, v)$ (also $w_1(e)$ oder $w_2(e)$) erst bekannt wird, wenn u oder v betreten werden. Darüberhinaus sind ein Startknoten $s \in V$ und Zielknoten $t \in V$ sowie ein fester Faktor $r \geq 1$ gegeben. Gesucht ist ein $s - t$ -Weg $\pi(s, t)$, so dass die Reisekosten im ungünstigsten Fall höchstens um den Faktor r schlechter sind, als eine optimale Lösung.

Außerdem können wir folgendes Entscheidungsproblem formulieren:

Definition 2.6. Gegeben sei eine Problem Instanz $\mathcal{P}$ vom DVG. Das **DVG-Entscheidungsproblem** fragt, ob für ein festes $r \geq 1$ eine Lösung für $\mathcal{P}$ existiert.

Da die Beweisführung sehr komplex ausfällt möchten wir an dieser Stelle einen kurzen Überblick über unser Vorhaben geben: Wir zeigen in den nachfolgenden Abschnitten, dass das Entscheidungsproblem Double Valued Graph (DVG) auf QSAT reduziert werden kann. Die Zugehörigkeit zur Komplexitätsklasse $PSPACE$ ist leicht einsehbar, da sich alle $2^{|E|}$ Kombinationen ausprobieren ließen. Damit muss unter diesen, falls überhaupt eine Lösung für das DVG-Entscheidungsproblem für diesen konkreten Faktor r existiert, auch der Teilgraph sein, für den die letztlich richtige Gewichtsfunktion w_1 oder w_2 gilt. Für jeden dieser Graphen können wir mit polynomiell Platzbedarf die minimalen Kosten eines Pfades $\pi(s, t)$ von s nach t berechnen. Die Speicherung des Ergebnisses, sowie der aktuelle Fortschritt, ließen sich ebenfalls mit polynomiell Platzaufwand auf einer Turingmaschine realisieren. Damit ist insgesamt gezeigt, dass das DVG-Entscheidungsproblem $PSPACE$ -vollständig ist.

Eine Instanz des Problems DVG kann außerdem durch eine weiter unten vorgestellte Konstruktion in eine Instanz des CTP transformiert werden. Dadurch ist der Beweis der $PSPACE$ -Vollständigkeit für das CTP gelungen. Wie diese Beweisführung im Detail erfolgt, erläutern die nächsten Abschnitte.

2.1 Reduktion von DVG auf QSAT

Im Folgenden stellen wir den aufbereiteten Beweis von Papadimitriou und Yannakakis vor.

Beweis. Für die Reduktion vom DVG-Entscheidungsproblem auf QSAT betrachten wir eine Instanz von QSAT der Form

$$Q = (Q_1x_1)(Q_2x_2)\dots(Q_nx_n) : F, \text{ wobei } Q_i \in \{\forall, \exists\},$$

die in 3KNF vorliegt. Es sei nun n die Anzahl der Variablen, p die Anzahl der Klauseln in F und $m := 7(n+p)$. Wir konstruieren einen gerichteten, gewichteten Graphen $G = (V, E)$, der über weniger als m und daher polynomiell viele Knoten verfügt und bei dem die Kanten jeweils mit bis zu zwei von vier möglichen Längen gewichtet sind: $1 \ll K \ll L \ll M$. Das Beispiel aus Abbildung 2.1 stellt die Konstruktion des Graphen für eine konkrete Formel dar. Die Gewichtung einer gerichteten Kante $e = (u, v) \in E$ erfolgt gemäß des DVG mit einer Menge von zwei möglichen Werten, deren relevanter Wert erst bekannt wird, wenn u erreicht wird. Genauer wählen wir:

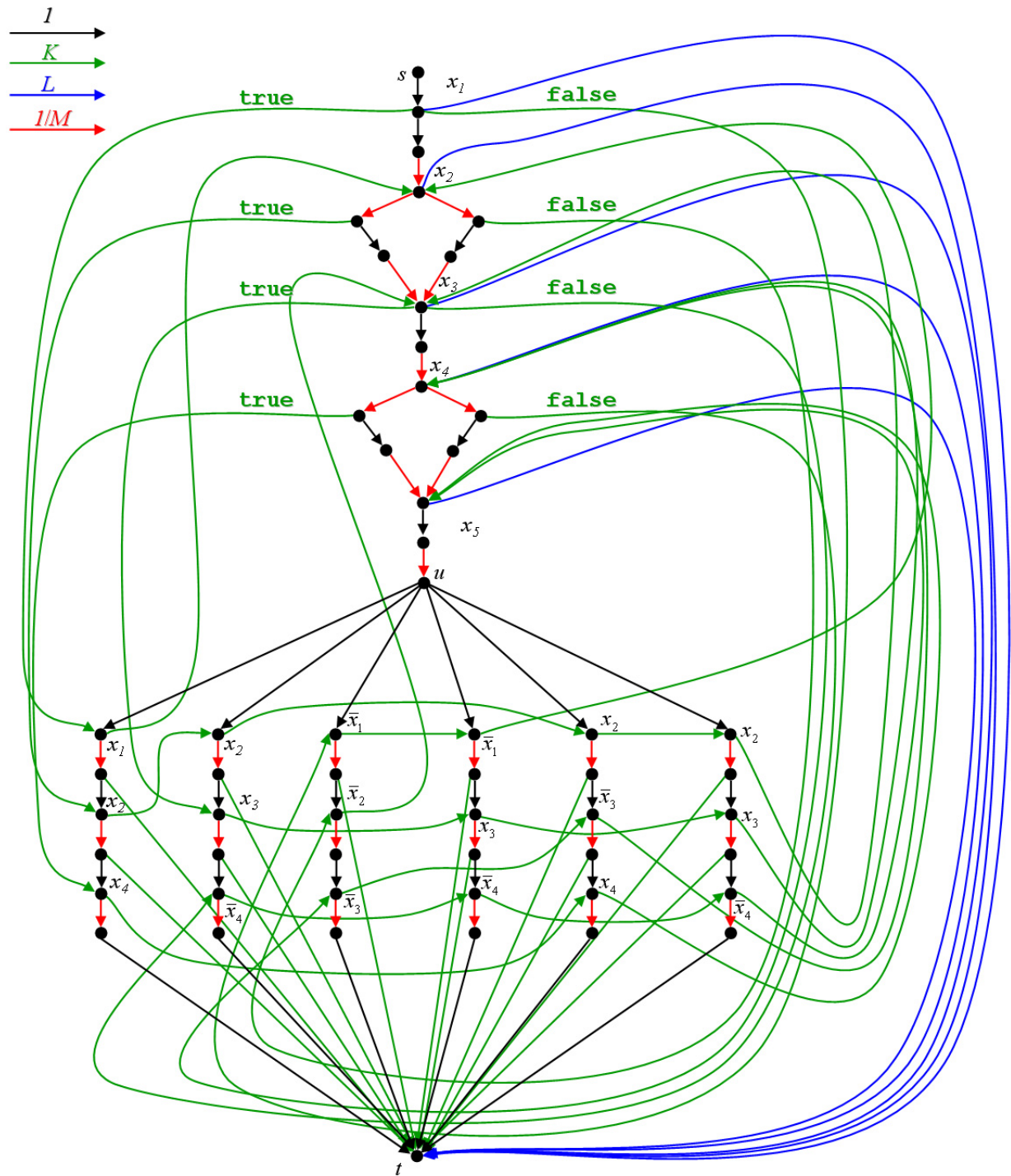
$$\begin{aligned} K &:= m^3 \\ L &:= m^6 \\ M &:= m^{12} \end{aligned}$$

Die beiden möglichen Längen einer Kante im DVG separieren wir durch „|“ zwischen den beiden möglichen Traversierungskosten. Kanten, die mit nur einem Wert markiert sind, haben auch nur diese einen möglichen Reisekosten. Sollte eine Markierung nicht vorhanden sein, gelten für die Länge dieser Kante Kosten von 1. Im Beispiel haben wir der Übersichtlichkeit wegen farbliche Markierungen gewählt, deren Bedeutungen der Legende zu entnehmen sind. Den relativen Faktor bezüglich eines optimalen Pfades, der erreicht werden soll und im Voraus bekannt ist, wählen wir

$$r := m^4.$$

Es ist nun unser Ziel einen Graphen zu konstruieren, der als Grundlage für ein Spiel zwischen zwei Personen dient. Bei diesem versucht einer der beiden Akteure einen kürzesten Pfad zwischen zwei Knoten bei unvollständiger Karte zu finden, während der andere die Konditionen für diese Suche möglichst schlecht wählt. Für den Suchenden, im Folgenden auch Spieler genannt, soll genau dann eine Graph-Strategie existieren, die einen Faktor r gewährleistet, wenn Q lösbar ist.

Der obere Teil im Beispiel aus Abbildung 2.1 stellt die Belegung der Variablen dar, während der untere Teil die Klauseln der Formel F in 3KNF repräsentiert. Betrachten wir jedoch zunächst den unteren Teil. Für jede Klausel gibt es sieben aufeinander folgende gerichtete Kanten, die - von $u \in V$ ausgehend - abwechselnd mit 1 und $1/M$ gelabelt werden und schließlich in $t \in V$ münden. Für jede der drei möglichen Literale einer der p Klauseln existiert, falls notwendig, am betreffenden Knoten eine ausgehende Kante mit Kosten K , die zu weiteren Vorkommen dieser Variablen in anderen Klauseln führt. Dies betrifft jedoch nicht die entsprechende Negation dieses Literals. Betrachten wir hierzu die im oberen Teil des Graphen aus Abbildung 2.1 am x_2 -Komplex beginnende Kante mit der Beschriftung **true** und Gewichtung K . Diese Kante führt zum ersten Klauselstrang ganz links im unteren Teil des Graphen. Einen Knoten, in den eine solche mit K gewichtete Kante führt, bezeichnen wir im Folgenden als Literalknoten. Er repräsentiert ein Literal innerhalb einer der p Klauseln in F . Von hier führt eine Kante mit Gewichtung K zum



$$\exists x_1 \forall x_2 \exists x_3 \forall x_4 : (x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee x_3 \vee \bar{x}_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_3 \vee \bar{x}_4) \wedge (x_2 \vee \bar{x}_3 \vee x_4) \wedge (x_2 \vee x_3 \vee \bar{x}_4)$$

Abbildung 2.1: Beispiel für die Konstruktion des Graphen (DVG)

Vorkommen dieses Literals im Klauselstrang direkt rechts daneben. Dies repräsentiert das nächste Vorkommen des Literals x_2 in der zweiten Klausel von F . Auf diese Weise werden analog auch die letzten beiden Literalvorkommen ganz rechts im unteren Teil des Graphen mit erneuten Kosten K verbunden. Es ist also nicht möglich ein Vorkommen von $\bar{x}_2$ zu erreichen. Von hier geht es über eine mit K gewichtete Kante zurück in den oberen Teil des Graphen zum Knoten mit der Beschriftung x_3 .

Ein Literalstrang verbindet dabei die drei Literale einer Klausel miteinander. Von allen drei Literalen geht dabei ein zweielementiger Pfad mit Gewichtung $1|M$ für die erste Kante und Kosten 1 für die darauffolgende Kante aus. Von den ersten beiden Literalen führen weitere Kanten zum nachfolgenden Literal derselben Klausel, während der Pfad vom letzten Literal am Zielknoten t ankommt. Die beiden Knoten zwischen den drei Literalknoten verfügen desweiteren über eine Kante mit Gewichtung K , die direkt zum Zielknoten t führt. Die Konstruktion des unteren Teils ist damit abgeschlossen und ermöglicht die Repräsentation der zu erfüllenden Formel F .

Der obere Teil des Graphen muss nun Möglichkeiten bereithalten, die Variablenbelegung gemäß der Wahl der Quantoren Q_i zu repräsentieren. Je nachdem, ob eine Existenz- oder Allquantifizierung darzustellen ist, muss genau definiert sein, wie der Graph zu konstruieren ist. Um beide Fälle zu betrachten, orientieren wir uns erneut an dem Beispiel aus Abbildung 2.1. Ganz oben sehen wir den Startknoten $s \in V$. Dieser verfügt über genau eine ausgehende Kante (s, v_1) , die mit Kosten $|(s, v_1)| := 1$ gewichtet wird und in den Repräsentantenknoten der ersten Variable x_1 führt. Dies sei, gemäß unseres Beispiels, eine existenzquantifizierte Variable.

Von Knoten dieser Art gehen stets vier Kanten aus. Eine mit Kosten L , die direkt zum Zielknoten t führt und zwei weitere Kanten mit Kosten K , die die beiden booleschen Belegungen der Variablen repräsentieren. Das Verfolgen eines dieser beiden Pfade bewirkt ein Durchlaufen des unteren Teiles des Graphen, wobei jeder Klauselstrang besucht wird, der über ein Vorkommen dieser Variablen mit entsprechendem Wahrheitswert verfügt. Außerdem führt einen eine Kante mit Kosten K zur nächsten Variablen. Im Falle unsere Beispiels ist dies die allquantifizierte Variable x_2 . Die vierte Kante, die vom Repräsentantenknoten für x_1 ausgeht, verlässt diesen mit Kosten 1 und führt über eine Kante mit Kosten $1|M$ direkt in den nächsten Knoten.

Im Fall einer universellen Variablen verwenden wir nur drei ausgehende Kanten. Es existiert erneut eine mit L gewichtete Kante zum Zielknoten t , sowie zwei ausgehende Kanten mit Kosten $1|M$, die zu unterschiedlichen Knoten führen. An diesen Knoten geht jeweils eine mit K gewichtete Kante zu eventuellen Vorkommen dieses Literals in Klauselstränge des unteren Graphenabschnittes ab. Dabei repräsentieren diese beiden Kanten entsprechend die Wahrheitswerte **true** und **false**. Darüber hinaus existieren erneut Kantenfolgen, die - wie

in Abbildung 2.1 ersichtlich - achsensymmetrisch über einen zweielementigen Pfad mit den Kosten 1 und $1|M$ in den Knoten der nächsten Variable x_3 führen.

Nach dieser Vorschrift wird der obere Teil des Graphen für alle n konstruiert und entsprechend mit den Vorkommen der jeweiligen Literale im unteren Teil verknüpft. Die letzte Variable führt demnach in einen Knoten x_{n+1} , der in unserem Beispiel durch x_5 repräsentiert wird. Von diesem führt wie zuvor eine Kante mit Kosten L direkt zum Zielknoten. Darüber hinaus gibt es nur eine weitere ausgehende Kante mit Kosten 1, die gefolgt von einer Kante mit Kosten $1|M$ in den bereits angesprochenen Knoten u des unteren Teils des Graphen führt. Die Konstruktion des Graphen ist damit abgeschlossen.

Wir betrachten die Formel F erneut als Spiel zwischen Spieler und Gegenspieler und nehmen weiter ohne Beschränkung der Allgemeinheit an, dass die Quantoren alternierend in der Form $\exists x_1 \forall x_2 \exists x_3 \forall x_4 \dots \exists x_{n-1} \forall x_n$ vorliegen. Wie beim eingangs erwähnten Spiel GEOGRAPHIE bedeutet dies, dass abwechselnd eine Entscheidung von einem Spieler und dann von einem Gegenspieler getroffen wird. Es ist nun zu zeigen, dass der Graph genau dann eine Graph-Strategie für den Spieler ermöglicht, wenn die Formel F erfüllbar ist.

Gehen wir daher zuerst davon aus, dass F erfüllbar ist: In diesem Fall muss eine Graph-Strategie existieren, mit der der Spieler gegen jeden Gegenspieler gewinnt. Der Spieler versucht also eine s - t -Pfad durch den Graphen G zu finden, dessen Kosten das r -fache einer optimalen Lösung nicht übersteigen. Wie ist nun jedoch der genaue Zusammenhang zwischen einer Formel F in der QBF-Instanz Q und dem Graphen G der DVG-Instanz gegeben?

Jede der p Klauseln ist, wie oben besprochen, im unteren Teil von G repräsentiert. Es existiert zu Beginn des Spiels ein Pfad von s zu t durch jeden dieser Klauselstränge, der möglicherweise ausschließlich aus Kanten der Länge 1 besteht. Die Länge d dieses Pfades beträgt $3n \leq d \leq 4n$. Auf jedem dieser Pfade existieren jedoch insbesondere in jeder Klausel Kanten, die durch den Gegenspieler auf den Wert M erhöht werden können. Genau dazu will der Spieler gemäß seiner Gewinnstrategie den Gegenspieler zwingen. Wir wollen dies anhand unseres Beispiels aus Abbildung 2.1 verdeutlichen. Der Spieler wählt zum Beispiel den Teilpfad am x_1 -Komplex, der mit **true** beschriftet ist. Bezogen auf die Formel F , die durch diesen Graphen repräsentiert wird, käme dies einem Erfüllen der ersten Klausel $(x_1 \vee x_2 \vee x_3)$ gleich. Auf den Graphen bezogen bedeutet dies, dass er im oberen Teil des Graphen, an Knoten x_1 angekommen, nach links verzweigt und dem Pfad für **true** mit Kosten K folgt. Dies führt ihn zum ersten Klausel-Komplex, den er nun erreicht hat. Der Gegenspieler ist gezwungen die mit $1|M$ gewichtete Kante vom Knoten des Literalrepräsentanten x_1 zum nächsten Knoten des Literalrepräsentanten x_2 auf M zu setzen, da der Spieler sonst die Kante mit Kosten K zum Zielknoten t benutzen könnte, um das Spiel zu gewinnen. Dies ergibt sich aus der Überlegung, dass der Spieler nun einen s - t -Pfad mit geringen Kosten als $Kn = m^3n$ gefunden hätte. Da ein kürzester Pfad

vom Startknoten s zum Zielknoten t jedoch mindestens die Kosten n verursacht ist leicht einsehbar, dass der durch die Graph-Strategie des Spielers gefundene Weg mit Kosten $Kn = m^3n \ll m^4n = rn$ die r -fachen Kosten einer optimalen Lösung nicht überschreitet.

Der Graph ist so konstruiert, dass der Spieler nun zu allen Knoten gelangt, die Literale für die Belegung $x_1 = \mathbf{true}$ repräsentieren. Da diese in unserem Beispiel für die erste Klausel nicht existent sind, geht es direkt weiter zum ersten Knoten des x_2 -Komplexes. Dieser ist so konstruiert, dass er bezogen auf die Instanz Q eine universelle Variable repräsentiert. Für das Spiel, dass 2 Spieler bei einer QBF-Instanz mit alternierenden Quantoren spielen können bedeutet dies, dass nun der Gegenspieler eine Entscheidung treffen muss. Er entscheidet sich beispielsweise für $x_2 := \mathbf{false}$, um die zweite Klausel nicht bereits zu erfüllen. Für unseren Graphen bedeutet dies, dass der Gegenspieler die linke der beiden ausgehenden Kanten auf M und die rechte auf 1 setzt. Er kann nicht beide auf M setzen, da ansonsten der Pfad von s nach t über alle mit 1 gewichteten Kanten entfällt und der Spieler die Möglichkeit hat, nun den Pfad mit Kosten L zum Zielknoten t zu traversieren, der eine Einhaltung des Faktors r gewährleisten würde. Es würde nämlich in diesem Fall für die Kosten des kürzesten s - t -Pfades $d := 1 + L = 1 + m^6$ gelten. Der Spieler hat bis zu diesem Punkt Kosten von nicht mehr als $n \cdot p \cdot K = n \cdot p \cdot m^3$ verursacht. Diese Kosten ergeben sich, falls *jedes* Literal in *jeder* Klausel vorkommt, was durch die geforderte Form von F in 3KNF für mehr als drei Literale demnach bereits nicht mehr erreicht wird. Wegen $n = \frac{m}{7} - p$ und $p = \frac{m}{7} - n$ gilt $n \cdot p \cdot m^3 \leq m^5$. Unter Berücksichtigung der noch ausstehenden Kosten der mit L gewichteten Kante zum Zielknoten t , ergeben sich somit Gesamtkosten von

$$n \cdot p \cdot K + L \leq n \cdot p \cdot m^3 + m^6 \leq m^5 + m^6 \ll m^4 \cdot (m^6 + 1) = r \cdot (L + 1). \quad (2.1)$$

Damit würde der Spieler also das Verhältnis r einhalten. Der Gegenspieler muss demnach mindestens eine der beiden Kanten mit 1 gewichten, dies könnte im Falle des Beispiels aus Abbildung 2.1 zum aktuell betrachteten Zeitpunkt die rechte Kante mit Kosten $1|M$ sein. Der Spieler traversiert nun alle Kanten im Graphen G die Klauseln mit einem Vorkommen von $\overline{x_2}$ repräsentieren, um so den Faktor r einhalten zu können. Dieses Verhalten entspricht dem Erfüllen von Klauseln in der zugehörigen Instanz Q einer QBF. Diese Kantentraversierung verursacht pro Klausel-Komplex Kosten von K und führt den Spieler anschließend mit weiteren Kosten 1 zum ersten Knoten des x_3 -Komplexes. Dieser repräsentiert bezogen auf das QBF-Spiel wieder eine Entscheidung des Spielers, weshalb analog zur Situation an Knoten x_1 fortgefahren wird. Dies wiederholt sich so lange, bis alle n Variablen-Komplexe traversiert wurden und der Spieler am Knoten x_{n+1} angekommen ist. In unserem Beispiel ist dies der erste Knoten am x_5 -Komplex.

Da wir davon ausgegangen sind, dass Q erfüllbar ist, kennt der Spieler eine Graph-Strategie, die es ihm erlaubt einen s - t -Pfad zu traversieren, der die r -fachen Kosten einer optimalen Lösung nicht übersteigt. Bezogen auf das QBF-Spiel bedeutet dies die Kenntnis der sukzessiven Belegung der Existenzvariablen x_i , die F für jede Wahl der im Schritt zuvor

gewählten universellen Variable x_{i-1} durch den Gegenspieler erfüllt. Demnach wurden auch alle Knoten der Klauselstränge im unteren Teil des Graphen besucht und der Gegenspieler musste mindestens eine Kante jedes Stranges auf M setzen. Andernfalls hätte der Spieler vorzeitig einen s - t -Pfad gefunden, der die r -fachen Kosten einer optimalen Lösung nicht überschreitet. Somit existiert zu diesem Zeitpunkt kein Pfad von u zu t mit Kosten geringer als M . Der Spieler kann nun nach der Rechnung aus Gleichung (2.1) die L -Kante von x_{n+1} zu t traversieren und hat demnach eine Graph-Strategie gefunden, mit der er das Problem DVG unter Berücksichtigung des zugehörigen Faktors r löst.

Betrachten wir nun den Fall, in dem Q nicht gültig ist. Dann muss eine Graph-Strategie für den Gegenspieler existieren, mit der er den Spieler stets zu Wegkosten zwingen kann, die den Faktor r bezüglich eines kürzesten Weges von s nach t in G übersteigen. Hierfür betrachten wir exemplarisch den Graphen aus Abbildung 2.4, der die unerfüllbare Instanz $Q := \exists x_1 \forall x_2 \exists x_3 \forall x_4 : (x_1 \vee x_2 \vee x_4) \wedge (\bar{x}_1 \vee x_2 \vee x_4)$ repräsentiert. Egal wie sich nun der Spieler verhält - es ist dem Gegenspieler immer möglich ihn zu Wegkosten zu zwingen, die das r -fache eines optimalen Pfades übersteigen. Es ist auf Grund der Konstruktion sichergestellt, dass der Spieler einen der beiden Klauselstränge im unteren Teil des Graphen nicht traversieren kann. Der Spieler könnte versuchen wenigstens einen der Klauselstränge zu traversieren, um den Gegenspieler dazu zu zwingen die Kosten eines Pfades vom Knoten u zum Knoten t über diesen Klauselstrang auf M zu wählen. Dann erreicht der Spieler jedoch den ersten Knoten des Variablen-Komplexes x_5 ohne den anderen Klauselstrang traversiert zu haben. In diesem musste der Gegenspieler die Kosten nicht auf M wählen. Deshalb muss ein Pfad von u zu t mit Kosten 7 existieren. Nun existiert ein Pfad von s zu u mit Kosten $4n$. Höhere Kosten ergäben sich nur dann, wenn der Spieler an einer Existenzvariablen x_i den Weg über die mit 1 gewichtete Kante auf dem Pfad zum nächsten Variablen-Komplex x_{i+1} wählte und sich somit gegen eine Traversierung des unteren Teiles des Graphen entscheiden würde. Der Spieler erhofft sich ebenfalls am nächsten Variablen-Komplex x_{i+1} anzukommen. In diesem Fall würde jedoch der Gegenspieler die nun folgende Kante auf diesem Pfad mit Gewichtung $1|M$ auf den höheren der beiden Werte setzen und den Spieler somit zu Gesamtkosten von mindestens M zwingen. Dann verliert der Spieler allerdings ebenfalls das Spiel, da, wie bereits oben erwähnt, stets ein s - t -Pfad mit Kosten $L + 1$ existiert und somit gilt:

$$r \cdot (L + 1) = m^4 \cdot (m^6 + 1) \ll m^{12} = M. \quad (2.2)$$

An einer universellen Variablen ergibt sich dieselbe Argumentation. Auch hier kann der Spieler stets zu Kosten von M gezwungen werden, wodurch er die r -fachen Kosten eines kürzesten Weges von s nach t nicht einhält und das Spiel verliert.

Der Spieler hat nun am Knoten x_5 (bzw. allgemein: x_{n+1}) zwei Möglichkeiten. Er kann zum einen der mit 1 gewichteten Kante auf dem Pfad zum Knoten u folgen. Dann setzt der Gegenspieler die Kosten der nächsten Kante auf dem Pfad zu u auf M , wodurch der

Spieler nach der Argumentation aus Gleichung (2.2) von oben das Spiel verliert. Traversiert er die Kante mit Kosten L zum Zielknoten t , so übersteigen die Kosten den zulässigen Faktor r , da $L = m^6 \gg m^4 \cdot (7 + 4n)$ gilt und der Spieler hat ebenfalls verloren. Demnach existiert immer eine Graph-Strategie für den Gegenspieler, die bei Unerfüllbarkeit von Q jeden Spieler zwingt, den Faktor r zu überschreiten.

□

2.2 Transformation des DVG zum CTP

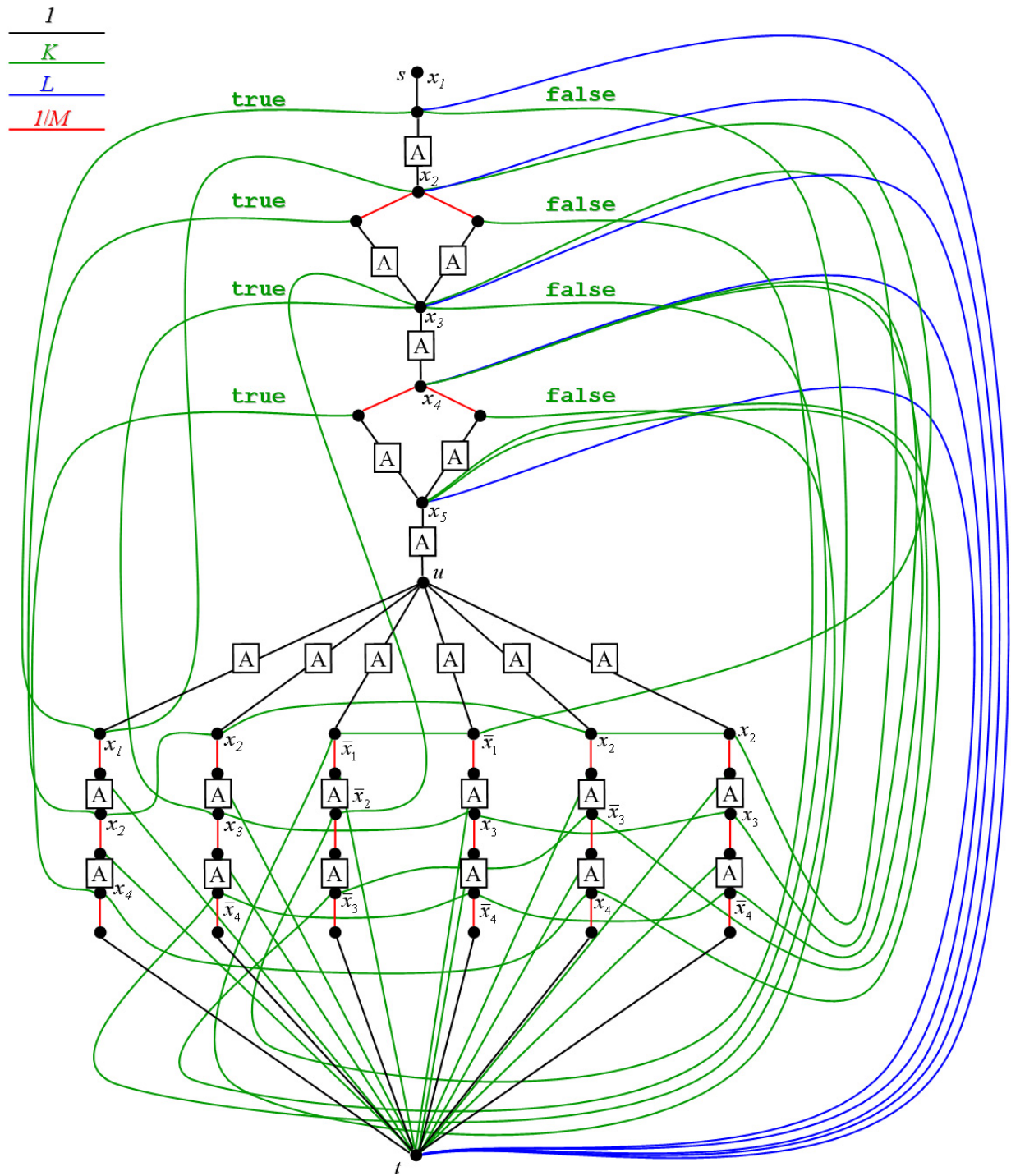
Wir zeigen nun, wie obige Konstruktion für den Graphen G in einen Graphen G' des CTP-Entscheidungsproblems überführt werden kann. Bei diesem handelt es sich im Gegensatz zum Graphen G des DVG um einen ungerichteten Graphen. Außerdem existieren nicht zwei Gewichtsfunktionen, sondern die Kanten können zusätzlich zur Gewichtung mit einem Fragezeichen versehen sein. Dieses Fragezeichen symbolisiert, dass die Kante evtl. nicht passierbar ist, was einer Gewichtung von ∞ gleich käme. Wie schon vom DVG bekannt ist die Passierbarkeit einer Kante immer erst bei Erreichen eines ihrer inzidenten Knoten bekannt.

Hierzu konstruieren wir eine neue Menge F' von Klauseln, die gegeben ist durch drei Kopien der Menge F , gefolgt von einer Menge tautologischer Klauseln $x_1 \vee \overline{x_1}, \dots, x_n \vee \overline{x_n}$, sowie weiteren drei Kopien der Klauselmenge F . Da G' ungerichtet sein muss, benötigen wir ausserdem eine Hilfskonstruktion, die in Abbildung 2.5 dargestellt ist. Die genaue Funktionalität erklären wir weiter unten. Es sei nun, wie in Abbildung 2.2 dargestellt, D der gerichtete Graph, der entsteht, wenn wir gemäß voriger Konstruktion F' als korrespondierende Wahrheitsformel verwenden. Es gibt also, wie aus Abbildung 2.3 ersichtlich, weiterhin im unteren Teil des Graphen einen Pfad, der über Kanten der Länge K alle Köpfe der Klauseln mit einem Vorkommen des Literals x_i oder $\overline{x_i}$ miteinander verbindet. Dieser Pfad durch die Literale x_i oder $\overline{x_i}$ lässt sich nun durch gleichmäßiges Hinzufügen an seinem Anfang und seinem Ende unter Zuhilfenahme von weiteren K -Kanten so erweitern, dass jeder Pfad durch die Literale exakt die Länge

$$T = 6mK$$

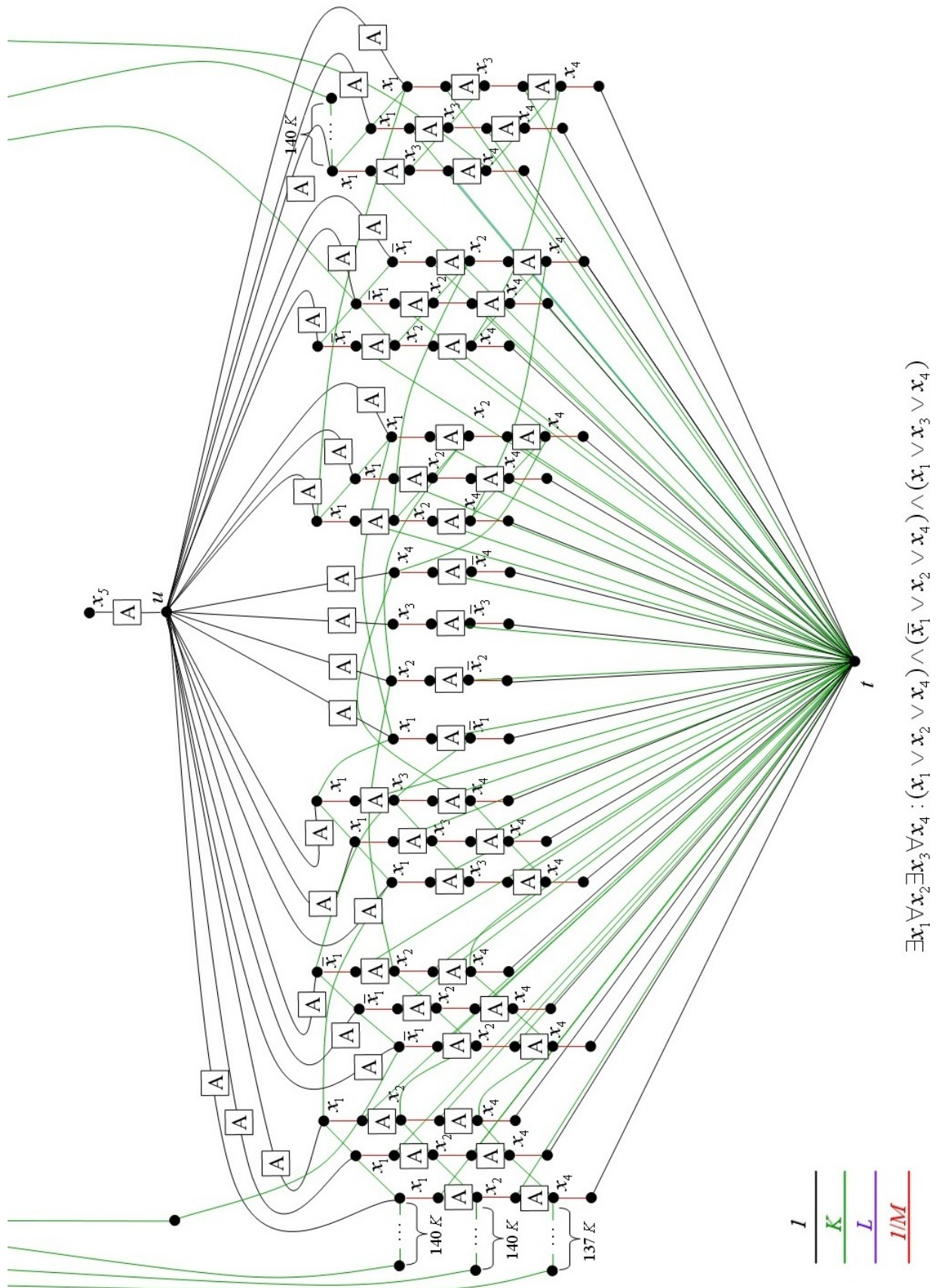
aufweist.

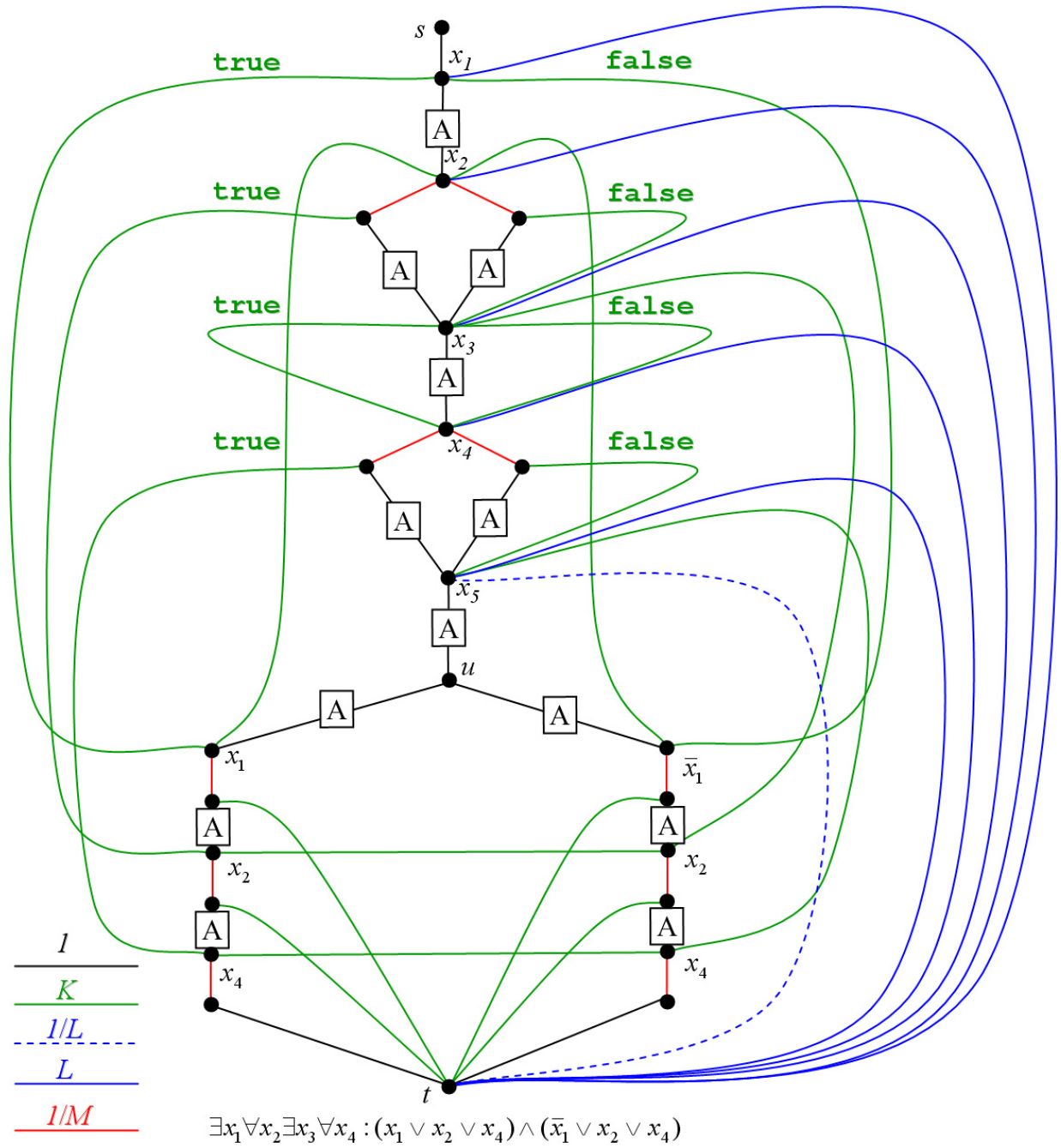
Um die Problematik genauer zu verstehen, betrachten wir Abbildung 2.4. In diesem Beispiel ist die für die Konstruktion des Graphen zu Grunde gelegte QBF-Instanz Q unerfüllbar. Es wäre nun innerhalb des QBF-Spiels unerheblich, welche Entscheidung der Spieler für x_1 wählt - der Gegenspieler kann stets die universellen Variablen x_2 und x_4 auf **false** setzen und somit erzwingen, dass die Formel unerfüllbar wäre.



$$\exists x_1 \forall x_2 \exists x_3 \forall x_4 : (x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee x_3 \vee \bar{x}_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_3 \vee \bar{x}_4) \wedge (x_2 \vee \bar{x}_3 \vee x_4) \wedge (x_2 \vee x_3 \vee \bar{x}_4)$$

Abbildung 2.2: DVG als Instanz des CTP-Entscheidungsproblems

Abbildung 2.3: schematische Repräsentation der Formel F'

Abbildung 2.4: Beispiel einer unerfüllbaren Instanz Q

In dem in Abbildung 2.4 dargestellten Graphen gäbe es im gerichteten Fall immer einen Klauselstrang, den der Spieler nicht passieren kann. Da nun aber die gerichteten durch ungerichtete Kanten ersetzt wurden, hat der Spieler die Möglichkeit, beide Wahrheitswerte von ein und derselben Variable für die Erfüllung der Formel zu verwenden. Dies gelingt ihm durch Backtracking nach einer Entscheidung für den Wahrheitswert der Variable x_1 . Der Spieler kann vom Startknoten s ausgehend mit einem Kostenaufwand von $4K + 1$ den ersten Knoten des x_2 -Komplexes erreichen und gleichzeitig den Gegenspieler dazu zwingen, beide Klauselstränge im unteren Teil des Graphen als erfüllt zu markieren. Dies bedeutet, es existiert kein Pfad von u durch den unteren Teil des Graphen, der ausschließlich aus Kanten mit geringeren Kosten als M besteht. Der Gegenspieler hat nun zwei Möglichkeiten: Entweder er setzt beide Variablen x_2 und x_4 auf **false**, um somit keine der beiden Klauseln zu erfüllen oder er ermöglicht dem Spieler den noch unbesuchten Klauselstrang zu erfüllen. Damit hätte der Spieler entgegen der Erfüllbarkeit von Q das Problem DVG gelöst. Setzt der Gegenspieler jedoch beide Variablen x_2 und x_4 auf **false**, so ist es dem Spieler möglich unter Gesamtkosten von $3 + 7K$ den Knoten x_5 zu erreichen. Die Kosten der Kante von x_5 zu t können nun vom Gegenspieler wahlweise auf 1 oder L gesetzt werden. In beiden Fällen erreicht der Spieler jedoch mit Kosten $3 + 7K = 3 + 7m^3 < m^4 \cdot 16 = r \cdot 16$ bzw. $3 + 7K + L = 3 + 7m^3 + m^6 < m^4 \cdot (1 + m^6) = r \cdot (1 + L)$ den Zielknoten t und erfüllt das Problem DVG mit zugehörigem Faktor r .

Desweiteren ändern wir die Beschriftung der Kante (x_{n+1}, t) von 1 auf $1/L$. In Abbildung 2.5 ist die Konstruktion eines Hilfsgraphen mit ungerichteten Kanten angegeben, der genau das Gleiche leistet, wie die gerichtete Variante aus Abbildung 2.1 auf Seite 23. Es existieren

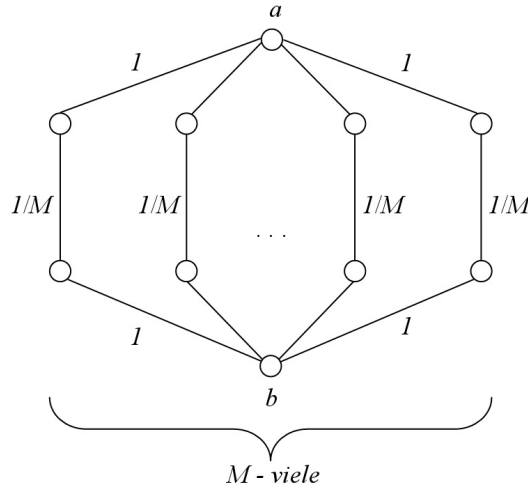


Abbildung 2.5: Hilfskonstrukt A zur Imitation von gerichteten Kanten

M Pfade von a nach b . Der Gegenspieler kann die Kanten also so gewichten, dass der Suchende erst nach $M/2$ vielen Pfaden denjenigen mit Kosten 3 zum Knoten b gefunden hat.

Damit besteht die Möglichkeit den Spieler auch bei Verwendung von ungerichteten Kanten zu Kosten M für die Traversierung dieses Graphabschnittes zu zwingen. Der Gegenspieler ist bei dieser Darstellung mächtiger als zuvor bei der gerichteten Kante mit der Gewichtung $1/M$ in Abbildung 2.1 auf Seite 23. Dies hat zur Folge, dass er für eine Traversierung vom Knoten a zum Knoten b in Abbildung 2.5 Kosten von $2(M/2 - 1) + 3 = M + 1$ konstruieren, während immer noch $M/2 - 1$ Pfade zwischen den Knoten a und b vorhanden sind, die theoretisch mit Kosten 3 überquert werden können. Außerdem hat der Gegenspieler so die Möglichkeit zu verhindern, dass der Spieler auf entgegengesetztem Wege diesen Graphabschnitt vom Knoten b zum Knoten a traversiert. Da es sich um ungerichtete Kante handelt, wäre dies sonst durchaus möglich und muss daher verhindert werden. Nachdem wir nun begründet haben, warum das Hilfskonstrukt A aus Abbildung 2.5 gerade das Gewünschte leistet ersetzen wir in unserem gerichteten Graphen D alle Vorkommen einer Kantenfolge mit Kosten 1 gefolgt von $1/M$, die in x_{i+1} oder u münden und jede mit 1 gewichtete Kante im unteren Teil des Graphen durch dieses Konstrukt aus ungerichteten Kanten. Dadurch erreichen wir, dass der Suchende die Kanten nicht rückwärts traversieren kann, da der Gegenspieler die Kosten bis auf M erhöhen kann. Alle übrigen gerichteten Kanten werden nun noch in ungerichtete Kanten umgewandelt. Den daraus resultierenden Graphen bezeichnen wir mit G . Mit d benennen wir die Länge des kürzesten s - t -Pfades, der entsteht, wenn der Gegenspieler alle Kantengewichte auf die niedrigere der beiden Alternativen setzt. Da vom Startknoten s zum ersten Knoten des x_1 -Komplexes und außerdem vom ersten Knoten eines x_i -Komplexes zum ersten Knoten eines x_{i+1} -Komplexes jeweils 3 Kanten mit Gewichtung 1 zu passieren sind, ergibt sich folgende Rechnung: Der kürzeste Pfad über alle mit 1 gewichteten Kanten von s über die x_i -Komplexe zu t Kosten d in Höhe von $3n \leq d \leq 4n$ verursacht. Damit die Konstruktion gelingt wählen wir als gefordertes Verhältnis $r := n(2 + T)/d$ und zeigen im Folgenden, dass wir eine Instanz des CTP-Entscheidungsproblems konstruiert haben, die sich äquivalent zum Doppelt Gewichteten Graphen aus obiger Reduktion verhält. Es gilt erneut zu zeigen, dass genau dann eine Graph-Strategie für den Suchenden existiert, wenn Q erfüllbar ist.

Wir nehmen an, Q sei gültig. Der Suchende wird erneut die Strategie verfolgen alle Klauseln zu erfüllen, um den Gegenspieler damit zu zwingen, die Kosten des kürzesten Weges signifikant zu erhöhen. Sollte der Gegenspieler an einer universellen Variable x_i die Kosten beider ausgehenden Kanten mit Gewichtung $1/M$ auf M setzen, so ist die Distanz eines kürzesten Weges mindestens K . Der Spieler hat vom Startknoten s bis zu diesem Zeitpunkt eine Distanz von maximal $n(1 + T)$ zurückgelegt. Nun kann er die mit L

gewichtete Kante zum Zielknoten t traversieren, wodurch er den Faktor r wie folgt einhält:

$$\begin{aligned}
& n(1 + T) + L \\
&= n(1 + 6m^4) + m^6 \\
&< m(1 + 6m^4) + m^6 && \text{wegen } m = 7(n + p) \\
&= m + 6m^4 + m^6 \\
&< \frac{n(2 + 6m^4)}{4n} m^3 && \text{wegen } m = 7(n + p) \\
&\leq \frac{n(2 + T)}{d} K \\
&= rK
\end{aligned}$$

Tritt dieser Fall nicht ein, kommt der Spieler mit Gesamtkosten $n(1 + T)$ am Knoten x_{n+1} an. Hier kann der Gegenspieler die Kante zwischen x_{n+1} und t entweder auf 1 oder L setzen. Entscheidet er sich für 1, traversiert der Spieler diese Kante, wodurch seine Kosten maximal $n(2 + T)$ betragen. Dadurch würde er bei Kosten d , für einen kürzesten Pfad von s nach t in G , das Verhältnis r wegen $n(2 + T) \leq \frac{n(2+T)}{d}d = rd$ einhalten. Setzt der Gegenspieler die Kosten dieser Kante auf L , betragen die Kosten eines kürzesten s - t -Pfades in G mindestens K , da der Spieler alle Klauselstränge im unteren Teil des Graphen erfolgreich blockieren ließe. Nach obiger Rechnung ist auch hier das Verhältnis r eingehalten worden.

Wir wollen nun annehmen, dass die Formel Q *nicht* erfüllbar sei. Die Argumentation verläuft sehr ähnlich zu der Reduktion aus Abbildung 2.1 auf Seite 23. Es ergibt sich jedoch insofern ein Unterschied, da wir nun einen ungerichteten Graphen betrachten, der es dem Suchenden erlaubt, eine getroffene Entscheidung unter Berücksichtigung von weiteren Kosten rückgängig zu machen. Wir zeigen also erneut, dass für den Gegenspieler gegen jeden Spieler eine Graph-Strategie existiert, mit der der Spieler zu einem höheren Faktor als r gezwungen werden kann. Wir müssen also garantieren, dass der Suchende dieselben Kosten wie im gerichteten Graphen verursacht, obwohl er nun die Möglichkeit hat, Schritte zurückzugehen. Das gewährleisten wir durch die trivialen Klauseln $x_i \vee \bar{x}_i$ in der Mitte unserer erweiterten Formel F' . Der Suchende muss diese Klauseln in jedem Fall besuchen, um F' zu erfüllen. Dann hat er aber bereits eine Distanz von mindestens $T/2$ zurückgelegt. Sollte er sich nun dazu entscheiden zurückzugehen, erfordert dies Gesamtkosten $\geq T$ und es ergeben sich somit keine geringeren Kosten als im gerichteten Graphen.

Des Weiteren muss verhindert werden, dass der Suchende am Knoten einer Existenzvariablen x_i beide Pfade, also den für x_i und für $\bar{x}_i$, verfolgt, um somit eventuell effektiver alle Klauseln der Formel F' zu erfüllen.

Betrachten wir diesen Fall etwas genauer: Wir haben gefordert, dass an beiden Seiten der Pfad durch ein Literal vom Knoten x_i zum Knoten x_{i+1} exakt die Länge $T = 6mK$ hat. Dies erreichten wir durch gleichmäßiges Hinzufügen von Kanten der Länge K auf

beiden Seiten, deren Anzahl wir mit $\hat{K}_a$ bezeichnen. Analog bezeichne $\hat{K}_i$ die Anzahl an K -Kanten genau eines Literalpfades im Inneren des unteren Teilgraphen. Somit ergibt sich unweigerlich für die Gesamtanzahl $\hat{K}$ für einen Durchlauf von einer Variablen x_i zu ihrem Nachfolger x_{i+1} der Wert $\hat{K} = \hat{K}_i + \hat{K}_a$. Die Anzahl an Kanten im Inneren des unteren Teilgraphen hängt von der Anzahl an Klauseln p' ab, in denen das betreffende Literal vorkommt. Es gilt $\hat{K}_i = 6p'$, da je drei Kanten benötigt werden, um die Kopien einer Klausel auf beiden Seiten der tautologischen Klauseln miteinander zu verbinden. Es ist $m = 7(n + p)$, wobei n die Anzahl der Variablen und p die Anzahl der Klauseln bezeichnet. Damit ergibt sich mit $\hat{K} = 6m$ durch Einsetzen der Werte in obige Formeln

$$\begin{aligned}\hat{K}_a &= 42n + 42p - \hat{K}_i \\ &= 42n + 42p - 6p' \\ &\stackrel{(1)}{\geq} 42n + 36p.\end{aligned}$$

Zu (1): Da ein Literal höchstens in jeder Klausel vorkommen kann, muss gelten $p \geq p'$. Wir betrachten nun das Verhältnis zwischen Außen- und Gesamtkanten, für das gilt:

$$\frac{\hat{K}_a}{6m} \geq \frac{42n + 36p}{42n + 42p} \geq \frac{36n + 36p}{42n + 42p} = \frac{6}{7} = 2\frac{3}{7} \geq 2\frac{3}{9} = 2\frac{1}{3}.$$

Da die Gesamtanzahl $\hat{K}_a$ an äußeren Kanten gleichmäßig auf beide Seiten aufgeteilt wird, lässt sich demnach aussagen, dass mindestens $\frac{1}{3}$ der Gesamtstrecke T für den Besuch einer einzelnen Klausel notwendig ist.

Wenn der Spieler nun am Knoten x_{n+1} ankommt, hat er für jede Variable mindestens die Kosten T und für mindestens eine sogar die Kosten $\frac{4}{3}T$ verursacht. Die Kosten $\frac{4}{3}T$ werden immer dann benötigt, wenn der Spieler versucht die zwei Pfade beider Wahrheitswerte eines Variablenkomplexes x_i zu traversieren. Der dadurch entstandene Weg lässt sich zu Gesamtkosten von mindestens $(n + \frac{1}{3})T$ aufaddieren. Der Gegenspieler gewichtet nun die Kante von x_{n+1} zu t mit 1 und garantiert dadurch einen kürzesten s - t -Pfad der Länge d . Damit überschreitet der Spieler wegen $(n + \frac{1}{3})T > rd$ den einzuhaltenden Faktor und verliert das Spiel. Sollte der Spieler für jede Variable weniger als Kosten $4T/3$ verursacht haben, muss noch eine Klausel im unteren Teil des Graphen unerfüllt sein, über die immer noch ein Weg von s über u zu t bestehend aus Kanten der Länge 1 existiert. Dann setzt der Gegenspieler die Länge der Kante zwischen x_{n+1} und t auf L und zwingt den Spieler somit Kosten $nK + L = nm^3 + m^6 \gg \frac{n(6mK+2)}{d}d = rd$ aufzuwenden. Das Spiel kann durch den Spieler in diesem Fall also ebenfalls nicht gewonnen werden.

Um die Konstruktion abzuschließen, wollen wir veranschaulichen, wie sich aus G eine Instanz G' des CTP-Entscheidungsproblems herleiten lässt, bei der alle Kanten Kosten 1 haben und manche über ein zusätzliches Fragezeichen verfügen. Ein Fragezeichen symbolisiert dabei die Möglichkeit eines Kantenausfalls, der jedoch erst bei Erreichen eines ihrer

inzidenten Knoten bemerkt wird. Da die Kantenlängen in G polynomiell beschränkt sind, kann jede einfach gewichtete Kante aus G durch eine Kette von entsprechenden Kanten der Länge 1 in G' dargestellt werden. Die doppelt-gewichteten Kanten aus G sind Kanten mit Kosten $1|M$ oder $1|L$. Es sei, wie in Abbildung 2.6 dargestellt, (a, b) eine gerichtete doppelt-gewichtete Kante. Diese können wir, wie aus Abbildung 2.7 ersichtlich, durch eine

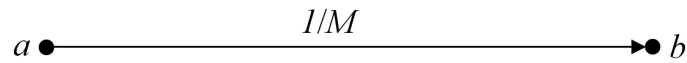


Abbildung 2.6: gerichtete doppelt-gewichtete Kante (a, b)

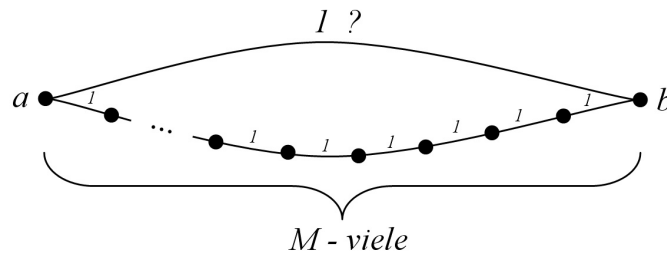


Abbildung 2.7: ungerichtetes Hilfskonstrukt $\{a, b\}$

Kante der Länge 1 mit einem Fragezeichen und einem parallel verlaufenden Pfad zwischen a und b aus Kantenstücken der Länge 1 ohne Fragezeichen mit entsprechender Länge M oder L darstellen.

Damit ist die Konstruktion des CTP-Entscheidungsproblems vollständig.

Kapitel 3

Die Strategien Backtrack und Greedy

3.1 Darstellung der Strategien

In diesem Kapitel werden zwei Strategien zum Lösen des k -Canadian Traveller Problem (k -CTP) untersucht. Beide Strategien verfolgen das Ziel, in einem bekannten Graphen einen möglichst kurzen Weg von einem Startknoten zu einem Zielknoten zurückzulegen. Zusätzlich gibt es eine den Strategien im Voraus bekannte Konstante, die angibt, wie viele Kanten maximal blockiert sein können. Wir betrachten daher im Folgenden stets einen ungerichteten, gewichteten Graphen $G = (V, E)$ mit Start- und Zielknoten $s, t \in V$, mit $s \neq t$, sowie eine obere Schranke $k \in \mathbb{N}$ blockierter Kanten in G . Der Konvention folgend bezeichnen wir mit $\mathbb{N}$ die positiven natürlichen Zahlen. Eine Strategie exploriert den Graphen dabei im Sinne eines Online-Problems. Die Information, ob eine Kante überquert werden kann oder aber blockiert ist, wird immer erst bei Erreichen eines ihrer inzidenten Knoten bekannt. Wir wollen jedoch davon ausgehen, dass die Wahl der k blockierten Kanten so ausfällt, dass der Zielknoten t vom Startknoten s erreichbar bleibt. Es existiert also nach Wegfall von k Kanten immer noch ein Pfad π von s nach t mit $|\pi(s, t)| < \infty$. Diese Annahme ist sinnvoll, da wir die Güte unserer Strategie mit einem kompetitiven Faktor c bewerten wollen. Nach Definition 1.7 ist eine Strategie c -kompetitiv, wenn es eine additive Konstante $\alpha > 0$ gibt, so dass für alle Probleminstanzen $P \in \Pi$ die Ungleichung

$$S(P) \leq c \cdot \text{OPT}(P) + \alpha$$

erfüllt ist. Eine untere Schranke für den kompetitiven Faktor kann dabei durch einen Gegenspieler gefunden werden, der die Strategie kennt und die blockierten Kanten in einem Graphen so auswählt, dass der relative Umweg verglichen mit der optimalen Lösung maximiert wird. Nach Definition 1.7 ist es einer Strategie S gestattet, beliebig große initiale Anfangskosten $\alpha > 0$ zu verursachen, die keinen Einfluss auf c haben. Damit wir dennoch eine untere Schranke für den kompetitiven Faktor von S angeben können, ist es wichtig, dass wir Probleminstanzen $P \in \Pi$ mit beliebig großen optimalen Kosten $\text{OPT}(P)$ wählen, für die $S(P) \geq c \cdot \text{OPT}(P)$ gilt. Für diese Probleminstanzen gilt folgendes Lemma:

Lemma 3.1. *Sei Π ein Problem und S eine Strategie, die jede Instanz $P \in \Pi$ korrekt löst. Seien $S(P)$ die Kosten, die S verursacht, und $\text{OPT}(P)$ die Kosten einer optimalen Lösung von P . Dann gilt:*

$$\begin{aligned} &\forall b > 0 \exists P \in \Pi : \text{OPT}(P) \geq b \text{ und } S(P) \geq c \cdot \text{OPT}(P) \\ &\Rightarrow S \text{ ist nicht besser als } c\text{-kompetitiv.} \end{aligned}$$

Beweis. Es gelte die Voraussetzung vor dem Folgerungspfeil und wir nehmen an, $S(P)$ sei besser als c -kompetitiv. Dann existiert ein $\varepsilon > 0$ und nach Definition 1.7 ein $\alpha > 0$, so dass für alle Probleminstanzen $P \in \Pi$ gilt:

$$S(P) \leq (c - \varepsilon) \cdot \text{OPT}(P) + \alpha. \quad (*)$$

Nach Voraussetzung existiert für $b = \frac{\alpha}{\varepsilon}$ eine Probleminstanz P mit

$$\text{OPT}(P) \geq b \text{ und } S(P) \geq c \cdot \text{OPT}(P).$$

Daraus können wir folgern:

$$\begin{aligned} S(P) &\geq c \cdot \text{OPT}(P) && \text{nach Voraussetzung} \\ &> c \cdot \text{OPT}(P) - \varepsilon && \varepsilon > 0 \\ &= c \cdot \text{OPT}(P) - \varepsilon \frac{\alpha}{\varepsilon} + \alpha \\ &\geq c \cdot \text{OPT}(P) - \varepsilon \cdot \text{OPT}(P) + \alpha && \text{OPT}(P) \geq \frac{\alpha}{\varepsilon} \\ &= (c - \varepsilon) \cdot \text{OPT}(P) + \alpha. \end{aligned}$$

Dies ist ein Widerspruch zu (*). Daraus folgt, dass S nicht besser als c -kompetitiv ist.

□

Auf dieses Lemma werden wir im weiteren Verlauf der Arbeit zurückgreifen, wenn es darum geht, eine untere Schranke für den kompetitiven Faktor einer Strategie herzuleiten.

3.1.1 Die Strategie Backtrack

Gegeben sei ein ungerichteter, gewichteter Graph $G = (V, E)$ und eine Konstante $k \in \mathbb{N}$, die die Anzahl maximal blockierter Kanten angibt. Die Strategie *Backtrack* sei folgendermaßen definiert: Sie beginnt an einem festen Startknoten $s \in V$ und berechnet alle kürzesten Wege vom Start- zu einem festen Zielknoten $t \in V$ und entscheidet sich für einen.

Da ein kürzester Weg nicht zwingend eindeutig ist, sondern beliebig viele kürzeste Wege vom Startknoten s zum Zielknoten t existieren können, handelt es sich bei *Backtrack* um eine nicht eindeutige Strategie, die einen dieser Wege auswählt. In den meisten Fällen

werden wir die Szenarien so gestalten, dass keine Mehrdeutigkeit betrachtet werden muss. Lässt sich dies jedoch nicht vermeiden, so muss um zur Eindeutigkeit zu gelangen eine tie-breaking-rule eingeführt werden, die deterministisch genau einen dieser kürzesten Pfade auswählt. Sollten wir von einer tie-breaking-rule Gebrauch machen, werden wir diese an der entsprechenden Textstelle einführen. Wie bereits oben besprochen, werden wir stets nur von einer Strategie sprechen, wenngleich sich eine (unter Umständen nicht eindeutige) Strategie dahinter verbirgt.

Betrachten wir nun die weitere Vorgehensweise der Strategie *Backtrack*. Wir nehmen dazu an, dass sie sich für einen kürzesten Weg entschieden hat. Sie verfolgt diesen Weg, bis sie entweder t erreicht oder zu einem ersten Knoten $v_1 \in V$ gelangt, der inzident zu einer blockierten Kante $e_1 \in E$ ist, über die dieser kürzeste Weg führt. In diesem Fall geht sie den bis hierhin zurückgelegten Weg, der nach Konstruktion immer ein kürzester ist, zurück zum Startknoten s , und berechnet erneut alle kürzesten Wege $\pi_{G_1}^{\min}(s, t)$ auf dem Restgraphen $G_1 := (V, E \setminus \{e_1\})$. Nach einer tie-breaking-rule entscheidet sie sich für einen dieser kürzesten Wege und läuft auf diesem erneut von s zu t , bis sie entweder t erreicht oder auf Grund einer weiteren blockierten Kante auf diesem gerade berechneten Pfad zum Startknoten s zurückkehrt. Sei e_i die i -te gefundene blockierte Kante, dann sei $G_i := (V, E \setminus \{e_1, \dots, e_i\})$ der Graph G ohne die ersten i gefundenen blockierten Kanten. Dabei bezeichnen wir mit $G_0 := G$, den Graphen, in dem noch keine blockierte Kante entdeckt wurde. Der Graph G_k hingegen kann keine weitere gesperrte Kante enthalten, da bereits die maximale Anzahl an k möglichen Blockaden entdeckt wurde. Da wir angenommen hatten, dass nur solche Kanten entfernt werden dürfen, so dass Start- und Zielknoten miteinander verbunden bleiben, gilt, dass spätestens jetzt die Strategie vom Startknoten s aus den Zielknoten t über $\pi_{G_k}^{\min}(s, t)$ erreichen kann. Es bezeichne j die Anzahl an blockierten Kanten, die die Strategie nach einer erfolgreichen Suche des Zielknotens t entdeckt hat.

Dadurch ergeben sich Gesamtkosten von

$$2 \sum_{i=0}^{j-1} |\pi_{G_i}^{\min}(s, v_{i+1})| + |\pi_{G_j}^{\min}(s, t)|, \text{ für ein } j \in \{0, \dots, k\}, \text{ mit } G_0 := G.$$

3.1.2 Die Strategie Greedy

Es sei erneut ein ungerichteter, gewichteter Graph $G = (V, E)$ und eine Konstante $k \in \mathbb{N}$, die die maximale Anzahl blockierter Kanten festlegt, gegeben. Auch die *Greedy*-Strategie verfolgt zunächst einen kürzesten Pfad vom Startknoten s zum Zielknoten t . Anders als die Strategie *Backtrack* kehrt die Strategie *Greedy* nicht nach Entdecken einer blockierten Kante $e_i \in E$, mit $i \in \{1, \dots, k\}$, zum Startknoten $s \in V$ zurück, sondern nutzt die Tatsache aus, dass sie sich evtl. bereits dem Ziel genähert hat. Sie versucht daher, die bereits zurückgelegte Wegstrecke zu ihrem Vorteil zu verwenden. Sei also $v_1 \in V$ der Knoten, an dem eine erste blockierte Kante e_1 festgestellt wird. Die Strategie berechnet nun einen

kürzesten Weg $\pi_{G_1}^{\min}(v_1, t)$. Entweder wird sie den Zielknoten $t \in V$ über $\pi_{G_1}^{\min}(v_1, t)$ erreichen oder nach Entdecken einer weiteren blockierten Kante e_2 an einem Knoten $v_2 \in V$ einen kürzesten Pfad $\pi_{G_2}^{\min}(v_2, t)$ von ihrem aktuellen Standort zum Zielknoten t berechnen. Dies führt sie insgesamt maximal k mal aus und erreicht dann über $\pi_{G_j}^{\min}(v_j, t)$ den Zielknoten t . Hierbei bezeichne $j \leq k$ wieder die tatsächliche Anzahl an blockierten Kanten, die die Strategie bei der Suche nach dem Zielknoten t entdeckt hat. Wie bereits in Abschnitt 3.1.1 besprochen muss ein kürzester Weg keineswegs eindeutig sein, weshalb es sich auch bei unserer Definition von *Greedy* um eine nicht eindeutige Strategie handelt. Für die meisten Szenarien, die in den Kapiteln 3 und 4 dargestellt werden, ist ein kürzester Weg jedoch stets eindeutig. Sollte dies an einer Stelle nicht garantiert werden können, so werden wir an der betroffenen Textstelle darauf hinweisen. Wir verwenden daher im Folgenden den ungenaueren aber kürzeren Begriff der Strategie *Greedy*, wenn wir die nicht eindeutige Strategie *Greedy* meinen.

Als Sonderfall dieser Strategie kann es bei entsprechender Gewichtung der Kanten durch-
aus passieren, dass sie denselben Weg wie *Backtrack* auswählt. Als Bezeichnungen für Start-
und Zielknoten wählen wir $v_0 := s$ und $v_{j+1} := t$.

Die entsprechenden Gesamtkosten ergeben sich dann aus

$$\sum_{i=0}^j |\pi_{G_i}^{\min}(v_i, v_{i+1})|, \text{ für ein } j \in \{0, \dots, k\}.$$

In den folgenden Kapiteln werden wir die beiden Strategien auf verschiedenen Graphklassen untersuchen.

3.2 Backtrack und Greedy auf allgemeinen Graphen

Zu Beginn dieses Kapitels wollen wir die beiden Strategien auf allgemeinen Graphen betrachten und eine untere Schranke für den zugehörigen kompetitiven Faktor ermitteln. Wir konstruieren dafür ein Szenario, in dem wir der Strategie als Gegenspieler einen Graphen gerade so vorgeben, dass sie zu möglichst großen relativen Umwegen gezwungen wird. Mit allgemeinen Graphen meinen wir jedoch nur ungerichtete Graphen. Für gerichtete Graphen ergibt sich das Problem, dass die Strategie unter Umständen nicht in der Lage ist, wie durch *Backtrack* gefordert, zum Startknoten zurückzukehren. Dies kann bei einem ungerichteten Graphen nicht vorkommen. Wir betrachten aus dem genannten Grund im weiteren Verlauf dieser Arbeit stets einen ungerichteten Graphen $G = (V, E)$ und eine Konstante $k \in \mathbb{N}$, die eine obere Schranke für die Anzahl an möglichen Kantenblockaden angibt. Wir wählen für den Start- und Zielknoten die intuitiven Bezeichnungen $s, t \in V$ ($s = \text{start}$, $t = \text{target}$). Als Ergebnis zeigen wir für die jeweilige Strategie eine untere Schranke und beweisen, dass diese scharf ist. Es wird sich zeigen, dass die scheinbar intelligentere *Greedy*-Strategie nur einen deutlich schlechteren kompetitiven Faktor als die *Backtrack*-Strategie bei der Lösung des Problems garantieren kann.

3.2.1 Die Strategie Backtrack

Der Aufbau des worst-case-Szenarios erfolgt wie in Abbildung 3.1 dargestellt. Sei Π das

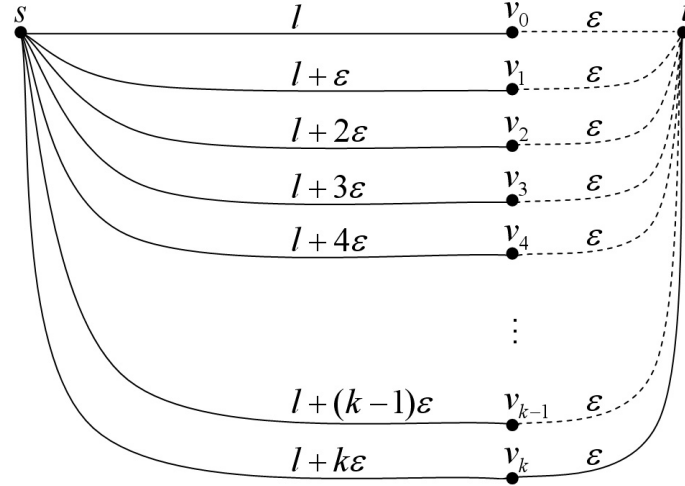


Abbildung 3.1: worst-case-Szenario für die *Backtrack*-Strategie auf allgemeinen Graphen

k -Canadian Traveller Problem auf allgemeinen Graphen. Wir konstruieren für jedes k eine Instanz eines worst-case-Szenarios für die Strategie *Backtrack*. Es sei daher $P_k \in \Pi$ ein worst-case-Szenario bei k ausfallenden Kanten. Wir bezeichnen mit $\text{Backtrack}(P_k)$ die Kosten der *Backtrack*-Strategie und mit $\text{OPT}(P_k)$ die Kosten einer optimalen Lösung zu einer Problem Instanz P_k . Jede Instanz P_k ist gegeben durch einen Graphen $G = (V, E)$, die Start- und Zielknoten $s, t \in V$ und eine Konstante k , die die maximale Anzahl an blockierten Kanten in G angibt. Mit G_i , für $i \in \{0, \dots, k\}$, bezeichnen wir den Graphen, in dem die ersten i blockierten Kanten entfernt sind. Zur Konstruktion dieser Szenarien verfahren wir wie folgt: Am Startknoten s beginnen $k+1$ Kanten $\{s, v_i\}$ der Länge $|s, v_i| = l + i\varepsilon$, mit $i \in \{0, \dots, k\}$ und $l \geq 1$. Von jedem Knoten v_i führt eine Kante $\{v_i, t\}$ der Länge $|v_i, t| = \varepsilon$ zum Zielknoten t . Die *Backtrack*-Strategie traversiert nun vom Startknoten s stets einen kürzesten Weg $\pi_{G_i}^{\min}(s, t) = (s, v_i, t)$ zum Zielknoten t . Erreicht *Backtrack* einen Knoten v_i , mit $i \in \{0, \dots, k-1\}$, blockiert ein Gegenspieler die Kante $\{v_i, t\}$. Dadurch wird die *Backtrack*-Strategie k mal zur Rückkehr zum Startknoten s veranlasst. Danach kann ein Gegenspieler keine weitere Kante blockieren und die Strategie *Backtrack* erreicht über den freien Pfad $\pi_{G_k}^{\min}(s, t) = (s, v_k, t)$ der Länge $|\pi_{G_k}^{\min}(s, t)| = d_{G_k}(s, t) = l + k\varepsilon + \varepsilon$ den Zielknoten t . Dies ist der unterste Pfad in Abbildung 3.1. Für die hier beschriebenen Problem Instanzen $P_k \in \Pi$ summieren sich die Gesamtkosten der *Backtrack*-Strategie auf $\text{Backtrack}(P_k) = 2 \sum_{i=0}^{k-1} |s, v_i| + |s, v_k, t|$. Die Kosten einer optimalen Lösung betragen

hingegen $\text{OPT}(P_k) = |(s, v_k, t)|$. Zu jedem $\delta > 0$ existiert ein $\varepsilon > 0$, so dass gilt:

$$\begin{aligned}
 \forall k \in \mathbb{N} \exists P_k \in \Pi : \frac{\text{Backtrack}(P_k)}{\text{OPT}(P_k)} &= \frac{2 \sum_{i=0}^{k-1} |\{s, v_i\}| + |(s, v_k, t)|}{|(s, v_k, t)|} \\
 &= \frac{2 \sum_{i=0}^{k-1} (l + i\varepsilon) + (l + k\varepsilon + \varepsilon)}{l + k\varepsilon + \varepsilon} \\
 &\geq \frac{(2k+1)l}{l + (k+1)\varepsilon} \\
 &\geq 2k+1 - \delta. \tag{*}
 \end{aligned}$$

Ungleichung (*) gilt nicht für beliebiges ε . Da aber der zuvor angegebene Bruch für $\varepsilon \searrow 0$ von unten gegen $2k+1$ konvergiert, können wir für jedes beliebige $\delta > 0$ den Wert ε klein genug wählen, dass Ungleichung (*) erfüllt ist. Da in obiger Konstruktion l und demnach insbesondere $\text{OPT}(P_k)$ beliebig groß gewählt werden kann, ist nach Lemma 3.1 bewiesen, dass *Backtrack* nicht besser als $(2k+1-\delta)$ -kompetitiv ist. Da außerdem der Wert für δ beliebig nah an 0 gewählt werden kann, gilt sogar eine untere Schranke von $2k+1$. Wie nachfolgender Satz zeigt, ist es umgekehrt jedoch nicht möglich, *Backtrack* zu höheren relativen Kosten zu veranlassen.

Satz 3.2. *Die Strategie Backtrack ist auf allgemeinen Graphen $(2k+1)$ -kompetitiv und nicht c -kompetitiv für alle $c < 2k+1$.*

Beweis. Wir zeigen nun, dass keine Graphen existieren, die die Strategie *Backtrack* zu einem größeren kompetitiven Faktor als $2k+1$ zwingen. Es sei Π das k -Canadian Traveller Problem auf allgemeinen Graphen und $P \in \Pi$ eine beliebige Probleminstanz. Diese sei gegeben durch einen ungerichteten, gewichteten Graphen $G = (V, E)$, die Anzahl $k \in \mathbb{N}$ der maximal blockierten Kanten sowie Start- und Zielknoten $s, t \in V$. Mit $G_i := (V, E_i)$, für $i \in \{0, \dots, k\}$, bezeichnen wir den Graphen, in dem die ersten i blockierten Kanten entfernt sind. Die Kosten, die *Backtrack* nach Entdecken der k -ten blockierten Kante vom Startknoten s aus verursacht, seien $d_{G_k}(s, t)$. Wegen $E_i \supseteq E_{i+1}$ ist kein kürzester Weg von s nach t in G_i länger als ein kürzester Weg von s nach t in G_{i+1} . Damit lässt sich für alle $P \in \Pi$ folgende Ungleichungskette aufstellen: $d_G(s, t) \leq d_{G_1}(s, t) \leq \dots \leq d_{G_k}(s, t)$. Da die *Backtrack*-Strategie auf maximal k blockierte Kanten treffen kann, bevor t erreicht wird, ergibt sich aus dieser Argumentation folgende obere Schranke:

$$\begin{aligned}
 \forall P \in \Pi : \frac{\text{Backtrack}(P)}{\text{OPT}(P)} &\leq \frac{2 \sum_{i=0}^{k-1} d_{G_i}(s, t) + d_{G_k}(s, t)}{d_{G_k}(s, t)} \\
 &\leq \frac{2 \sum_{i=0}^{k-1} d_{G_k}(s, t) + d_{G_k}(s, t)}{d_{G_k}(s, t)} \\
 &= \frac{(2k+1) \cdot d_{G_k}(s, t)}{d_{G_k}(s, t)} \\
 &= 2k+1.
 \end{aligned}$$

□

Nachdem wir den Beweis zu Satz 3.2 fertiggestellt hatten, wurden wir darauf aufmerksam, dass S. Westphal in seiner Arbeit [SW07], die im März dieses Jahres verfasst wurde, die gleiche Problemstellung untersucht und ebenfalls oben genanntes Ergebnis vorstellt.

Wir werden in den späteren Abschnitten dieses Kapitels untersuchen, inwieweit sich die obere Schranke für die Strategie *Backtrack* herabsetzen lässt, wenn Einschränkungen bezüglich der Graphklasse gemacht werden.

3.2.2 Die Strategie Greedy

Abbildung 3.2 veranschaulicht die Konstruktion des im folgenden beschriebenen worst-case-Szenarios für die Strategie *Greedy*. Es sei Π das k -Canadian Traveller Problem auf

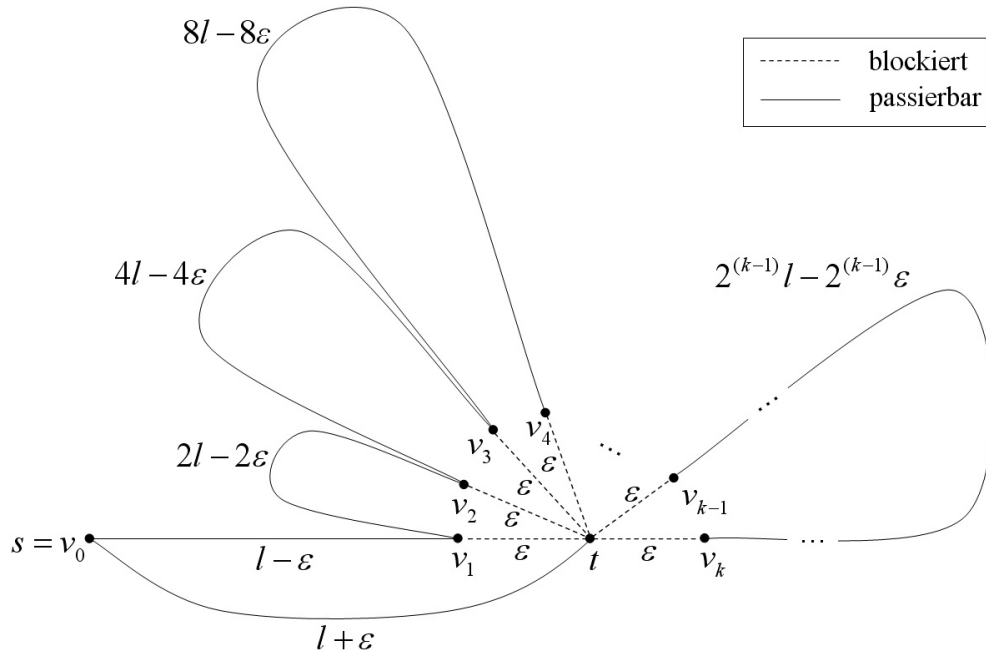


Abbildung 3.2: worst-case-Szenario für die *Greedy*-Strategie auf allgemeinen Graphen

allgemeinen Graphen. Zur Begründung für die Kosten einer unteren Schranke des kompetitiven Faktors konstruieren wir zu jedem $k \in \mathbb{N}$ ein entsprechendes worst-case-Szenario P_k . Eine Probleminstanz P_k ist dabei gegeben durch einen ungerichteten, gewichteten Graphen $G = (V, E)$, eine konstante Anzahl k maximal ausfallender Kanten sowie Start- und Zielknoten $s, t \in V$. Wir bezeichnen mit $\text{Greedy}(P_k)$ die Kosten der *Greedy*-Strategie und

mit $\text{OPT}(P_k)$ die Kosten einer optimalen Lösung auf einer Instanz $P_k \in \Pi$ bei k ausfallenden Kanten. Die Konstruktion eines worst-case-Szenarios gestalten wir wie folgt: Vom Startknoten s führt eine Kante $\{s, t\}$ der Länge $|\{s, t\}| = l + \varepsilon$ sowie ein Pfad (s, v_1, t) zum Zielknoten t . Dieser besteht aus einer ersten langen Kante $\{v_0, v_1\}$, mit $v_0 := s$, der Länge $|\{v_0, v_1\}| = l - \varepsilon$ sowie einer zweiten, sehr kurzen Kante $\{v_1, t\}$ der Länge $|\{v_1, t\}| = \varepsilon$. Die Strategie *Greedy* wählt gemäß ihrer Präferenz den Pfad $\pi_G^{\min}(s, t) = (v_0, v_1, t)$, der um ε kürzer ist als die Kante $\{s, t\}$. Am Knoten v_1 stellt sich die Kante $\{v_1, t\}$ als blockiert heraus. Um die Strategie zu möglichst großen relativen Kosten zu zwingen, konstruieren wir einen alternativen Pfad (v_1, v_2, t) . Die Kosten dieses Pfades wählen wir um ε kleiner als die des Pfades (v_1, v_0, t) , um *Greedy* zur Traversierung dieses Weges zu veranlassen. Die beiden Kanten des Pfades (v_1, v_2, t) haben die Längen $|\{v_1, v_2\}| = 2l - 2\varepsilon$ und $|\{v_2, t\}| = \varepsilon$, wobei die Letztere durch einen Gegenspieler blockiert wird. Hierdurch wird *Greedy* am Knoten v_2 zur Berechnung eines neuen kürzesten Pfades nach t veranlasst. Die Konstruktion des Graphen führen wir auf diese Weise analog bis zum Pfad (v_{k-1}, v_k, t) fort. Jeder der Pfade (v_i, v_{i+1}, t) besteht aus zwei Kanten der Länge $|\{v_i, v_{i+1}\}| = 2^i l - 2^i \varepsilon$ und $|\{v_{i+1}, t\}| = \varepsilon$, wobei sich jeweils die Kante $\{v_{i+1}, t\}$ bei Erreichen des Knotens v_{i+1} als blockiert herausstellt. Nach Ankunft am Knoten v_k wurde mit Ausfall der Kante $\{v_k, t\}$ bereits die k -te Blockade eingesetzt, weshalb kein weiterer alternativer Weg mit ausfallender Kante konstruiert werden kann. Die *Greedy*-Strategie traversiert nun den Pfad $\pi_G^{\min}(v_k, t) = (v_k, v_{k-1}, \dots, v_0, t)$ und erreicht ihr Ziel. Insgesamt verursacht die Strategie *Greedy* zum Lösen der Problem Instanz P_k die Kosten $\text{Greedy}(P_k) = 2 \sum_{i=0}^{k-1} |\{v_i, v_{i+1}\}| + |\{s, t\}|$. Demgegenüber stehen optimale Kosten von $\text{OPT}(P_k) = |\{s, t\}|$. Wir können für ein beliebiges $\delta > 0$ ein $\varepsilon > 0$ angeben, so dass gilt:

$$\begin{aligned}
\forall k \in \mathbb{N} \exists P_k \in \Pi : \frac{\text{Greedy}(P_k)}{\text{OPT}(P_k)} &= \frac{2 \sum_{i=0}^{k-1} |\{v_i, v_{i+1}\}| + |\{s, t\}|}{|\{s, t\}|} \\
&= \frac{2 \sum_{i=0}^{k-1} (2^i l - 2^i \varepsilon) + (l + \varepsilon)}{l + \varepsilon} \\
&\geq 2 \sum_{i=0}^{k-1} 2^i + 1 - \delta \\
&= 2^{k+1} - 1 - \delta.
\end{aligned} \tag{*}$$

Ungleichung (*) ist nicht für beliebiges ε erfüllt. Jedoch konvergiert der zuvor angegebene Bruch für $\varepsilon \searrow 0$ von unten gegen $2 \sum_{i=0}^{k-1} 2^i + 1$, weshalb für jedes beliebige $\delta > 0$ der Wert ε klein genug gewählt werden kann, dass Ungleichung (*) erfüllt ist. Da l beliebig groß gewählt werden kann, gilt diese Ungleichung auch mit einem beliebig großen Wert für $\text{OPT}(P_k)$, womit nach Lemma 3.1 eine untere Schranke von $2^{k+1} - 1 - \delta$ für die Strategie *Greedy* auf beliebigen Graphen gezeigt ist. Da δ beliebig nah an 0 gewählt werden kann, gilt sogar eine untere Schranke von $2^{k+1} - 1$. Um zu zeigen, dass diese Schranke für die Strategie *Greedy* auf allgemeinen Graphen scharf ist, müssen wir beweisen, dass kein

Szenario existiert, das einen höheren kompetitiven Faktor erzwingen kann.

Satz 3.3. *Die Strategie Greedy ist auf allgemeinen Graphen $(2^{k+1}-1)$ -kompetitiv und nicht c -kompetitiv für alle $c < 2^{k+1} - 1$.*

Beweis. Es sei Π das k -Canadian Traveller Problem auf allgemeinen Graphen. Eine beliebige Probleminstanz $P \in \Pi$ besteht aus einem ungerichteten, gewichteten Graphen $G = (V, E)$, einer Konstanten k sowie den Start- und Zielknoten $s, t \in V$. Mit G_i bezeichnen wir den Graphen, in dem die ersten $i \in \{1, \dots, k\}$ ausfallenden Kanten entfernt wurden. Dann lässt sich für alle $P \in \Pi$ folgende Argumentation anwenden: Es sei mit $l := d_{G_k}(s, t)$, für $l > 0$, die Länge des kürzesten Pfades von s nach t in G_k gegeben. Nun ist der zurückgelegte Weg der Strategie *Greedy* vom Startknoten s zu einem Knoten v_1 in G , an dem die erste blockierte Kante e_1 gefunden wird, beschränkt durch $d_G(s, v_1) \leq l$, da sonst der Zielknoten t bereits vorher erreicht worden wäre. Die größtmögliche Distanz vom Knoten v_1 zum Knoten v_2 ist daher beschränkt durch $d_{G_1}(v_1, v_2) \leq d_{G_1}(v_1, t) \leq 2l$. Diese Distanz kann die Strategie in $G_1 := (V, E \setminus \{e_1\})$ maximal zurücklegen, bis sie auf eine weitere blockierte Kante e_2 trifft. Induktiv ist dann der maximal zurückgelegte Weg der Strategie *Greedy* vom Knoten v_i zum Knoten v_{i+1} im Graphen G_i beschränkt durch $d_{G_i}(v_i, v_{i+1}) \leq (2^i)l$. Dann ergibt sich für den maximal zurückgelegten Weg von s zu v_k bei k blockierten Kanten eine Gesamtdistanz $\leq \sum_{i=0}^{k-1} (2^i)l$. Die Kosten $d_{G_k}(v_k, t)$ sind nun beschränkt durch $\sum_{i=0}^{k-1} (2^i)l + l$. Diese obere Schranke ergibt sich aus dem von *Greedy* zwischen s und v_k zurückgelegten Weg und den Kosten l , des nach Annahme kürzesten Pfades von s nach t in G_k . Die Strategie *Greedy* erreicht demnach spätestens nach einer Weglänge von $2 \sum_{i=0}^{k-1} (2^i)l + l$ ihr Ziel. Als obere Schranke für den kompetitiven Faktor erhalten wir schließlich:

$$\begin{aligned} \forall P \in \Pi : \frac{\text{Greedy}(P)}{\text{OPT}(P)} &\leq \frac{2 \sum_{i=0}^{k-1} (2^i)l + l}{l} \\ &= 2 \sum_{i=0}^{k-1} 2^i + 1 \\ &= 2(2^k - 1) + 1 \\ &= 2^{k+1} - 1. \end{aligned}$$

□

Auch für die *Greedy*-Strategie werden wir in den späteren Abschnitten dieses Kapitels untersuchen, inwieweit sich die obere Schranke herabsetzen lässt, wenn Einschränkungen auf die Graphklasse gemacht werden.

3.3 Backtrack und Greedy auf speziellen Graphklassen

Wie wir im vorherigen Abschnitt erfahren haben, erhalten wir für die Strategien *Backtrack* und *Greedy* auf allgemeinen Graphen keine konstanten kompetitiven Faktoren. Für *Backtrack* konnten wir einen kompetitiven Faktor von $2k + 1$ und für *Greedy* einen kompetitiven Faktor von $2^{k+1} - 1$ nachweisen. Dabei ist $k \in \mathbb{N}$ eine obere Schranke für die Anzahl blockierter Kanten des zugrunde liegenden Graphen. In diesem Abschnitt werden wir der Frage nachgehen, ob sich die bisherigen Ergebnisse dadurch verbessern lassen, dass wir die Strategien *Backtrack* und *Greedy* auf spezielle Graphklassen anwenden. Wir untersuchen, ob die durch bestimmte Graphklassen definierten Einschränkungen für mögliche Gegenspieler-Strategien Auswirkungen auf die kompetitiven Faktoren von *Backtrack* und *Greedy* haben. Es sei darauf hingewiesen, dass es zum Erhalt der Lösbarkeit des k -Canadian Traveller Problem unerlässlich ist, dass die zugrunde liegenden Graphen Zyklen enthalten. Wird aus einem zyklensfreien Graphen eine Kante des kürzesten Pfades vom Startknoten s zum Zielknoten t entfernt, so wird das Problem unlösbar, da auf Grund der Zyklensfreiheit keine Ausweichstrecken existieren.

3.3.1 Vollständiger Graph

Im Folgenden zeigen wir, dass sich die Ergebnisse von allgemeinen Graphen auf vollständige Graphen übertragen lassen. Wir betrachten einen ungerichteten, gewichteten vollständigen Graphen $G = (V, E)$ und eine Konstante k , die die Anzahl an maximal ausfallenden Kanten darstellt.

Definition 3.4. Ein *vollständiger Graph* K_n ist ein ungerichteter Graph $G = (V, E)$ ohne Schlingen und Mehrfachkanten mit $|V| = n$ und $|E| = \binom{n}{2} = \frac{n(n-1)}{2}$, für $n > 1$.

Wir konstruieren die Szenarien exakt wie beim allgemeinen Graphen. Die Kanten, die fehlen, um einen vollständigen Graphen zu erzeugen, werden hinzugefügt und mit ∞ gewichtet. Durch diese Konstruktion werden weder *Backtrack*, noch *Greedy* diese Kanten traversieren und es gelten mit den Ergebnissen der oberen Schranken vom allgemeinen Graphen für die beiden Strategien folgende Korollare:

Korollar 3.5. Die Strategie *Backtrack* ist auf vollständigen Graphen $(2k + 1)$ -kompetitiv und nicht c -kompetitiv für alle $c < 2k + 1$.

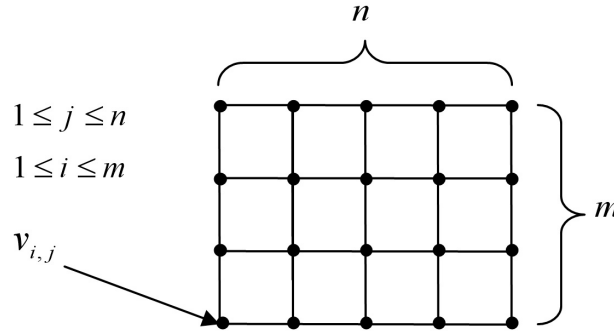
Korollar 3.6. Die Strategie *Greedy* ist auf vollständigen Graphen $(2^{k+1} - 1)$ -kompetitiv und nicht c -kompetitiv für alle $c < 2^{k+1} - 1$.

3.3.2 Vollständiger (m, n) -Gittergraph

Im folgenden Abschnitt werden wir untersuchen, wie sich die Strategien *Backtrack* und *Greedy* auf vollständigen (m, n) -Gittergraphen verhalten.

Definition 3.7. Ein **vollständiger (m,n) -Gittergraph** ist ein ungerichteter Graph $G = (V, E)$ von mn Knoten in der Ebene mit $V = \{v_{i,j} \mid 1 \leq i \leq m, 1 \leq j \leq n, i, j \in \mathbb{N}\}$, bei dem Kanten der Länge $l > 0$ zwischen $v_{i,j}$ und $v_{i',j'}$ genau dann existieren, wenn $i = i'$ und $|j - j'| = 1$ oder $j = j'$ und $|i - i'| = 1$.

Abbildungung 3.3 zeigt ein Beispiel eines vollständigen (m,n) -Gittergraphen. Strukturell er-



Abbildungung 3.3: Beispiel eines vollständigen (m,n) -Gittergraphen

geben sich für die zugrunde liegenden Graphen zwei wesentliche Unterschiede. Zum einen haben alle Kanten die selbe Länge und zum anderen garantiert die Vollständigkeit der Gitterstruktur ein gewisses Reservoir an Ausweichmöglichkeiten, sollte eine Kante des kürzesten Pfades vom Startknoten s zum Zielknoten t gesperrt sein. Es stellt sich nun die Frage, inwieweit unsere beiden Strategien Nutzen daraus ziehen können, d.h. die von allgemeinen Graphen bekannten kompetitiven Faktoren verbessern können.

Die Strategie Backtrack

Erinnern wir uns noch einmal an das worst-case-Beispiel der Strategie *Backtrack* auf allgemeinen Graphen (vgl. Abbildung 3.1). Dieses Beispiel ist nicht ohne Weiteres auf Gittergraphen übertragbar, da zu Konstruktionszwecken unterschiedliche Kantenlängen verwendet wurden. Als geeignetes Mittel zur Abschätzung des kompetitiven Faktors erwies sich das Einführen von Kanten der Länge $l + i\varepsilon$, mit $l \geq 1$ und $1 \leq i \leq k + 1$. Dabei war k eine obere Schranke für die Anzahl möglicher Blockaden. Hierdurch konnte ein Gegenspieler die Strategie zu einem maximalen relativen Umweg verglichen mit der optimalen Lösung zwingen und der dabei auftretende kompetitive Faktor konnte auf Grund der für $\varepsilon \searrow 0$ wegfallenden ε -Terme leicht abgeschätzt werden.

Wie in Abbildung 3.4 dargestellt, ist es durch eine einfache Konstruktion möglich, auch in einem Gittergraphen $G = (V, E)$ bestimmte Längenverhältnisse von Kanten zueinander zu erzeugen. Man stelle sich einen Pfad von Knoten $u \in V$ nach Knoten $v \in V$ aus $\frac{1}{\varepsilon} + 1$ Kanten der Länge ε vor, wobei $\frac{1}{\varepsilon} \in \mathbb{N}_{>0}$ ist. Für das Längenverhältnis der ersten $\frac{1}{\varepsilon}$ Kanten

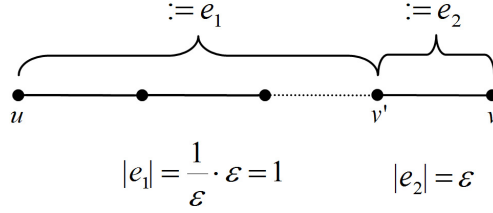


Abbildung 3.4: Längenverhältnisse bestimmter Kanten zueinander

zur $(\frac{1}{\varepsilon} + 1)$ -ten Kante des Pfades von u nach v gilt dann

$$\frac{|\pi(u, v')|}{|\pi(v', v)|} = \frac{\frac{1}{\varepsilon} \cdot \varepsilon}{\varepsilon} = \frac{1}{\varepsilon},$$

wobei v' der direkte Vorgängerknoten von v auf dem Pfad von u nach v ist. Wenn es für die betrachtete Probleminstanz unerheblich ist, ob die Knoten des Pfades von u nach v zusätzliche Verzweigungen haben, ist es ohne Einschränkung möglich, für diesen Pfad anzunehmen, er bestünde aus nur zwei Kanten - einer Kante $e_1 := \pi(u, v')$, mit $|e_1| = \frac{1}{\varepsilon} \cdot \varepsilon = 1$, sowie einer Kante $e_2 := \pi(v', v) = \{v', v\}$, mit $|e_2| = \varepsilon$. Solche Verzweigungen des Pfades treten bei vollständigen (m, n) -Gittergraphen an allen Knoten außer eventuell den vier Eckknoten $c_1, \dots, c_4$ des Gitternetzes auf, da für alle $v \in V \setminus \{c_1, \dots, c_4\}$ gilt: $\deg(v) \in \{3, 4\}$, wobei $\deg(v)$ den Grad von Knoten v bezeichnet. Wie wir sehen werden, stellt dies für unser Problem keine Einschränkung dar.

Das heißt, wir können das worst-case-Beispiel der Strategie *Backtrack* für allgemeine Graphen direkt auf vollständige (m, n) -Gittergraphen übertragen. Abbildung 3.5 illustriert die Konstruktion. Sei Π das k -Canadian Traveller Problem auf vollständigen (m, n) -Gittergraphen. Um eine untere Schranke des kompetitiven Faktors für die *Backtrack*-Strategie zum Lösen von Π anzugeben, konstruieren wir für jedes k eine Instanz $P_k \in \Pi$ eines worst-case-Szenarios. Gegeben sei ein gewichteter vollständiger (m, n) -Gittergraph $G = (V, E)$ mit $m = k + 1$ und $n = \frac{1}{\varepsilon} + 2$ sowie einer Knotenmenge V und einer Kantenmenge E mit $|e| = \varepsilon$, für alle $e \in E$. Die $k + 1$ Zeilen von G seien gegeben durch die Knotenfolgen $(v_{1,1}, \dots, v_{1,n-1}, v_{1,n}), \dots, (v_{k+1,1}, \dots, v_{k+1,n-1}, v_{k+1,n})$. Die letzte Kante einer i -ten Zeile bezeichnen wir mit e_i , für $i \in \{1, \dots, k + 1\}$. Des Weiteren enthalte G zwei ausgewiesene Knoten $s := v_{1,1} \in V$ und $t := v_{1,n} \in V$, die Start und Ziel repräsentieren (vgl. Abbildung 3.5). Die Strategie *Backtrack* verfolgt zunächst den Pfad $\pi_G^{\min}(s, t) = (v_{1,1}, \dots, v_{1,n-1}, v_{1,n})$, also den kürzesten Pfad von s nach t in G . Dabei benutzen wir die oben genannte Konstruktion und können somit annehmen, dass $|(v_{1,1}, \dots, v_{1,n-1})| = 1$ und $|\{v_{1,n-1}, v_{1,n}\}| \searrow 0$, für $\varepsilon \searrow 0$. Ein Gegenspieler wird nun die im Verhältnis zum Pfad $v_{1,1}, \dots, v_{1,n-1}$ sehr kurze Kante e_1 blockieren, um den relativen Umweg der Strategie zu maximieren. Wird diese Blockierung an Knoten $v_{1,n-1}$ festgestellt, so verfolgt die Strategie *Backtrack* den bis zu diesem Knoten zurückgelegten Weg in umgekehrter Reihenfolge, bis wieder der Startknoten s erreicht wird. Die zu diesem Zeitpunkt zurückgelegte Strecke hat die Länge $2 \cdot |(v_{1,1}, \dots, v_{1,n-1})| = 2$. Von s aus wird erneut ein kürzester Pfad zum Knoten t berechnet, diesmal jedoch auf dem

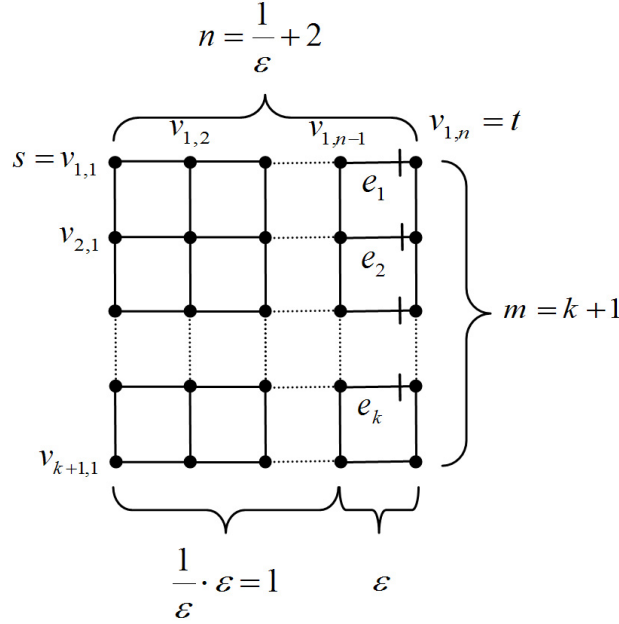


Abbildung 3.5: Gegenspieler-Strategie

Restgraphen $G_1 := (V, E \setminus \{e_1\})$, der aus G durch Entfernen der blockierten Kante e_1 entsteht. Da *Backtrack* keine eindeutige Strategie ist, ist zwar der Pfad $\pi_{G_1}^{\min}(s, t)$ selbst nicht eindeutig, jedoch seine Länge, und es ist unausweichlich, dass der Pfad über die Kante e_2 nach t führt. Da e_2 für die *Backtrack*-Strategie ohnehin die ungünstigste Stelle für eine Blockade ist, entsteht durch die Mehrdeutigkeit des Pfades $\pi_{G_1}^{\min}(s, t)$ für einen Gegenspieler kein Nachteil. Die Strategie *Backtrack* stellt die Blockade erst bei Erreichen des Knotens $v_{2,n-1}$ fest und geht über einen kürzesten Pfad zurück zum Startknoten. Dieser Vorgang wiederholt sich k mal, wobei k die maximale Anzahl blockierter Kanten ist. Pro blockierter Kante entstehen dabei Kosten von ≥ 2 . Die Strategie *Backtrack* findet also erst im $(k+1)$ -ten Schritt den kürzesten freien Weg von s nach t in G . Seien $\text{Backtrack}(P_k)$ die Kosten für *Backtrack* beim Lösen der Probleminstanz P_k , dann gilt für alle $P_k \in \Pi$:

$$\begin{aligned} \text{Backtrack}(P_k) &= 2 \sum_{i=0}^{k-1} (1 + i\varepsilon) + k\varepsilon + 1 + \varepsilon + k\varepsilon \\ &= 2k + 1 + (k^2 + k + 1)\varepsilon. \end{aligned}$$

Seien $\text{OPT}(P_k)$ die Kosten einer optimalen Lösung der Probleminstanz P_k . Der kürzeste *freie* Weg von s nach t in G entspricht dem kürzesten Weg von s nach t in $G_k := (V, E \setminus \{e_1, \dots, e_k\})$, also z. B. $\pi_{G_k}^{\min}(s, t) = (v_{1,1}, v_{2,1}, \dots, v_{k+1,1}, \dots, v_{k+1,n-1}, v_{k+1,n}, v_{k,n}, \dots, v_{1,n})$. Für die Kosten einer optimalen Lösung gilt demnach für alle $P_k \in \Pi$:

$$\begin{aligned} \text{OPT}(P_k) &= k\varepsilon + 1 + \varepsilon + k\varepsilon \\ &= 1 + (2k + 1)\varepsilon. \end{aligned}$$

Für jedes $\delta > 0$ können wir $\varepsilon > 0$ so wählen, dass für die betrachteten Probleminstanzen P_k gilt:

$$\frac{2k+1+(k^2+k+1)\varepsilon}{1+(2k+1)\varepsilon} \geq 2k+1-\delta \quad (*)$$

$$\Rightarrow \frac{\text{Backtrack}(P_k)}{\text{OPT}(P_k)} \geq 2k+1-\delta$$

$$\Rightarrow \text{Backtrack}(P_k) \geq (2k+1-\delta) \cdot \text{OPT}(P_k). \quad (**)$$

Die Ungleichung $(*)$ gilt nicht für beliebiges ε . Weil aber der Bruch auf der linken Seite für $\varepsilon \searrow 0$ gegen $2k+1$ konvergiert, können wir für beliebiges vorgegebenes $\delta > 0$ den Wert ε klein genug wählen, so dass die Ungleichung $(*)$ erfüllt ist. In Ungleichung $(**)$ können wir alle P_k so wählen, dass $\text{OPT}(P_k)$ beliebig groß wird. Dazu skalieren wir einfach alle Kantengewichte mit demselben großen Faktor. Nach Lemma 3.1 ist also bewiesen, dass *Backtrack* auf vollständigen (m, n) -Gittergraphen nicht besser als $(2k+1-\delta)$ -kompetitiv ist. Da wir für δ einen Wert beliebig nah an 0 wählen können, erhalten wir für den kompetitiven Faktor eine untere Schranke von $2k+1$. Da die hier gezeigte untere Schranke mit der oberen Schranke aus Kapitel 3.1.1 übereinstimmt und Gittergraphen ein Spezialfall allgemeiner Graphen sind, erhalten wir:

Korollar 3.8. *Die Strategie Backtrack ist auf vollständigen (m, n) -Gittergraphen $(2k+1)$ -kompetitiv und nicht c -kompetitiv für alle $c < 2k+1$.*

Wir erhalten also für die Strategie *Backtrack* auf vollständigen (m, n) -Gittergraphen denselben kompetitiven Faktor wie auf allgemeinen Graphen. Das überrascht nicht, schließlich konnten wir das worst-case-Beispiel direkt auf Gittergraphen übertragen und von allgemeinen Graphen abweichende Wegkosten fallen asymptotisch betrachtet nicht ins Gewicht. Aus der Tatsache, dass alle Kanten eines vollständigen (m, n) -Gittergraphen die selbe Länge haben, kann die Strategie *Backtrack* keinen Nutzen ziehen. Auch die Vollständigkeit der Gitterstruktur und die daraus resultierende Vielzahl an Ausweichmöglichkeiten im Falle von Kantenausfällen wird durch die Strategie nicht ausgenutzt. Das Problem ist, dass *Backtrack* immer zum Startpunkt s zurückkehrt, sobald eine blockierte Kante auf den Pfad $\pi_{G_i}^{\min}(s, t)$, mit $0 \leq i \leq k-1$, festgestellt wird. Die Möglichkeit, sich dem Zielknoten t bereits ein Stück genähert zu haben, wird von der Strategie außer Acht gelassen, obwohl bei vollständigen (m, n) -Gittergraphen an jedem Knoten gute Chancen bestehen, einen alternativen Pfad nach t zu finden, der nicht den Rückweg über s beinhaltet und daher kürzer sein könnte.

Die Strategie Greedy

Wir betrachten wieder vollständige (m, n) -Gittergraphen $G = (V, E)$, mit $m, n \in \mathbb{N}_{>1}$ sowie Start- und Zielknoten $s, t \in V$. Rein intuitiv liegt die Vermutung nahe, dass sich für die Strategie *Greedy* auf vollständigen (m, n) -Gittergraphen der von allgemeinen Graphen bekannte kompetitive Faktor von $2^{k+1}-1$ verbessert. Schließlich berechnet *Greedy* direkt nach

Auffinden einer blockierten Kante einen neuen kürzesten Weg zum Zielknoten, ohne vorher zu s zurückzukehren. Dieses strategische Verhalten unterstützt ein vollständiger (m, n) -Gittergraph dadurch, dass alle Knoten, außer den vier Eckknoten $c_1, \dots, c_4$ des Gitternetzes, einen Knotengrad ≥ 3 haben, also für alle $v \in V \setminus \{c_1, \dots, c_4\}$ $\deg(v) \in \{3, 4\}$ gilt, wobei $\deg(v)$ den Grad von Knoten v bezeichnet. Ein Knotengrad ≥ 3 bedeutet für die *Greedy*-Strategie, dass bei Erreichen eines Knotens v stets mindestens eine Ausweichmöglichkeit besteht, falls v inzident zu einer blockierten Kante des zu diesem Zeitpunkt kürzesten bekannten Pfades von s nach t ist und keine weitere von v ausgehende Kante blockiert ist. Dabei ist der Pfad, über den v betreten wurde, hier nicht berücksichtigt. Die Chancen stehen also nicht schlecht, dass die *Greedy*-Strategie in einer solchen Situation Nutzen aus der Tatsache ziehen kann, sich möglicherweise dem Ziel bereits ein Stück genähert zu haben. Entgegenkommen könnte der Strategie *Greedy* auch die zweite strukturelle Besonderheit von vollständigen (m, n) -Gittergraphen: Für alle $e_1, e_2 \in E$ ist $|e_1| = |e_2|$, d.h. alle Kanten des Graphen haben die gleiche Länge. Diese Eigenschaft könnte einem Gegenspieler Schwierigkeiten beim Maximieren der Umwege bereiten. Bei *Backtrack* war es möglich, vollständige (m, n) -Gittergraphen zu konstruieren, bei denen unter Berücksichtigung der verwendeten Strategie beliebige Längenverhältnisse für bestimmte Kanten zueinander angenommen werden konnten (vgl. Abbildung 3.4). Bei *Greedy* ist das nicht mehr möglich, da diese Strategie von einem Knoten v , an dem sich der aktuelle Pfad als blockiert herausstellt, stets direkt einen neuen kürzesten Weg zum Zielknoten berechnet und nicht, wie die *Backtrack*-Strategie, einen festen Referenzpunkt ansteuert. *Greedy* versucht zur Berechnung einer alternativen Route eventuelle Abzweigungen der Knoten des blockierten Pfades zu nutzen, die, wie bereits gesehen, an fast allen Knoten existieren.

Im Folgenden beschreiben wir ein Szenario, in dem ein Gegenspieler die *Greedy*-Strategie auf vollständigen (m, n) -Gittergraphen zu großen relativen Umwegen zwingt. Das Szenario ist so angelegt, dass kürzeste Wege zwischen zwei Knoten nicht eindeutig sind. Damit ein Gegenspieler das Verhalten der *Greedy*-Strategie voraussagen und somit große Umwege erzwingen kann, legen wir eine Regel fest, nach der die Strategie kürzeste Wege verfolgt, falls diese nicht eindeutig sind. Wir benutzen an dieser Stelle die Linke-Hand-Regel, um *Greedy* zu einer *eindeutigen* Strategie zu machen. Diese aus der Roboterbewegungsplanung stammende Regel besagt, dass in jedem Explorations-Schritt entlang eines kürzesten Pfades ein Schritt nach links einem Schritt geradeaus und ein Geradeauschritt einem Schritt nach rechts bevorzugt wird. Geradeaus bezeichne dabei die aktuelle Explorationsrichtung. Diese können wir uns anschaulich als Blickrichtung oder Richtung des letzten Schrittes vorstellen. Initial sei die Explorationsrichtung so gewählt, dass sie einem möglichen kürzesten Pfad vom aktuellen Standpunkt zum Zielknoten entspricht. Ein Beispiel zeigt Abbildung 3.6. Dort existieren zwei kürzeste Pfade $\pi_1^{\min}(s, t) := (s, v_1, t)$ und $\pi_2^{\min}(s, t) := (s, v_2, t)$ von s nach t . Ganz gleich welcher kürzeste Pfad der initialen Explorationsrichtung entspricht, wird die *Greedy*-Strategie nach der Linke-Hand-Regel dem Pfad $\pi_1^{\min}(s, t)$ folgen. Ist die initiale Blickrichtung von s nach v_1 , so folgt die *Greedy*-Strategie $\pi_1^{\min}(s, t)$, da in Richtung der linken Hand kein alternativer kürzester Pfad zum Knoten t existiert. Ist die initiale

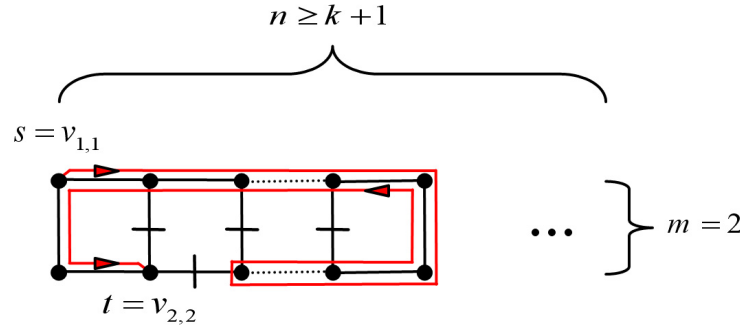


Abbildung 3.8: Gegenspieler-Strategie

in $G_1 = (V, E \setminus \{s_2\})$ zu berechnen. Diesen Pfad blockiert ein Gegenspieler an der Kante s_3 und *Greedy* muss erneut einen kürzesten Pfad berechnen, diesmal von $v_{1,3}$ nach t in G_2 oder kurz $\pi_{G_2}^{\min}(v_{1,3}, t) = (v_{1,3}, v_{1,4}, v_{2,4}, v_{2,3}, v_{2,2})$. Diese Strategie setzt ein Gegenspieler solange fort, bis alle Kanten $s_2, \dots, s_k$ blockiert sind und nur noch eine letzte Blockade durch einen Gegenspieler gesetzt werden darf. Die *Greedy*-Strategie ist dann an Knoten $v_{1,k}$ angekommen und berechnet einen kürzesten Weg $\pi_{G_{k-1}}^{\min}(v_{1,k}, t) = (v_{1,k}, v_{1,k+1}, v_{2,k+1}, \dots, v_{2,2})$ vom aktuellen Standort zu t in G_{k-1} , da $\pi_{G_{k-2}}^{\min}(v_{1,k-1}, t)$ an s_k blockiert ist. Um den Umweg durch die k -te Blockade noch einmal zu maximieren, setzt ein Gegenspieler diese auf Kante $w_{2,2}$. Die *Greedy*-Strategie erfährt davon erst bei Erreichen des Knotens $v_{2,3}$ und muss somit einen letzten kürzesten Pfad $\pi_{G_k}^{\min}(v_{2,3}, t) = (v_{2,3}, \dots, v_{2,k+1}, v_{1,k+1}, \dots, v_{1,1}, v_{2,1}, v_{2,2})$ von $v_{2,3}$ nach t in G_k berechnen. Dieser Pfad führt die *Greedy*-Strategie schließlich zum Ziel, da das Maximum von k Kantenblockaden bereits eingesetzt wurde. Es sei darauf hingewiesen, dass keine der ersten k Berechnungen der kürzesten Wege vom jeweils aktuellen Standort nach t in $G, G_1, \dots, G_{k-1}$ eindeutig ist. Die oben beschriebene Auswahl trifft *Greedy* auf Grund der Linke-Hand-Regel. Dabei gelingt es einem Gegenspieler, die *Greedy*-Strategie mit jeder der ersten $k - 1$ Blockaden weiter vom Zielknoten t zu entfernen. Die k -te Blockade sorgt abschließend noch für eine Verdopplung des durch die ersten $k - 1$ Blockaden erzeugten Umwegs. Seien $\text{Greedy}(P_k)$ die Kosten für den von *Greedy* beim Lösen der Probleminstanz P_k zurückgelegten Weg, dann gilt für alle $P_k \in \Pi$, mit $k \geq 1$:

$$\text{Greedy}(P_k) = 4k.$$

Optimal wäre es, direkt den kürzesten *freien* Weg von s nach t in G zu benutzen. Dies ist der Pfad $\pi_{G_k}^{\min}(s, t) = (v_{1,1}, v_{2,1}, v_{2,2})$. Seien $\text{OPT}(P_k)$ die Kosten dieser optimalen Lösung. Dann gilt für alle $P_k \in \Pi$:

$$\text{OPT}(P_k) = 2.$$

Wir haben also gezeigt, dass für die betrachteten Probleminstanzen P_k , mit $k \geq 1$, gilt:

$$\begin{aligned} \text{Greedy}(P_k) &= 4k \\ &= 2k \cdot 2 \\ &= 2k \cdot \text{OPT}(P_k). \end{aligned} \tag{*}$$

Weil wir hier eine untere Schranke betrachten, ist nicht ausgeschlossen, dass Probleminstanzen existieren, für die in Zeile (*) sogar „ $>$ “ statt „ $=$ “ gilt. Für die von uns betrachteten Instanzen P_k können wir die *Greedy*-Kosten jedoch exakt abschätzen und schreiben demzufolge „ $=$ “ statt „ $\geq$ “. Wir können alle P_k so wählen, dass $\text{OPT}(P_k)$ beliebig groß wird. Dazu skalieren wir einfach alle Kantengewichte mit demselben großen Faktor. Nach Lemma 3.1 ist also bewiesen, dass *Greedy* nicht besser als $2k$ -kompetitiv ist. Damit haben wir eine untere Schranke für den kompetitiven Faktor der *Greedy*-Strategie bei Anwendung der Linke-Hand-Regel auf vollständigen (m, n) -Gittergraphen angegeben. Folgender Satz fasst das Ergebnis noch einmal zusammen:

Satz 3.9. *Die Strategie Greedy ist auf vollständigen (m, n) -Gittergraphen bei Anwendung der Linke-Hand-Regel nicht besser als $2k$ -kompetitiv. Nach Kapitel 3.2.2 ist der kompetitive Faktor höchstens $2^{k+1} - 1$.*

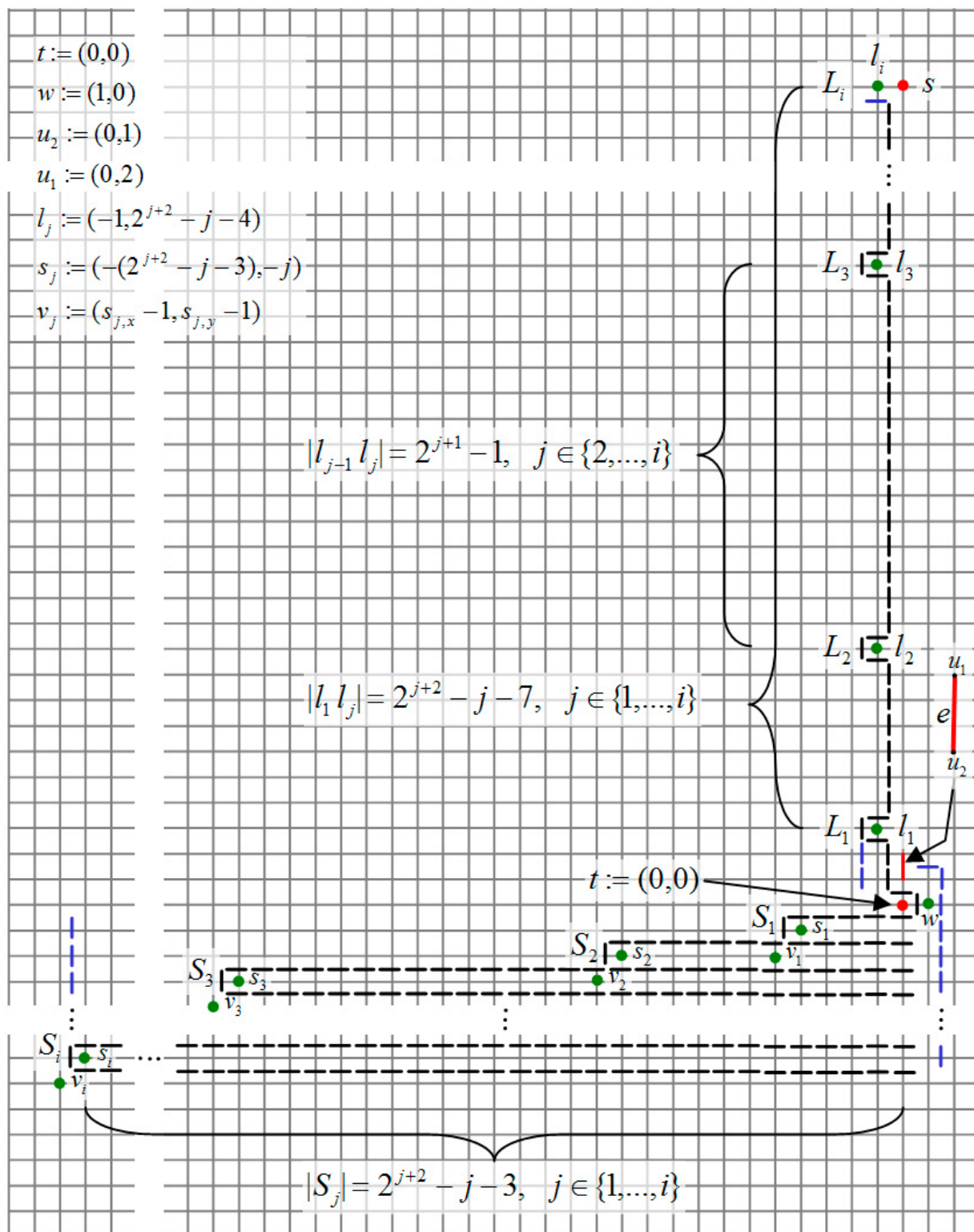
Analog zur hier angewandten Linke-Hand-Regel, wäre es auch möglich, eine Rechte-Hand-Regel zu definieren. Das oben gezeigte Ergebnis würde sich hierdurch nicht verändern, da die Argumentation bis auf Symmetrie identisch wäre. Wichtig ist nur, dass einem Gegenspieler im Vorfeld bekannt ist, welche Regel die *Greedy*-Strategie anwendet, falls kürzeste Wege nicht eindeutig sind.

Obwohl wir nicht wissen, ob *Greedy* auf vollständigen (m, n) -Gittergraphen tatsächlich $2k$ -kompetitiv ist, so liegt die Vermutung doch nahe, dass der kompetitive Faktor von $2^{k+1} - 1$ für *Greedy* auf allgemeinen Graphen deutlich unterschritten wird. Möglicherweise ist die *Greedy*-Strategie im worst-case auf vollständigen (m, n) -Gittergraphen sogar etwas schneller als die *Backtrack*-Strategie, für die wir in Korollar 3.8 bewiesen haben, dass $2k + 1$ der bestmögliche kompetitive Faktor ist.

Bei dem Versuch eine obere Schranke von $2k$ für den kompetitiven Faktor zu zeigen, scheiterten jedoch verschiedene Beweisansätze. Ein vielversprechender Ansatz basierte auf der Annahme, dass *Greedy* auf vollständigen (m, n) -Gittergraphen jede Kante höchstens zweimal betritt. Es stellte sich jedoch heraus, dass folgender Satz gilt:

Satz 3.10. *Auf vollständigen (m, n) -Gittergraphen existieren Szenarien, in denen die Greedy-Strategie eine Kante beliebig oft besucht.*

Beweis. Wir zeigen die Aussage zunächst für eine gerade Anzahl von Kantenbesuchen. Es seien $i \in \mathbb{N}$ eine beliebige Zahl, $G = (V, E)$ ein vollständiger (m, n) -Gittergraph mit $m \geq 2^{i+2} - 3$, $n \geq 2^{i+2} - i$ sowie $s, t \in V$. Für alle $e \in E$ sei $|e| = 1$. Mit $G' := (V, E')$ bezeichnen wir den Graphen, in dem alle Ausfallkanten fehlen. Wir konstruieren das Szenario wie in Abbildung 3.9 dargestellt. Dort sind zudem alle relevanten Knoten sowie deren Koordinaten angegeben, wobei der Knoten t den Ursprung definiert. Die Koordinaten sind an einer späteren Stelle dieses Beweises von größerer Bedeutung, weshalb wir dann noch einmal genauer darauf eingehen werden. Die Farbe einer Blockade (schwarz oder blau)

Abbildung 3.9: $2i$ Besuche der roten Kante

ist zunächst ebenfalls nicht relevant - in beiden Fällen steht die betreffende Kante nicht zur Verfügung. Am Ende des Beweises kommen wir auf die Farbgebung noch einmal zurück. Start- und Zielknoten befinden sich in derselben Spalte von G und haben den Abstand $d_G(s, t) = 2^{i+2} - i - 4$. Auf ihrem Weg von s nach t entdeckt die *Greedy*-Strategie an jedem Knoten eine neue blockierte Kante, es sei denn sie befindet sich an dem Knoten einer Kante, die in eins der insgesamt i vielen Löcher $L_i, \dots, L_1$ führt. Die Knoten in den Löchern bezeichnen wir mit $l_i, \dots, l_1 \in V$. Das Loch L_i bemerkt *Greedy* bereits am Startknoten. Das Loch L_1 bemerkt *Greedy*, nachdem genau $|l_i l_1| = 2^{i+2} - i - 7$ Schritte von s nach t zurückgelegt wurden, denn es gilt:

$$|l_1 l_j| := 2^{j+2} - j - 7, \text{ für alle } j \in \{1, \dots, i\}.$$

Der Abstand zwischen zwei Löchern L_{j-1}, L_j im Graphen G beträgt:

$$|l_{j-1} l_j| := 2^{j+1} - 1, \text{ für alle } j \in \{2, \dots, i\}.$$

Die eingeführten Weglängen lassen sich anhand der Abbildung 3.9 nachvollziehen. Der Pfad $\pi_G^{\min}(s, t)$, dem *Greedy* zu Beginn von s nach t folgt, wird erst an Knoten $u_2 \in V$ unterbrochen, da sich dort die Kante $\{u_2, t\}$ als blockiert herausstellt. Die in Abbildung 3.9 rot eingefärbte Kante $e := \{u_1, u_2\}$ wurde zu diesem Zeitpunkt das erste Mal überquert. Die *Greedy*-Strategie folgt nun dem Pfad $(u_2, \dots, w, t)$ über den Knoten $w \in V$, denn ihr ist zu diesem Zeitpunkt noch nicht bekannt, dass die Kante $\{w, t\}$ ebenfalls blockiert ist. Beginnend in w berechnet *Greedy* nach jedem Schritt einen neuen kürzesten Pfad nach t , da sich dieser mit Entdecken jeder neuen Blockade ändert. Das setzt sich solange fort, bis *Greedy* den Knoten $s_1 \in V$ erreicht und feststellt, dass eine Sackgasse betreten wurde. Wie in Abbildung 3.9 dargestellt, besteht das Szenario aus insgesamt i vielen Sackgassen $S_1, \dots, S_i$ mit jeweils folgender Länge:

$$|S_j| := 2^{j+2} - j - 3, \text{ für alle } j \in \{1, \dots, i\}.$$

Es sei

$$\pi_{S_j}^{\text{in}} := (s_j, \dots, w, \dots, u_2, u_1, \dots, l_j, \dots, t)$$

der von *Greedy* nach Erreichen des Knotens s_j angenommene kürzeste Pfad von s_j nach t über die Kante e und

$$\pi_{S_j}^{\text{out}} := (s_j, \dots, v_j, \dots, t)$$

der von *Greedy* nach Erreichen des Knotens s_j angenommene kürzeste Pfad von s_j nach t , der nicht über die Kante e führt. Entscheidend dafür, dass *Greedy* nach Verlassen der Sackgasse S_1 ein zweites Mal die Kante e passiert, ist also:

$$|\pi_{S_1}^{\text{in}}| < |\pi_{S_1}^{\text{out}}|.$$

Wir werden später zeigen, dass für alle $j \in \{1, \dots, i\}$ gilt:

$$|\pi_{S_j}^{\text{in}}| < |\pi_{S_j}^{\text{out}}|. \quad (*)$$

Aus diesem Grund folgt *Greedy* dem Pfad $\pi_{S_1}^{\text{in}} = (s_1, \dots, w, \dots, u_2, u_1, \dots, l_1, \dots, t)$ nach t und überquert die Kante e ein zweites Mal. Mit Erreichen des Knotens l_1 stellt *Greedy* aber fest, dass der Pfad $\pi_{S_1}^{\text{in}}$ blockiert ist. Es sei

$$\pi_{L_j}^{\text{in}} := (l_j, \dots, u_1, u_2, \dots, w, \dots, v_j, \dots, t)$$

der von *Greedy* nach Erreichen des Knotens l_j angenommene kürzeste Pfad von l_j nach t über die Kante e und

$$\pi_{L_j}^{\text{out}} := (l_j, \dots, l_{j+1}, \dots, t)$$

der von *Greedy* nach Erreichen des Knotens l_j angenommene kürzeste Pfad von l_j nach t , der nicht über die Kante e führt. Entscheidend dafür, dass *Greedy* nach Verlassen des Lochs L_1 ein drittes Mal die Kante e passiert, ist also:

$$|\pi_{L_1}^{\text{in}}| < |\pi_{L_1}^{\text{out}}|.$$

Wir werden gleich zeigen, dass für alle $j \in \{1, \dots, i-1\}$ gilt:

$$|\pi_{L_j}^{\text{in}}| < |\pi_{L_j}^{\text{out}}|. \quad (**)$$

Aus diesem Grund folgt *Greedy* zunächst dem Pfad $\pi_{L_1}^{\text{in}} = (l_1, \dots, u_1, u_2, \dots, w, \dots, v_1, \dots, t)$ nach t und überquert die Kante e ein drittes Mal. Nachdem die *Greedy*-Strategie feststellt, dass dieser Pfad erneut blockiert wird, folgt sie mit gleicher Begründung wie zuvor bereits in S_1 dem Verlauf der Sackgasse S_2 , bis sie den Knoten s_2 erreicht. Auf diese Weise passiert *Greedy* die Kante e genau $2i$ mal, bevor t erreicht wird - einmal auf dem Pfad $\pi_G^{\text{min}}(s, t)$, i mal nach Verlassen der Sackgassen $S_1, \dots, S_i$ sowie $i-1$ mal nach Verlassen der Löcher $L_1, \dots, L_{i-1}$.

Bleibt noch zu zeigen, dass die Ungleichungen (*) und (**) tatsächlich gelten. Für die relevanten Knoten definieren wir zunächst folgende Koordinaten, vergleiche Abbildung 3.9:

- $t := (0, 0)$
- $w := (1, 0)$
- $u_2 := (0, 1)$
- $u_1 := (0, 2)$
- $l_j := (-1, 2^{j+2} - j - 4)$
- $s_j := (-(2^{j+2} - j - 3), -j)$
- $v_j := (s_{j,x} - 1, s_{j,y} - 1)$

Dabei bezeichnen $s_{j,x}$ die x -Koordinate und $s_{j,y}$ die y -Koordinate von Knoten s_j . Analog dazu gibt in den nachfolgenden Umformungen $l_{j,y}$ die y -Koordinate des Knotens l_j an. Den Graphen, in dem der *Greedy*-Strategie alle ausfallenden Kanten, die bis zum Besuch des j -ten Lochs (also bis zum Besuch von Knoten l_j) entdeckt wurden, bekannt sind, bezeichnen wir mit $G_j := (V, E_j)$. Die Korrektheit von $(*)$ begründet sich für $j = 1$ wegen

$$\begin{aligned}
|\pi_{S_1}^{\text{in}}| &\stackrel{!}{<} |\pi_{S_1}^{\text{out}}| \\
&\Leftrightarrow \\
d_{G'}(s_1, w) + d_{G'}(w, l_1) + d_G(l_1, t) &\stackrel{!}{<} d_{G'}(s_1, v_1) + d_G(v_1, t) && \text{vgl. Abb. 3.9} \\
&\Leftrightarrow \\
(|s_{1,x}| + 1 + 1) + (2 + |l_{1,y}|) + (|l_{1,y}| + 1) &\stackrel{!}{<} (2|s_{1,x}| + 4) + (1 + 2 + |s_{1,x}|) && \text{vgl. Abb. 3.9} \\
&\Leftrightarrow \\
2|l_{1,y}| &\stackrel{!}{<} 2|s_{1,x}| + 2 && \text{Umformung} \\
&\Leftrightarrow \\
2(2^{1+2} - 1 - 4) &\stackrel{!}{<} 2(2^{1+2} - 1 - 3) + 2 && \text{Werte einsetzen} \\
&\Leftrightarrow \\
0 &< 4 \quad \checkmark
\end{aligned}$$

und wie folgt für alle $j \in \{2, \dots, i\}$:

$$\begin{aligned}
|\pi_{S_j}^{\text{in}}| &\stackrel{!}{<} |\pi_{S_j}^{\text{out}}| \\
&\Leftrightarrow \\
d_{G'}(s_j, w) + d_{G'}(w, l_j) + d_{G_{j-1}}(l_j, t) &\stackrel{!}{<} d_{G'}(s_j, v_j) + d_G(v_j, t) && \text{vgl. Abb. 3.9} \\
&\Leftrightarrow \\
(|s_{j,x}| + 1 + j) + (2 + |l_{j,y}|) + (|l_{j,y}| + 3) &\stackrel{!}{<} (2|s_{j,x}| + 4) + (j + 2 + |s_{j,x}|) && \text{vgl. Abb. 3.9} \\
&\Leftrightarrow \\
2|l_{j,y}| &\stackrel{!}{<} 2|s_{j,x}| && \text{Umformung} \\
&\Leftrightarrow \\
2(2^{j+2} - j - 4) &\stackrel{!}{<} 2(2^{j+2} - j - 3) && \text{Werte einsetzen} \\
&\Leftrightarrow \\
0 &< 2. \quad \checkmark
\end{aligned}$$

Die Korrektheit von $(**)$ lässt sich für alle $j \in \{1, \dots, i-1\}$ folgendermaßen belegen:

$$\begin{aligned}
|\pi_{L_j}^{\text{in}}| &\stackrel{!}{<} |\pi_{L_j}^{\text{out}}| \\
&\Leftrightarrow \\
d_{G'}(l_j, w) + d_{G'}(w, v_j) + d_G(v_j, t) &\stackrel{!}{<} d_{G'}(l_j, l_{j+1}) + d_{G_j}(l_{j+1}, t) \quad \text{vgl. Abb. 3.9} \\
&\Leftrightarrow \\
(|l_{j,y}| + 2) + (j + 3 + |s_{j,x}|) + (j + 2 + |s_{j,x}|) &\stackrel{!}{<} (2 + |l_{j+1,y}| - |l_{j,y}|) \\
&\quad + (|l_{j+1,y}| + 3) \quad \text{vgl. Abb. 3.9} \\
&\Leftrightarrow \\
2|s_{j,x}| + 2j + 2 &\stackrel{!}{<} 2|l_{j+1,y}| - 2|l_{j,y}| \quad \text{Umformung} \\
&\Leftrightarrow \\
2(2^{j+2} - j - 3) + 2j + 2 &\stackrel{!}{<} 2(2^{j+3} - (j+1) - 4) \\
&\quad - 2(2^{j+2} - j - 4) \quad \text{Werte einsetzen} \\
&\Leftrightarrow \\
0 &< 2. \quad \checkmark
\end{aligned}$$

Somit können wir für alle $i \in \mathbb{N}$ ein Szenario angeben, in dem die *Greedy*-Strategie eine Kante $2i$ mal passiert.

Um den Beweis von Satz 3.10 zu vervollständigen, müssen wir noch zeigen, dass sich ebenso für alle $i \in \mathbb{N}$ ein Szenario angeben lässt, in dem die *Greedy*-Strategie eine Kante $2i - 1$ mal passiert. Das erreichen wir, indem wir in dem Szenario aus Abbildung 3.9 den letzten Sackgassenkorridor S_i öffnen. In dem daraus resultierenden Szenario ist die Anzahl der Kantenbesuche von e genau $2i - 1$, also ungerade.

Insgesamt ist damit gezeigt, dass wir stets ein Szenario angeben können, in dem die *Greedy*-Strategie eine Kante beliebig oft besucht - unabhängig davon, ob die Anzahl der Kantenbesuche gerade oder ungerade ist. Daraus folgt die Korrektheit von Satz 3.10. Die angegebene Konstruktion bedarf allerdings exponentiell vieler Blockaden.

Abschließend sei erwähnt, dass die in Abbildung 3.9 blau dargestellten Blockaden ausschließlich verwendet werden, damit alle von der *Greedy*-Strategie zurückgelegten Wege eindeutig sind. Wüsste ein Gegenspieler, dass *Greedy* nach der Linke-Hand-Regel verfährt, falls kürzeste Wege nicht eindeutig sind, so könnte auf alle blauen Blockaden verzichtet werden. Da in dem Szenario durch Verwendung der blauen Blockaden alle von *Greedy* zurückgelegten Pfade stets eindeutig sind, gilt Satz 3.10 unabhängig von der für *Greedy* gewählten Richtungspräferenz im Fall mehrdeutiger kürzester Wege.

□

Zum Abschluss dieses Abschnitts sei noch einmal darauf hingewiesen, dass sich Satz 3.9 nicht auf *Greedy* im Allgemeinen bezieht. Die Aussage gilt nur, wenn *Greedy* um die Linke-Hand-Regel erweitert wird, so dass wir eine eindeutige Strategie erhalten. Dadurch kann ein Gegenspieler jeden Schritt von *Greedy* voraussagen und im Rahmen seiner Möglichkeiten entsprechend darauf reagieren.

Obwohl es den Anschein hat, dass die Linke-Hand-Regel die *Greedy*-Strategie nicht nennenswert einschränkt, fällt es ohne diese Regel deutlich schwerer ein Szenario zu finden, in dem *Greedy* einen ähnlich hohen kompetitiven Faktor hat. Mit Π bezeichnen wir wieder das k -Canadian Traveller Problem auf vollständigen (m, n) -Gittergraphen, wobei k eine obere Schranke für die Anzahl ausfallender Kanten angibt. Um eine untere Schranke des kompetitiven Faktors anzugeben, konstruieren wir wieder für jedes k eine Instanz $P_k \in \Pi$ eines worst-case-Szenarios. Abbildung 3.10 zeigt eine leicht modifizierte Form des Szenarios aus Abbildung 3.8. Ausfallende Kanten sind dort so gewählt, dass von *Greedy* berechne-

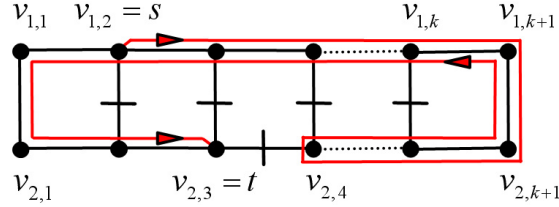


Abbildung 3.10: eindeutige kürzeste Wege der *Greedy*-Strategie

te kürzeste Wege stets eindeutig sind. Es sei $G = (V, E)$ ein gewichteter vollständiger (m, n) -Gittergraph, mit $m = 2$ und $n \geq k + 1 \geq 4$ sowie $|e| = 1$, für alle $e \in E$. Den Knoten der i -ten Zeile und j -ten Spalte von G bezeichnen wir mit $v_{i,j} \in V$. Start und Ziel seien die Knoten $s := v_{1,2}$ und $t := v_{2,3}$. Die *Greedy*-Strategie folgt nun dem Pfad $(s, \dots, v_{1,k+1}, v_{2,k+1}, \dots, v_{2,4}, \dots, v_{2,k+1}, v_{1,k+1}, \dots, v_{1,1}, v_{2,1}, \dots, t)$ und verursacht dabei für alle $P_k \in \Pi$, mit $k \geq 3$, die Kosten:

$$\begin{aligned} \text{Greedy}(P_k) &= 2(k - 1 + 1 + k - 3) + 4 \\ &= 4k - 2. \end{aligned}$$

Optimal wäre hingegen der Pfad $(s, v_{1,1}, v_{2,1}, \dots, t)$, der für alle $P_k \in \Pi$, mit $k \geq 2$, folgende Länge hat:

$$\text{OPT}(P_k) = 4.$$

Wir haben also gezeigt, dass für die betrachteten Probleminstanzen P_k , mit $k \geq 3$, gilt:

$$\begin{aligned} \text{Greedy}(P_k) &= 4k - 2 \\ &= \left(k - \frac{1}{2}\right) \cdot 4 \\ &= \left(k - \frac{1}{2}\right) \cdot \text{OPT}(P_k). \end{aligned}$$

Wir können auch hier wieder durch Skalierung der Kantengewichte alle P_k so wählen, dass $\text{OPT}(P_k)$ beliebig groß wird und haben damit nach Lemma 3.1 bewiesen, dass *Greedy* nicht besser als $(k - \frac{1}{2})$ -kompetitiv ist. Hiermit haben wir eine untere Schranke für den kompetitiven Faktor der *Greedy*-Strategie auf vollständigen (m, n) -Gittergraphen angegeben. Folgender Satz fasst das Ergebnis noch einmal zusammen:

Satz 3.11. *Die Strategie Greedy ist auf vollständigen (m, n) -Gittergraphen für alle $k \geq 3$ nicht besser als $(k - \frac{1}{2})$ -kompetitiv. Nach Kapitel 3.2.2 ist der kompetitive Faktor höchstens $2^{k+1} - 1$.*

Diese Aussage gilt im Unterschied zu Satz 3.9 für *Greedy* im Allgemeinen, also auch dann, wenn ein Gegenspieler nicht weiß, wie *Greedy* verfährt, wenn kürzeste Wege nicht eindeutig sind. Auch wenn Satz 3.11 keine Aussage über eine obere Schranke trifft, ist zu vermuten, dass die Linke-Hand-Regel für *Greedy* möglicherweise eine größere Einschränkung darstellt, als zu Beginn vielleicht angenommen.

3.3.3 Allgemeiner (m, n) -Gittergraph

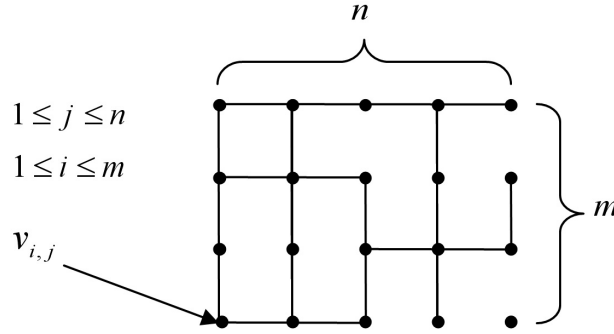
Im vorherigen Abschnitt 3.3.2 haben wir uns mit den Strategien *Backtrack* und *Greedy* auf vollständigen (m, n) -Gittergraphen beschäftigt. Diese Graphklasse zeichnet sich u. a. dadurch aus, dass zwei Knoten genau dann durch eine Kante verbunden sind, wenn sie in derselben Zeile oder derselben Spalte benachbart sind, vergleiche Definition 3.7 sowie Abbildung 3.3. Lockern wir diese Eigenschaft und verlangen nur noch, dass zwei Knoten durch eine Kante verbunden sein *können*, wenn sie in derselben Zeile oder derselben Spalte benachbart sind, so sprechen wir von allgemeinen (m, n) -Gittergraphen.

Definition 3.12. *Ein allgemeiner (m, n) -Gittergraph ist ein ungerichteter Graph $G = (V, E)$ von mn Knoten in der Ebene mit $V = \{v_{i,j} \mid 1 \leq i \leq m, 1 \leq j \leq n, i, j \in \mathbb{N}\}$, bei dem Kanten der Länge $l > 0$ zwischen $v_{i,j}$ und $v_{i',j'}$ existieren können, wenn $i = i'$ und $|j - j'| = 1$ oder $j = j'$ und $|i - i'| = 1$.*

Abbildung 3.11 zeigt ein Beispiel eines allgemeinen (m, n) -Gittergraphen. Von den auf Seite 47 genannten strukturellen Besonderheiten gegenüber allgemeinen Graphen bleibt also nur noch die Eigenschaft, dass alle Kanten die gleiche Länge haben. Was das für die Strategien *Backtrack* und *Greedy* bedeutet, werden wir in diesem Abschnitt beschreiben.

Die Strategie Backtrack

Das Verhalten hinsichtlich des kompetitiven Faktors der Strategie *Backtrack* auf allgemeinen (m, n) -Gittergraphen ist nach Abschnitt 3.3.2 trivial. Da vollständige (m, n) -Gittergraphen einen Spezialfall allgemeiner (m, n) -Gittergraphen darstellen, eignet sich das dort angegebene worst-case-Szenario für die untere Schranke auch uneingeschränkt für die jetzige Problemstellung. Dass kein Szenario einen höheren kompetitiven Faktor erzwingen kann, folgt aus Kapitel 3.1.1. Daraus ergibt sich folgendes Korollar:

Abbildung 3.11: Beispiel eines allgemeinen (m, n) -Gittergraphen

Korollar 3.13. *Die Strategie Backtrack ist auf allgemeinen (m, n) -Gittergraphen $(2k + 1)$ -kompetitiv und nicht c -kompetitiv für alle $c < 2k + 1$.*

Die Strategie Greedy

Wie wir sehen werden, gilt für die *Greedy*-Strategie auf allgemeinen (m, n) -Gittergraphen der gleiche kompetitive Faktor wie auf allgemeinen Graphen. Das in Abbildung 3.12 angegebene worst-case-Szenario ist bis auf einige konstruktionsbedingte Eigenschaften, die auf Grund der Gitterstruktur nötig sind, analog zu dem aus Kapitel 3.2.2 bekannten Szenario.

Sei Π das k -Canadian Traveller Problem auf allgemeinen (m, n) -Gittergraphen. Um eine untere Schranke des kompetitiven Faktors für die *Greedy*-Strategie zum Lösen von Π anzugeben, konstruieren wir für jedes k eine Instanz $P_k \in \Pi$ eines worst-case-Szenarios. Die in Abbildung 3.12 dargestellten Probleminstanzen P_k seien gegeben durch einen gewichteten allgemeinen (m, n) -Gittergraphen $G = (V, E)$ mit $m = 2k$ und $n = \frac{2^{k-2} - 2^{k-1}\varepsilon + \varepsilon}{\varepsilon} + 1$ sowie einer Knotenmenge V und einer Kantenmenge E mit $|e| = \varepsilon > 0$, für alle $e \in E$ und $\frac{1}{\varepsilon} \in \mathbb{N}_{>0}$. Mit $s, t \in V$ bezeichnen wir Start- und Zielknoten. In Abbildung 3.12 besteht jede gepunktete Kante in Wirklichkeit aus vielen einzelnen Kanten der Länge ε , die solange aneinander gereiht werden, bis die an den gepunkteten Kanten angegebenen Längen erreicht sind. Die Konstruktion erfolgt dabei nach dem aus Abbildung 3.4 in Abschnitt 3.3.2 vorgestellten Prinzip. In Abbildung 3.12 müssten die gepunkteten Kanten deshalb im Verhältnis zu den durchgezogenen Kanten sehr viel länger sein. Dann ließe sich die Grafik jedoch nicht mehr übersichtlich darstellen. Ebenfalls fehlen aus Gründen der Übersichtlichkeit alle Knoten $v \in V$ mit Knotengrad $\deg(v) = 0$. Vom Startknoten s aus folgt *Greedy* dem kürzesten Pfad $\pi_G^{\min}(s, t) = (s, \dots, v_1, t)$. Diesen Pfad blockiert ein Gegenspieler an der Kante $\{v_1, t\}$. Die Blockade zwingt die *Greedy*-Strategie dazu, einen neuen kürzesten Pfad $\pi_{G_1}^{\min}(v_1, t) = (v_1, \dots, v_2, v'_2, \dots, t)$ von v_1 nach t in $G_1 := (V, E \setminus \{v_1, t\})$ zu berechnen, der die Länge $|\pi_{G_1}^{\min}(v_1, t)| = 2 + \varepsilon$ hat und somit minimal (nämlich um 2ε) kürzer ist als der Pfad $\pi_{G_1}(v_1, t) = (v_1, \dots, s, u, \dots, t)$. Ein Gegenspieler blockiert daraufhin den Pfad $\pi_{G_1}^{\min}(v_1, t)$ an der Kante $\{v_2, v'_2\}$. An Knoten v_2 ist *Greedy* also wieder

gezwungen, einen neuen kürzesten Pfad $\pi_{G_2}^{\min}(v_2, t) = (v_2, \dots, v_3, v'_3, \dots, t)$ von v_2 nach t in $G_2 := (V, E \setminus \{\{v_1, t\}, \{v_2, v'_2\}\})$ zu berechnen. Auch $\pi_{G_2}^{\min}(v_2, t)$ ist wieder um 2ε kürzer als der alternative Pfad zurück über s nach t . Diese Strategie setzt ein Gegenspieler solange fort, bis alle Kanten $\{v_1, t\}, \{v_2, v'_2\}, \dots, \{v_k, v'_k\}$ blockiert sind und *Greedy* schließlich doch dem Pfad zurück über s zum Knoten t folgen muss. Es sei darauf hingewiesen, dass ein Gegenspieler seine letzte Blockade anstatt auf der Kante $\{v_k, v'_k\}$ auch auf der Kante $\{w, t\}$ platzieren könnte. Dadurch würde sich der von *Greedy* zurückgelegte Weg sogar um $2(k-1)\varepsilon$ verlängern. Da diese Platzierung der Blockade aber für $\varepsilon \searrow 0$ keinen Einfluss auf das Ergebnis hat, verzichten wir der Einfachheit halber darauf. Seien $\text{Greedy}(P_k)$ die Kosten für den von *Greedy* beim Lösen der Problemistanz P_k zurückgelegten Weg, dann gilt für alle $P_k \in \Pi$, mit $k \geq 1$:

$$\text{Greedy}(P_k) = 2[1 + \sum_{i=1}^{k-1} (2^i - 2^{i+1}\varepsilon + 2\varepsilon)] + 1 + 3\varepsilon.$$

Optimal wäre es, direkt den kürzesten *freien* Weg von s nach t in G zu benutzen. Seien $\text{OPT}(P_k)$ die Kosten dieser optimalen Lösung, dann gilt für alle $P_k \in \Pi$, mit $k \geq 1$:

$$\text{OPT}(P_k) = 1 + 3\varepsilon.$$

Für jedes $\delta > 0$ können wir ε so klein wählen, dass für die betrachteten Problemistanzen P_k , mit $k \geq 1$, gilt:

$$\begin{aligned} & \frac{2[1 + \sum_{i=1}^{k-1} (2^i - 2^{i+1}\varepsilon + 2\varepsilon)] + 1 + 3\varepsilon}{1 + 3\varepsilon} \geq 2^{k+1} - 1 - \delta \quad (*) \\ \Rightarrow & \frac{\text{Greedy}(P_k)}{\text{OPT}(P_k)} \geq 2^{k+1} - 1 - \delta \\ \Rightarrow & \text{Greedy}(P_k) \geq (2^{k+1} - 1 - \delta) \cdot \text{OPT}(P_k). \quad (**) \end{aligned}$$

Die Ungleichung $(*)$ gilt nicht für beliebiges ε . Weil aber der Bruch auf der linken Seite für $\varepsilon \searrow 0$ gegen $2^{k+1} - 1$ konvergiert, können wir für beliebiges vorgegebenes $\delta > 0$ den Wert ε klein genug wählen, so dass die Ungleichung $(*)$ erfüllt ist. In Ungleichung $(**)$ können wir alle P_k so wählen, dass $\text{OPT}(P_k)$ beliebig groß wird. Dazu skalieren wir einfach alle Kantengewichte mit demselben großen Faktor. Nach Lemma 3.1 ist also bewiesen, dass *Greedy* auf allgemeinen (m, n) -Gittergraphen nicht besser als $(2^{k+1} - 1 - \delta)$ -kompetitiv ist. Da wir für δ einen Wert beliebig nah an 0 wählen können, erhalten wir für den kompetitiven Faktor eine untere Schranke von $2^{k+1} - 1$. Zusammen mit der oberen Schranke für die *Greedy*-Strategie auf allgemeinen Graphen aus Kapitel 3.2.2 erhalten wir folgendes Korollar:

Korollar 3.14. *Die Strategie Greedy ist auf allgemeinen (m, n) -Gittergraphen $(2^{k+1} - 1)$ -kompetitiv und nicht c -kompetitiv für alle $c < 2^{k+1} - 1$.*

Es überrascht nicht, dass *Greedy* hier hinsichtlich des kompetitiven Faktors im schlimmsten Fall deutlich schlechter abschneidet, als es auf vollständigen (m, n) -Gittergraphen der Fall zu sein scheint. Schließlich ist dort gerade die Vollständigkeit der Gitterstruktur der Grund dafür, dass der *Greedy*-Strategie im Falle eines blockierten kürzesten Pfades ein gewisses Reservoir an Ausweichmöglichkeiten zur Verfügung steht. Auch wenn nicht bekannt ist, ob *Greedy* auf vollständigen (m, n) -Gittergraphen tatsächlich $(k - \frac{1}{2})$ -kompetitiv ist (bzw. $2k$ -kompetitiv bei Anwendung der Linke-Hand-Regel), liegt die Vermutung doch nahe, dass der kompetitive Faktor deutlich unter $2^{k+1} - 1$ liegt.

3.4 Backtrack und Greedy auf Spannern

3.4.1 t -Spanner

In Kapitel 3.2 haben wir die Strategien *Backtrack* und *Greedy* auf allgemeinen Graphen untersucht und gezeigt, dass *Backtrack* $(2k + 1)$ -kompetitiv und *Greedy* $(2^{k+1} - 1)$ -kompetitiv ist. In dem darauf folgenden Kapitel 3.3 sind wir der Frage nachgegangen, ob sich die kompetitiven Faktoren verbessern, wenn wir spezielle strukturelle Anforderungen an die Graphen stellen. Dabei konnten wir für *Backtrack* auf keiner der von uns untersuchten Graphklassen eine Verbesserung feststellen. Für die *Greedy*-Strategie auf vollständigen (m, n) -Gittergraphen konnte für den kompetitiven Faktor zumindest keine höhere untere Schranke als $k - \frac{1}{2}$ (bzw. $2k$ bei Anwendung der Linke-Hand-Regel) angegeben werden. Ob der kompetitive Faktor jedoch tatsächlich $k - \frac{1}{2}$ (bzw. $2k$) ist, ist nicht bekannt. Jetzt wollen wir die beiden Ansätze der Kapitel 3.2 und 3.3 auf gewisse Weise miteinander verknüpfen. Der Graph $G = (V, E)$, den *Backtrack* und *Greedy* zu Beginn ihrer Berechnungen als Eingabe erhalten, sei beliebig. Spezielle Anforderungen stellen wir lediglich an den Graphen $G' = (V, E')$, mit $E' := E \setminus \{e \in E \mid e \text{ ist eine blockierte Kante}\}$. Dabei sei die maximale Anzahl blockierter Kanten durch $k \in \mathbb{N}$ beschränkt. Die blockierten Kanten im Restgraphen G' müssen dabei so gewählt sein, dass G' die folgenden Eigenschaften eines t -Spanners erfüllt:

Definition 3.15. Sei $G = (V, E)$ ein ungerichteter, gewichteter Graph. Ein Teilgraph $G' := (V, E')$ von G heißt **t -Spanner**, wenn für alle $u, v \in V$ gilt:

$$d_{G'}(u, v) \leq t \cdot d_G(u, v).$$

Der kleinste Parameter t , der obige Ungleichung erfüllt, wird als *Stretch-Faktor* bezeichnet.

Durch das Blockieren von Kanten dürfen die entstehenden Umwege also nicht mehr beliebig groß werden. Wir wollen nun der Frage nachgehen, ob diese Einschränkung Einfluss auf das worst-case-Verhalten von *Backtrack* und *Greedy* hat.

Die Strategie Backtrack

In Abbildung 3.13 ist das worst-case-Szenario für *Backtrack* auf allgemeinen Graphen aus Kapitel 3.2.1 in leicht modifizierter Form dargestellt. Der dort abgebildete ungerichtete,

gewichtete Graph sei $G = (V, E)$ mit den Start- und Zielknoten $s, t \in V$. Der einzige

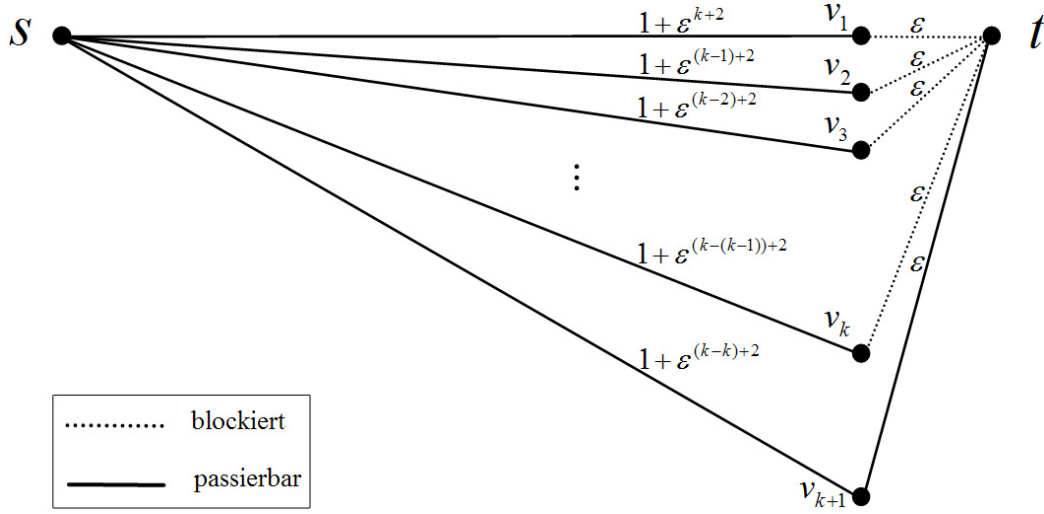


Abbildung 3.13: worst-case-Szenario für *Backtrack* auf allgemeinen Graphen

Unterschied zu dem aus Kapitel 3.2.1 bekannten Szenario ist die abweichende Gewichtung der Kanten $\{s, v_i\} \in E$, für die nun gilt:

$$|\{s, v_i\}| = 1 + \varepsilon^{(k-(i-1))+2}, \text{ für } i \in \{1, \dots, k+1\}.$$

Die neue Gewichtung wurde dabei so vorgenommen, dass folgende beiden Eigenschaften erfüllt bleiben:

1. Es gilt $|\{s, v_i\}| < |\{s, v_{i+1}\}|$, für $i \in \{1, \dots, k\}$ und $0 < \varepsilon < 1$.
2. Es gilt $\lim_{\varepsilon \searrow 0} |\{s, v_i\}| = 1$, für $i \in \{1, \dots, k+1\}$.

Die erste Eigenschaft stellt dabei sicher, dass *Backtrack* erst in der $(k+1)$ -ten Iteration den kürzesten freien Weg von s nach t findet. Die zweite Eigenschaft maximiert den kompetitiven Faktor, also den Quotienten aus den von *Backtrack* verursachten Kosten und den Kosten einer optimalen Lösung. Anhand des Szenarios aus Abbildung 3.13 lässt sich somit ebenfalls nachweisen, dass die *Backtrack*-Strategie auf allgemeinen Graphen nicht besser als $(2k+1)$ -kompetitiv ist. Da die Argumentation nahezu analog zu der aus Kapitel 3.2.1 ist, verzichten wir an dieser Stelle auf die genaue Herleitung. Es sei $k \in \mathbb{N}$ und $G_k = (V, E_k)$, mit $E_k := E \setminus \{\{v_1, t\}, \{v_2, t\}, \dots, \{v_k, t\}\}$, der Restgraph von $G = (V, E)$ aus Abbildung 3.13, in dem k Kanten blockiert wurden. Für alle Knotenpaare $v_i, t \in V$, mit $i \in \{1, \dots, k\}$, gilt:

$$\frac{d_{G_k}(v_i, t)}{d_G(v_i, t)} \rightarrow \infty, \text{ für } \varepsilon \searrow 0.$$

Der Graph G_k erfüllt also für kein endliches t die Eigenschaft, ein t -Spanner von G zu sein. Die gewählte Gewichtung ermöglicht es uns jedoch, diesbezüglich Abhilfe zu schaffen:

Wir erweitern den Graphen G (und damit auch den Graphen G_k) aus Abbildung 3.13 um k Kanten $a_1, \dots, a_k$. Der dadurch entstehende Graph $G' = (V, E')$ ist in Abbildung 3.14 dargestellt. Die rot eingefärbten Kanten $a_i := \{v_i, t\}$, mit $i \in \{1, \dots, k\}$, stellen Alternativen

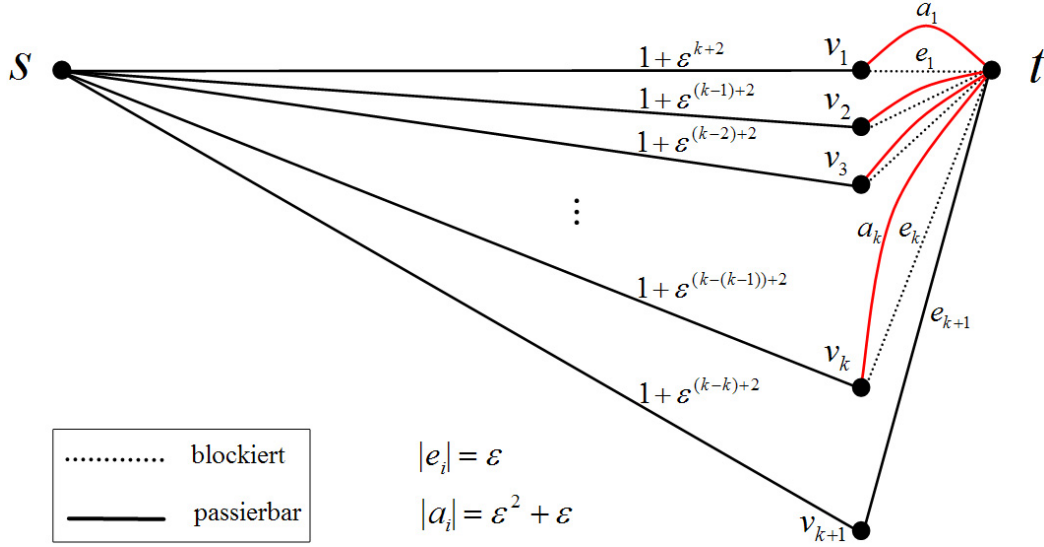


Abbildung 3.14: worst-case-Szenario für *Backtrack* auf t -Spannern

zu den bereits in G existierenden gepunkteten Kanten $\{v_i, t\} =: e_i$ dar. Die neuen Kanten müssen dabei so gewichtet werden, dass sowohl die von *Backtrack* auf dem Weg von s und t zurückgelegte Distanz, als auch der kürzeste freie Weg von s nach t unverändert bleiben. Beides erreichen wir dadurch, dass wir die Kanten $a_1, \dots, a_k$ wie folgt gewichten:

$$|a_i| = \varepsilon^2 + \varepsilon, \text{ für } i \in \{1, \dots, k\}.$$

Es sei $G'_k = (V, E'_k := E_k \setminus \{e_1, \dots, e_k\})$ der Graph, der aus G' durch Entfernen der Kanten $e_1, \dots, e_k$ entsteht. Durch obige Gewichtung der Kanten a_i ist sichergestellt, dass für die Pfade von s nach t in G'_k gilt:

$$|\{s, v_{k+1}\}| + |e_{k+1}| < |\{s, v_1\}| + |a_1| < \dots < |\{s, v_k\}| + |a_k|.$$

Der kürzeste Pfad von s nach t in G'_k , der eine alternative Kante a_i enthält, ist also länger als der kürzeste Vertreter ohne alternative Kante. Bevor die *Backtrack*-Strategie auf ihrem Weg von s nach t den Pfad $(s, v_1) \circ a_1$ in Betracht ziehen würde, hat sie bereits über den kürzeren Pfad $(s, v_{k+1}) \circ e_{k+1}$ den Zielknoten erreicht. Die Erweiterung von G zum Graphen G' hat also weder Einfluss auf die von *Backtrack* zwischen s und t zurückgelegte Distanz, noch auf die Länge des kürzesten freien Pfades zwischen Start und Ziel. Damit ist *Backtrack* für $\varepsilon \searrow 0$ auch auf dem Graphen G' $(2k+1)$ -kompetitiv. Anders als für die Graphen G und G_k gilt jetzt aber für alle Knoten $u, v \in V$:

$$\frac{d_{G'_k}(u, v)}{d_{G'}(u, v)} \leq \frac{d_{G'_k}(v_i, t)}{d_{G'}(v_i, t)} = \frac{\varepsilon^2 + \varepsilon}{\varepsilon} = 1 + \varepsilon \searrow 1, \text{ für } \varepsilon \searrow 0.$$

Der Graph G'_k ist also für $\varepsilon \searrow 0$ ein t -Spanner von G' , wobei $t > 1$ beliebig gewählt werden kann. Das bedeutet, dass ein Gegenspieler, selbst wenn er nur Kanten blockieren darf, so dass der Restgraph ein t -Spanner des Ursprungsgraphen ist, ein Szenario konstruieren kann, für das die *Backtrack*-Strategie nicht besser als $(2k+1)$ -kompetitiv ist. Zusammen mit der oberen Schranke aus Kapitel 3.2.1 erhalten wir das folgende Korollar:

Korollar 3.16. *Werden in einem ungerichteten, gewichteten Graphen G maximal k Kanten so entfernt, dass der Restgraph G' ein t -Spanner von G ist, so ist die Strategie *Backtrack* auf G für alle $t > 1$ $(2k+1)$ -kompetitiv und nicht c -kompetitiv für alle $c < 2k+1$.*

Setzen wir also voraus, dass der Restgraph ein t -Spanner des Ursprungsgraphen ist, verbessert sich der kompetitive Faktor von *Backtrack* für kein $t > 1$. Für $t = 1$ lässt sich allerdings kein worst-case-Szenario finden, in dem *Backtrack* für die hier betrachtete Problemklasse keinesfalls besser als $(2k+1)$ -kompetitiv ist, denn dann existiert für jede ausfallende Kante $e := \{u, v\}$ ein alternativer freier Pfad $\pi(u, v)$ von u nach v mit $|\pi(u, v)| = |e|$. Angenommen $\pi(u, v)$ wäre länger als e , dann wäre die t -Spanner Restriktion verletzt. Andersherum kann $\pi(u, v)$ nicht kürzer als e sein, weil die *Backtrack*-Strategie dann in keinem Fall von der blockierten Kante e zu einem Umweg veranlasst würde. Das Blockieren einer solchen Kante hätte für einen Gegenspieler also keinen Sinn. Wegen $|\pi(u, v)| = |e|$ wären dann die von der *Backtrack*-Strategie berechneten kürzesten Wege nicht mehr eindeutig, und es ließe sich kein Szenario angeben, in dem *Backtrack* keinesfalls besser als $(2k+1)$ -kompetitiv ist. Ohne eine zusätzliche Regel, die das Verhalten von *Backtrack* für den Fall mehrdeutiger kürzester Wege angibt, lässt sich deshalb kein geeignetes Szenario finden. Eine solche Regel ist uns bereits aus Kapitel 3.3.2 bekannt. Dort haben wir die *Greedy*-Strategie für den Fall uneindeutiger kürzester Wege auf die Linke-Hand-Regel festgelegt, um eine eindeutige Strategie zu erhalten. Nach einem ähnlichen Schema wollen wir nun für die *Backtrack*-Strategie eine zusätzliche Regel festlegen. Es handelt sich im Gegensatz zur Linke-Hand-Regel allerdings um eine Festlegung, die *Backtrack* nicht zu einer eindeutigen Strategie macht:

*Existieren in einem Graphen G_i mehrere kürzeste Pfade von s nach t , so bevorzugt *Backtrack* den Pfad, der aus den wenigsten Kanten besteht.* (*)

Dabei bezeichnen wir für jedes $i \in \{1, \dots, k\}$ mit G_i den Graphen, in dem die ersten i blockierten Kanten entfernt sind. Die Regel scheint *Backtrack* nicht nennenswert einzuschränken, reicht jedoch aus, um ein Szenario anzugeben, in dem *Backtrack* selbst für $t = 1$ nicht besser als $(2k+1)$ -kompetitiv ist. Das Szenario ist in Abbildung 3.15 dargestellt. Alle Pfade von s nach v_i , mit $i \in \{1, \dots, k+1\}$, haben die Länge $|(s, \dots, v_i)| = i$ und bestehen aus genau i Kanten. Zwischen den Knoten v_i , mit $i \in \{1, \dots, k\}$, und dem Knoten t verlaufen jeweils zwei Pfade - ein roter Pfad r_i und ein blauer Pfad b_i . Beide haben jeweils die Länge $|r_i| = |b_i| = \varepsilon$, wobei der blaue Pfad aus nur einer Kante besteht. Dadurch ist sichergestellt, dass der angegebene Graph nach Blockieren der blauen Kanten in jedem Fall ein 1-Spanner des Ursprungsgraphen ist. Die Knoten v_{k+1} und t sind durch die Kante $\{v_{k+1}, t\}$ der Länge

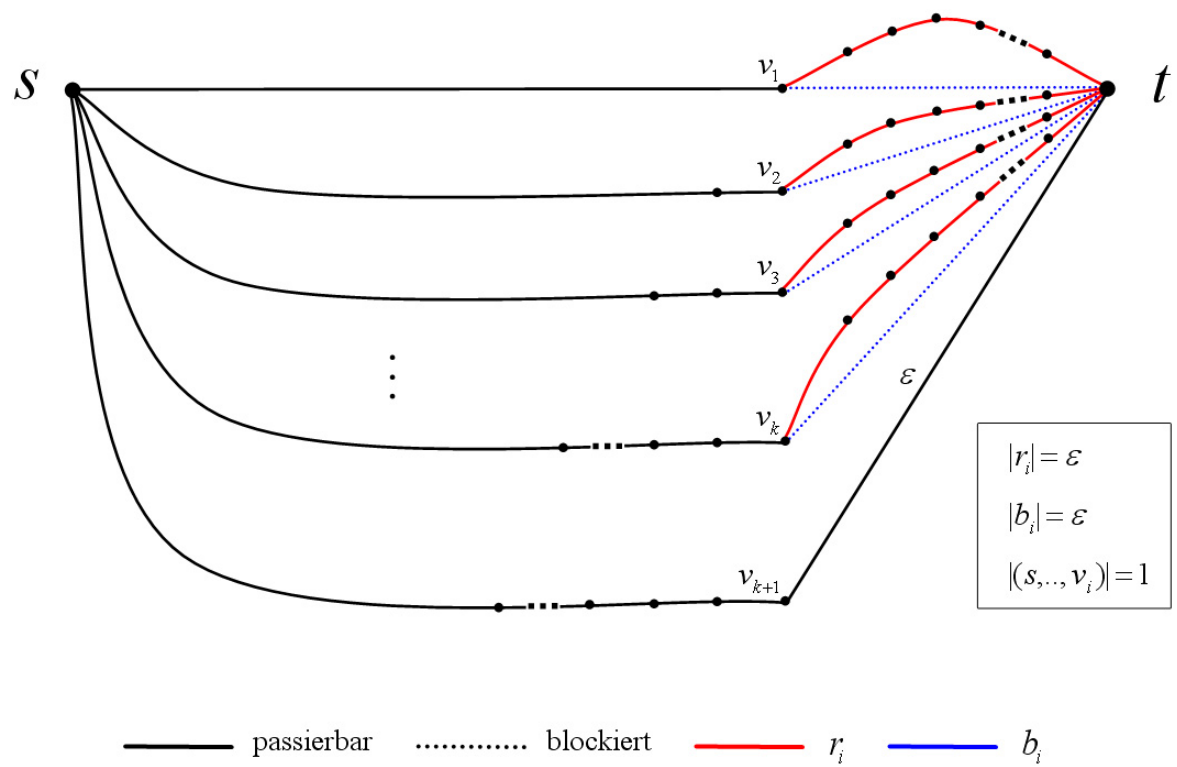


Abbildung 3.15: Präferenz des kürzesten Weges mit minimaler Kantenzahl

ε verbunden. Die Anzahl an Kanten in den roten Pfaden von v_i nach t ist $k + 2$, für alle $i \in \{1, \dots, k\}$. Da alle Pfade von s nach t in G die Länge $1 + \varepsilon$ haben, entscheidet die *Backtrack*-Strategie nach der oben genannten Regel (*), in welcher Reihenfolge die Pfade besucht werden. Demnach bevorzugt *Backtrack* zunächst alle Wege von s nach t , die blaue Teilstücke enthalten. Da sich alle blauen Kanten jedoch als blockiert herausstellen, kehrt *Backtrack* k mal zum Startknoten zurück, bevor schließlich über den Pfad $(s, \dots, v_{k+1}, t)$ das Ziel erreicht wird. Dieser Pfad wird gegenüber jedem anderen freien Pfad bevorzugt, da er unter allen Möglichkeiten aus den wenigsten Kanten besteht - nämlich aus genau i Kanten weniger, als die übrigen Pfade $(s, \dots, v_i, t)$ von s nach t in G' , mit $i \in \{1, \dots, k\}$. Es sei darauf hingewiesen, dass die Existenz des Pfades $(s, \dots, v_{k+1}, t)$ streng genommen nicht nötig wäre, da *Backtrack* eben so gut über den Pfad $(s, \dots, v_1, t)$ ans Ziel gelangen könnte. Um das Szenario aber analog zu den übrigen *Backtrack*-Szenarien zu konstruieren, haben wir diesen Pfad auch diesmal eingefügt. In dem Szenario aus Abbildung 3.15 ist die *Backtrack*-Strategie bei Anwendung oben genannter Regel offensichtlich ebenfalls nicht besser als $(2k + 1)$ -kompetitiv. Die genaue Berechnung dieser unteren Schranke ist nahezu analog zu der aus Kapitel 3.2.1, weshalb wir an dieser Stelle nicht weiter darauf eingehen werden. Zusammen mit der oberen Schranke aus Kapitel 3.2.1 erhalten wir folgendes Korollar:

Korollar 3.17. *Werden in einem ungerichteten, gewichteten Graphen G maximal k Kanten so entfernt, dass der Restgraph G' ein 1-Spanner von G ist, so ist die Strategie Backtrack auf G bei Anwendung der Regel (*) $(2k + 1)$ -kompetitiv und nicht c -kompetitiv für alle $c < 2k + 1$.*

Die Strategie Greedy

Bevor wir zur genauen Analyse der *Greedy*-Strategie auf t -Spannern kommen, möchten wir ein Ergebnis bereits zu Beginn dieses Abschnitts präsentieren:

Satz 3.18. *Werden in einem ungerichteten, gewichteten Graphen G maximal k Kanten so entfernt, dass der Restgraph G' ein 1-Spanner von G ist, so ist die Strategie Greedy auf G optimal (1-kompetitiv).*

Beweis. Es sei $G = (V, E)$ ein ungerichteter, gewichteter Graph und $G' := (V, E')$ der Restgraph von G , der durch Entfernen von maximal k Kanten aus G entsteht. Ist G' ein 1-Spanner von G , so existiert in G' für jede ausfallende Kante $e := \{u, v\}$ ein alternativer freier Pfad $\pi(u, v)$ von u nach v , mit $|\pi(u, v)| \leq t \cdot |e| = |e|$. Wir können annehmen, dass $|\pi(u, v)| = |e|$ gilt, denn andernfalls würde *Greedy* keinesfalls die Kante e überqueren wollen, sondern den kürzeren Pfad $\pi(u, v)$ bevorzugen. Ist e Teil eines von *Greedy* berechneten kürzesten Pfades zum Zielknoten, und wird der Knoten u erreicht, dann berechnet *Greedy* von u einen neuen kürzesten Pfad zum Zielknoten. Wegen $|\pi(u, v)| = |e|$ entsteht für *Greedy* durch die ausfallende Kante e jedoch kein Umweg.

□

Satz 3.18 lässt also schon erahnen, dass eine Beschränkung des Stretch-Faktors des Restgraphen auf *Greedy* einen größeren Einfluss hat, als es bei der *Backtrack*-Strategie der Fall war. Inwieweit ein beliebiger Stretch-Faktor den kompetitiven Faktor von *Greedy* beeinflusst, werden wir im Folgenden untersuchen.

Sei Π das k -Canadian Traveller Problem auf allgemeinen Graphen, in denen ausschließlich Kanten auf die Art entfernt werden, dass die entstehenden Restgraphen t -Spanner ihrer Ursprungsgraphen sind. Um eine untere Schranke des kompetitiven Faktors für die *Greedy*-Strategie zum Lösen von Π anzugeben, konstruieren wir für jedes k eine Instanz $P_k \in \Pi$ eines worst-case-Szenarios. Gegeben sei ein ungerichteter, gewichteter Graph $G = (V, E)$ mit Startknoten $s \in V$ und Zielknoten $t \in V$, wie in Abbildung 3.16 dargestellt. Um

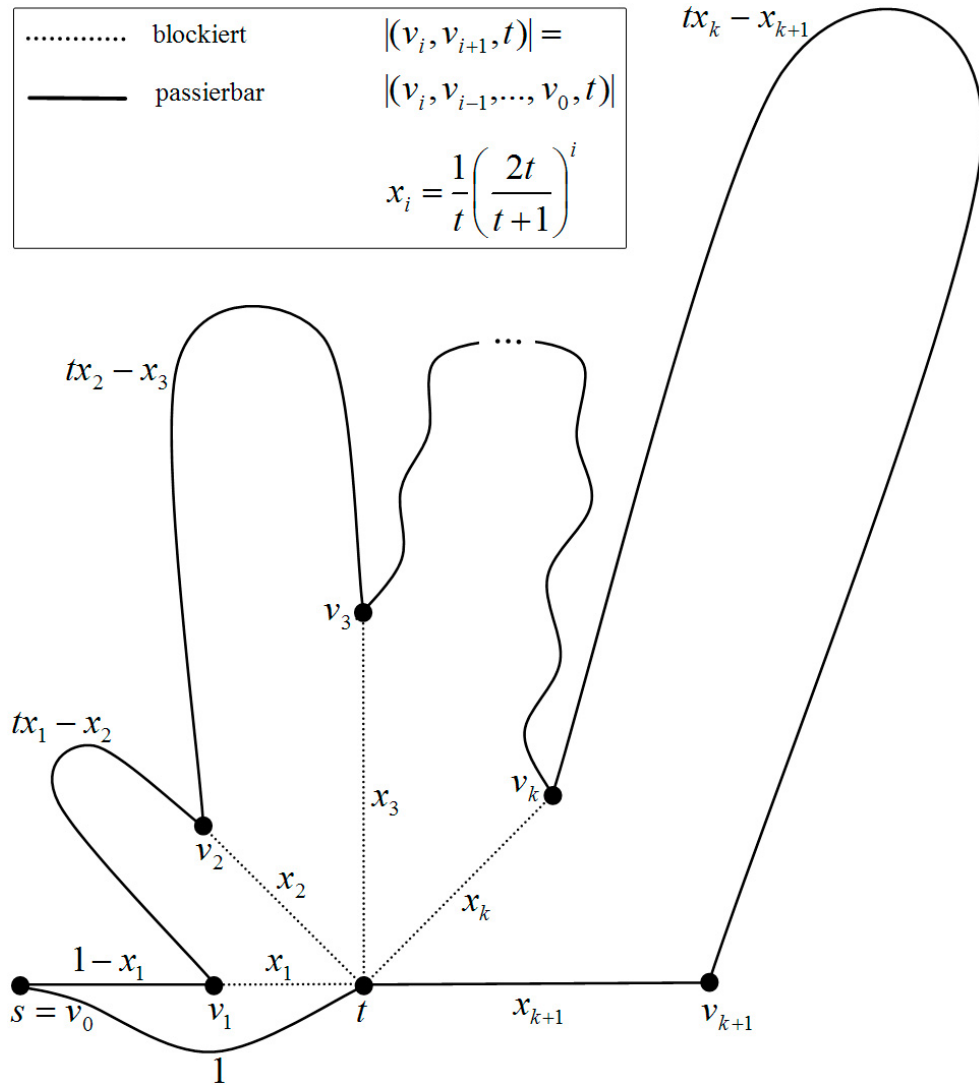


Abbildung 3.16: mehrdeutige kürzeste Wege für die *Greedy*-Strategie

die Notation zu erleichtern, bezeichnen wir den Startknoten alternativ auch mit $v_0 := s$. Der Pfad, dem die *Greedy*-Strategie von s nach t folgt, ist $(s, v_1, v_2, \dots, v_{k+1}, t)$. Für jedes $i \in \{1, \dots, k\}$ bezeichnen wir mit $G_i := (V, E \setminus \{\{v_1, t\}, \dots, \{v_i, t\}\})$ den Graphen, in dem die ersten i blockierten Kanten entfernt sind. Um die Konstruktion des worst-case-Szenarios so übersichtlich wie möglich zu halten, werden wir den Graphen G im ersten Schritt so entwickeln, dass die von der *Greedy*-Strategie berechneten kürzesten Pfade in den Graphen $G, G_1, \dots, G_k$ noch nicht eindeutig sind. Das erleichtert die Konstruktion und kann anschließend durch die Anpassung bestimmter Pfadlängen mit ε -Termen leicht korrigiert werden. Start- und Zielknoten sind durch die Kante $\{s, t\}$ der Länge $|\{s, t\}| = 1$ miteinander verbunden. Von s aus führt ein zweiter Pfad mit den Kosten $|(s, v_1, t)| = 1$ zum Zielknoten. Dieser besteht aus einem blockierten Teilstück der Länge $|\{v_1, t\}| = x_1$ sowie einem freien Teilstück der Länge $|\{s, v_1\}| = 1 - x_1$. Von v_1 aus führt ein weiterer Pfad der Länge $|(v_1, v_2, t)| = tx_1$ nach t . Dabei ist x_1 gerade so gewählt, dass gilt:

$$\begin{aligned} |(v_1, v_2, t)| &= |(v_1, s, t)| \\ \Leftrightarrow tx_1 &= 2 - x_1 \\ \Leftrightarrow x_1 &= \frac{2}{t+1}. \end{aligned}$$

Der Pfad (v_1, v_2, t) besteht ebenfalls aus einem blockierten und einem freien Teilstück. Die blockierte Kante hat diesmal die Länge $|\{v_2, t\}| = x_2$, die freie Kante die Länge $|\{v_1, v_2\}| = tx_1 - x_2$. Auch von v_2 aus führt ein weiterer Pfad nach t . Seine Länge beträgt $|(v_2, v_3, t)| = tx_2$. Dabei ist x_2 analog zu x_1 so gewählt, dass gilt:

$$\begin{aligned} |(v_2, v_3, t)| &= |(v_2, v_1, s, t)| \\ \Leftrightarrow tx_2 &= 2 - x_1 + tx_1 - x_2 \\ \Leftrightarrow tx_2 &= 2 - \frac{2}{t+1} + t \frac{2}{t+1} - x_2 \\ \Leftrightarrow x_2 &= \frac{4t}{(t+1)^2}. \end{aligned}$$

Wieder besteht der Pfad (v_2, v_3, t) aus einem blockierten und einem freien Teilstück. Die Längen betragen diesmal $|\{v_3, t\}| = x_3$ und $|\{v_2, v_3\}| = tx_2 - x_3$. Wie in Abbildung 3.16 dargestellt, wird der Graph für alle blockierten Kanten $\{v_i, t\}$, für $i \in \{1, \dots, k\}$, auf diese Weise aufgebaut. Der Pfad (v_k, v_{k+1}, t) besteht, im Gegensatz zu den Pfaden (v_i, v_{i+1}, t) , mit $i \in \{0, \dots, k-1\}$, ausschließlich aus freien Kanten und hat die Länge $|(v_k, v_{k+1}, t)| = tx_k$. Weil für jede blockierte Kante $\{v_i, t\}$ der Länge $|\{v_i, t\}| = x_i$, mit $i \in \{1, \dots, k\}$, konstruktionsbedingt ein alternativer freier Pfad $(v_i, \dots, v_1, s, t)$ der Länge $|(v_i, \dots, v_1, s, t)| = tx_i$ existiert, ist sichergestellt, dass der Graph G_k auch tatsächlich ein t -Spanner von G ist. Um die von *Greedy* erzeugten Kosten leichter abschätzen zu können, müssen wir eine Möglichkeit finden, x_i zu berechnen, ohne vorher $x_1, \dots, x_{i-1}$ berechnet zu haben. Dazu definieren wir für alle $i \in \{1, \dots, k+1\}$ wie folgt die Kosten w_i und erweitern die bisherige induktive Definition der Kosten x_i :

- $w_i := |(t, s, v_1, \dots, v_i)|$
- $x_i := \{v_i, t\}$

Die Länge des von *Greedy* bis zum Knoten v_i zurückgelegten Weges, zuzüglich der Länge des optimalen Weges von s nach t , bezeichnen wir demnach mit w_i und die Länge der Kanten zwischen v_i und t mit x_i . Mit x_i haben wir bisher die Länge der i -ten von *Greedy* entdeckten blockierten Kante bezeichnet. Wie oben angegeben, wollen wir aber auch die Länge der nicht blockierten Kante $\{v_{k+1}, t\}$ in die Definition einschließen. Dann gilt nach unserer induktiven Definition:

1. $tx_{i+1} = w_i + tx_i - x_{i+1}$, also $x_{i+1} = \frac{w_i + tx_i}{t+1}$
2. $w_i = tx_i$

Beide Zusammenhänge ergeben sich aus der Konstruktion des Szenarios und lassen sich anhand der Abbildung 3.16 nachvollziehen. Setzen wir die zweite Gleichung in die erste Gleichung ein, so erhalten wir:

$$x_{i+1} = \frac{2tx_i}{t+1}.$$

Diese Rekursion können wir mit Hilfe des oben bereits berechneten Startwertes $x_1 = \frac{2}{t+1}$ in folgende geschlossene Formel überführen:

$$x_i = \frac{1}{t} \left(\frac{2t}{t+1} \right)^i. \quad (1)$$

Wegen $w_i = tx_i$ gilt außerdem:

$$w_i = t \cdot \frac{1}{t} \left(\frac{2t}{t+1} \right)^i = \left(\frac{2t}{t+1} \right)^i. \quad (2)$$

Im letzten Schritt der Konstruktion müssen wir dafür sorgen, dass die von *Greedy* in den Graphen $G, G_1, \dots, G_k$ berechneten kürzesten Pfade eindeutig sind. Dazu verkürzen wir, wie in Abbildung 3.17 dargestellt, alle Kanten $\{v_i, v_{i+1}\}$ jeweils um $2^i \varepsilon$, für $i \in \{0, \dots, k\}$. Die *Greedy*-Strategie folgt vom Start aus nun eindeutig dem Pfad $(s, v_1, v_2, \dots, v_{k+1}, t)$ ins Ziel. Dabei wird der zu Beginn in s berechnete kürzeste Weg nach t in G an allen Knoten $v_1, \dots, v_k$ neu berechnet. Denn dort bemerkt die *Greedy*-Strategie, dass der zuvor berechnete kürzeste Weg durch einen Gegenspieler blockiert wurde. Die neuen kürzesten Wege von v_i nach t , für $i \in \{1, \dots, k\}$, werden jeweils auf den aktualisierten Graphen G_i berechnet. Dabei ist $\pi_{G_i}^{\min}(v_i, t) = (v_i, v_{i+1}, t)$ jeweils um ε kürzer als der Pfad $(v_i, v_{i-1}, \dots, s, t)$ zurück über s nach t . Nachdem wir nun eindeutig wissen, wie sich *Greedy* auf dem Graphen aus Abbildung 3.17 verhält, können wir die von *Greedy* verursachten Kosten ermitteln und eine untere Schranke für den kompetitiven Faktor angeben. Seien $\text{Greedy}(P_k)$ die Kosten für

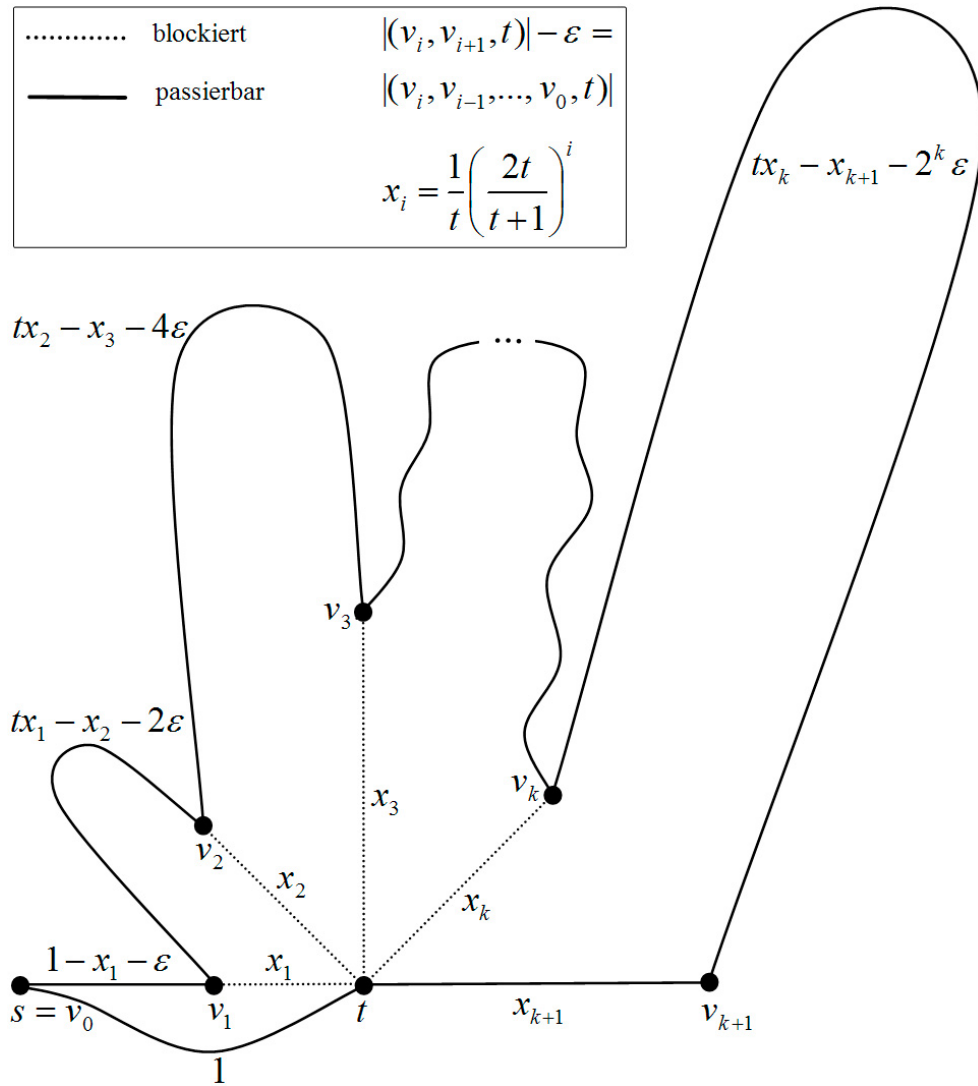


Abbildung 3.17: Konstruktion eindeutiger kürzester Wege durch Korrektur bestimmter Kantenlängen

den von *Greedy* beim Lösen der Problemistanz P_k zurückgelegten Weg, dann gilt für alle $P_k \in \Pi$:

$$\begin{aligned}
\text{Greedy}(P_k) &= w_{k+1} - |\{s, t\}| + x_{k+1} - \sum_{i=0}^k 2^i \varepsilon && \text{nach Def. von } w_i \text{ und } x_i \\
&= w_{k+1} - 1 + x_{k+1} - (2^{k+1} \varepsilon - \varepsilon) && \text{Einsetzen der Werte} \\
&= w_{k+1} - 1 + \frac{1}{t} \left(\frac{2t}{t+1} \right)^{k+1} - (2^{k+1} \varepsilon - \varepsilon) && \text{nach (1)} \\
&= \left(\frac{2t}{t+1} \right)^{k+1} - 1 + \frac{1}{t} \left(\frac{2t}{t+1} \right)^{k+1} - (2^{k+1} \varepsilon - \varepsilon) && \text{nach (2)} \\
&= \frac{t}{t} \left(\frac{2t}{t+1} \right)^{k+1} + \frac{1}{t} \left(\frac{2t}{t+1} \right)^{k+1} - 1 - (2^{k+1} - 1) \varepsilon \\
&= \frac{1+t}{t} \left(\frac{2t}{t+1} \right)^k \left(\frac{2t}{t+1} \right) - 1 - (2^{k+1} - 1) \varepsilon \\
&= 2 \left(\frac{2t}{t+1} \right)^k - 1 - (2^{k+1} - 1) \varepsilon.
\end{aligned}$$

Optimal wäre es, direkt den kürzesten *freien* Weg von s nach t in G mit der Länge

$$\text{OPT}(P_k) = 1$$

zu benutzen. Für jedes $\delta > 0$ können wir $\varepsilon > 0$ so wählen, dass für die betrachteten Problemistanzen P_k gilt:

$$\frac{2 \left(\frac{2t}{t+1} \right)^k - 1 - (2^{k+1} - 1) \varepsilon}{1} \geq 2 \left(\frac{2t}{t+1} \right)^k - 1 - \delta \quad (3)$$

$$\begin{aligned}
&\Rightarrow \frac{\text{Greedy}(P_k)}{\text{OPT}(P_k)} \geq 2 \left(\frac{2t}{t+1} \right)^k - 1 - \delta \\
&\Rightarrow \text{Greedy}(P_k) \geq (2 \left(\frac{2t}{t+1} \right)^k - 1 - \delta) \cdot \text{OPT}(P_k). \quad (4)
\end{aligned}$$

Die Ungleichung (3) gilt nicht für beliebiges ε . Weil aber der Bruch auf der linken Seite für $\varepsilon \searrow 0$ gegen

$$2 \left(\frac{2t}{t+1} \right)^k - 1$$

konvergiert, können wir für beliebiges vorgegebenes $\delta > 0$ den Wert ε klein genug wählen, so dass die Ungleichung (3) erfüllt ist. In Ungleichung (4) können wir alle P_k so wählen, dass $\text{OPT}(P_k)$ beliebig groß wird. Dazu skalieren wir einfach alle Kantengewichte mit demselben großen Faktor. Nach Lemma 3.1 ist also folgendes bewiesen: Werden in einem

ungerichteten, gewichteten Graphen G maximal k Kanten so entfernt, dass der Restgraph G_k ein t -Spanner von G ist, so ist *Greedy* auf G für alle $t \geq 1$ nicht besser als

$$\left(2\left(\frac{2t}{t+1}\right)^k - 1 - \delta\right)\text{-kompetitiv.}$$

Da wir für δ einen Wert beliebig nah an 0 wählen können, erhalten wir für den kompetitiven Faktor eine untere Schranke von

$$2\left(\frac{2t}{t+1}\right)^k - 1.$$

Im Folgenden werden wir zeigen, dass die obere Schranke des kompetitiven Faktors von *Greedy*, für die hier besprochene Problemklasse Π , der unteren Schranke entspricht. Damit erhalten wir folgenden Satz:

Satz 3.19. *Werden in einem ungerichteten, gewichteten Graphen G maximal k Kanten der Art entfernt, dass der Restgraph G' ein t -Spanner von G ist, so ist die Strategie *Greedy* auf G für alle $t \geq 1$ c -kompetitiv mit*

$$c = 2\left(\frac{2t}{t+1}\right)^k - 1$$

und nicht c' -kompetitiv für alle $c' < c$.

Beweis. Nach Definition 1.6 auf Seite 10 handelt es sich bei der *Greedy*-Strategie um eine Sammlung von Graph-Strategien. Wir definieren nun die Graph-Strategie *Greedy'* $\in$ *Greedy* wie folgt:

Definition 3.20. *Greedy' $\in$ Greedy bezeichnet die Graph-Strategie, die im Falle mehrdeutiger kürzester Wege stets einen Weg bevorzugt, der die Gesamtkosten $\text{Greedy}'(P)$ beim Lösen einer Probleminstanz $P \in \Pi$ unter allen in der Strategie *Greedy* enthaltenen Graph-Strategien maximiert und dabei so wenige Kanten wie möglich besucht.*

Umgangssprachlich formuliert könnte man also sagen, *Greedy'* ist ein schlechtester Vertreter aller *Greedy*-Graph-Strategien. Aus Konstruktionsgründen werden wir den Beweis zu Satz 3.19 zunächst für *Greedy'* führen und anschließend auf *Greedy* übertragen.

Sei Π das k -Canadian Traveller Problem auf allgemeinen Graphen, in denen ausschließlich solche Kanten entfernt werden, dass die entstehenden Restgraphen t -Spanner ihrer Ursprungsgraphen sind. Seien $G = (V, E)$ ein ungerichteter, gewichteter Graph, $s, t \in V$ und $\{v_i, v'_i\} \in E$ die in G blockierten Kanten in der Reihenfolge, in der sie jeweils an Knoten $v_i \in V$ von *Greedy'* entdeckt werden. Für jedes $i \in \{1, \dots, k\}$ sei $G_i := (V, E_i)$, mit $E_i := E \setminus \{\{v_1, v'_1\}, \dots, \{v_i, v'_i\}\}$, der Graph, in dem die ersten i blockierten Kanten entfernt sind. Insbesondere ist also $G_0 := G$. Außerdem definieren wir $v_0 := s$. Mit $\text{Greedy}'(P)$ bezeichnen wir die Kosten, die *Greedy'*, angesetzt auf eine Probleminstanz $P \in \Pi$, verursacht, und mit $\text{OPT}(P)$ die Kosten einer optimalen Lösung von P . Die Kosten w_i , x_i und y_i seien wie folgt definiert:

- $w_i := d_{G_k}(s, t) + \overbrace{|\{(v_0, v_1, \dots, v_i)\}|}^{\text{in } G_k}$, für $i \in \{0, \dots, k\}$
- $x_i := d_{G_{i-1}}(v_i, t)$, für $i \in \{1, \dots, k\}$
- $y_i := d_{G_i}(v_i, t)$, für $i \in \{0, \dots, k\}$

Mit w_i bezeichnen wir die Länge des Pfades, den *Greedy'* von s nach v_i zurücklegt, zuzüglich der Länge eines kürzesten Pfades zwischen s und t in G_k . Die Entfernung von Knoten v_i , an dem *Greedy'* die i -te ausfallende Kante entdeckt, nach t in G_{i-1} bezeichnen wir mit x_i . Die Länge des von *Greedy'* an Knoten v_i berechneten kürzesten Weges in G_i nach t nennen wir y_i . Für jede Problem Instanz $P \in \Pi$, in der *Greedy'* auf insgesamt $m \in \{1, \dots, k\}$ ausfallende Kanten trifft, bevor t erreicht wird, gelten die folgenden Bedingungen:

$$\bullet \quad y_i \leq w_i, \text{ für } i \in \{0, \dots, m\} \quad (1)$$

$$\bullet \quad y_i \leq tx_i, \text{ für } i \in \{1, \dots, m\} \quad (2)$$

Bedingung (1) lässt sich durch folgende Ungleichungskette nachvollziehen:

$$\begin{aligned}
 y_i &= d_{G_i}(v_i, t) && \text{nach Definition von } y_i \\
 &\leq d_{G_i}(v_i, s) + d_{G_i}(s, t) && \text{Ein kürzester Weg von } v_i \text{ nach } t \text{ über } s \text{ in } G_i \text{ ist höchstens länger als ein beliebiger kürzester Weg von } v_i \text{ nach } t \text{ in } G_i \text{ (Dreiecksungleichung).} \\
 &\leq \overbrace{|\{(v_i, \dots, v_1, s)\}|}^{\text{in } G_k} + d_{G_i}(s, t) && \text{Der Pfad } (v_i, \dots, v_1, s) \text{ in } G_k \text{ ist höchstens länger als ein kürzester Weg von } v_i \text{ nach } s \text{ in } G_i. \\
 &\leq \overbrace{|\{(v_i, \dots, v_1, s)\}|}^{\text{in } G_k} + d_{G_k}(s, t) && \text{wegen } E_k \subseteq E_i \\
 &= w_i. && \text{nach Definition von } w_i
 \end{aligned}$$

Die Korrektheit von Bedingung (2) lässt sich wie folgt herleiten:

$$\begin{aligned}
 y_i &= d_{G_i}(v_i, t) && \text{nach Definition von } y_i \\
 &\leq d_{G_k}(v_i, t) && \text{wegen } E_k \subseteq E_i \\
 &\leq t \cdot d_G(v_i, t) && G_k \text{ ist } t\text{-Spanner von } G \\
 &\leq t \cdot d_{G_{i-1}}(v_i, t) && \text{wegen } E_{i-1} \subseteq E_i \\
 &= tx_i. && \text{nach Definition von } x_i
 \end{aligned}$$

Der bis hierher beschriebene Aufbau der Problem Instanz P sowie die eingeführten Bezeichnungen gelten für alle Problem Instanzen in Π . Damit jedoch keine Unklarheiten bezüglich der Notation entstehen, werden wir die Bezeichnungen w_i , x_i und y_i an die jeweils betrachtete Problem Instanz anpassen. Beziehen sich die drei Größen beispielsweise auf eine Problem Instanz P' , so ändern wir die Bezeichnungen in w'_i , x'_i und y'_i .

Wir suchen nun eine obere Schranke für die Kosten $\text{Greedy}'(P)$. Die Frage lautet also: Wie muss eine Problem Instanz konstruiert sein, um die Kosten der Graph-Strategie Greedy' bei festen optimalen Lösungskosten zu maximieren? Aufschluss über diese Frage gibt das folgende Lemma, in dem die zu Beginn des Beweises eingeführten Bezeichnungen gelten:

Lemma 3.21. *Es sei $P^{\max} \in \Pi$ eine Problem Instanz, in der Greedy' auf insgesamt $m \in \{1, \dots, k\}$ ausfallende Kanten trifft, bevor t erreicht wird, und es sei $l := \text{OPT}(P^{\max})$. Für alle $i \in \{0, \dots, m\}$ seien die Eigenschaften $y_i^{\max} = w_i^{\max}$ und für alle $i \in \{1, \dots, m\}$ die Eigenschaften $y_i^{\max} = tx_i^{\max}$ erfüllt. Dann gilt für alle $P \in \Pi$ mit $\text{OPT}(P) = l$, in denen Greedy' ebenfalls auf m ausfallende Kanten trifft:*

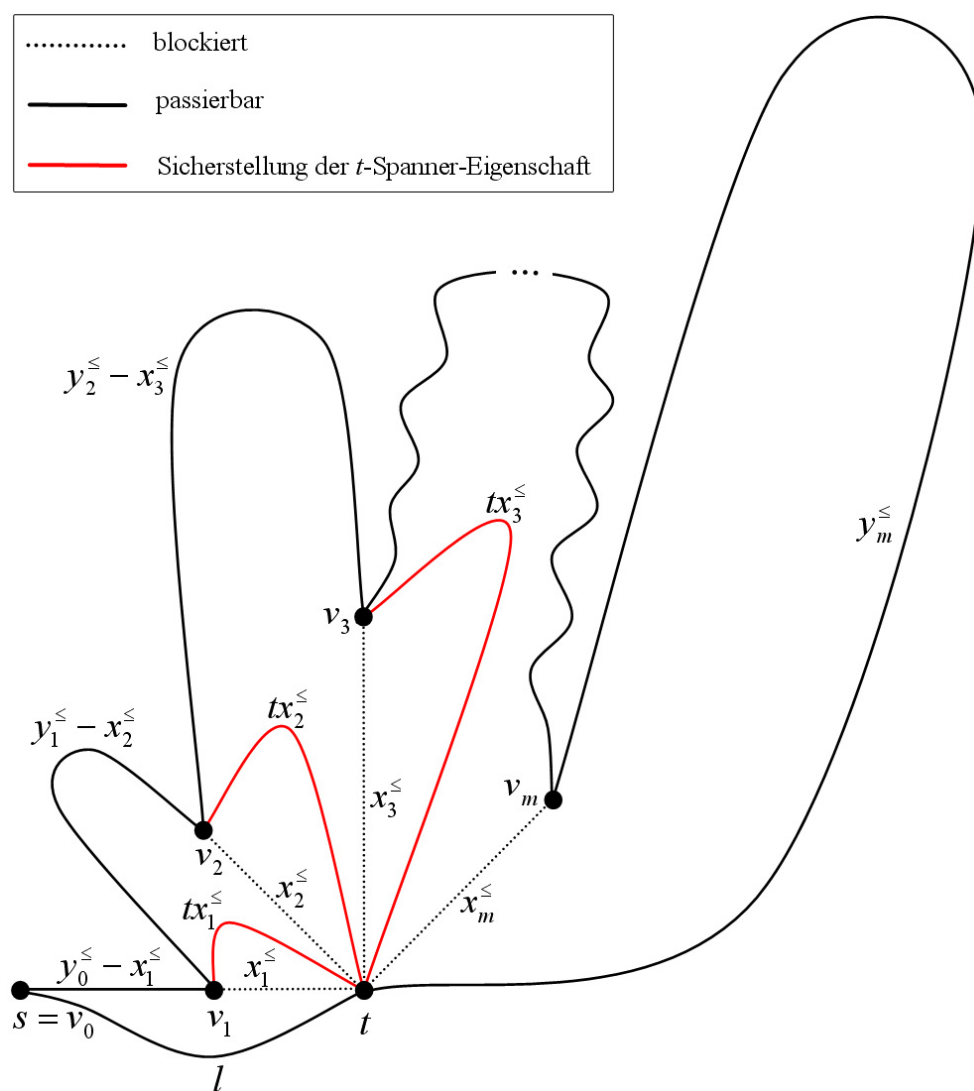
$$\text{Greedy}'(P^{\max}) \geq \text{Greedy}'(P).$$

Beweis. Es sei $P \in \Pi$ eine beliebige Problem Instanz, zu deren Lösung Greedy' auf genau $m \in \{1, \dots, k\}$ ausfallende Kanten trifft, bevor t erreicht wird, und es sei $l := \text{OPT}(P)$. Wir können annehmen, dass der Stretch-Faktor $t > 1$ ist. Denn nach Satz 3.18 gilt für $t = 1$, unabhängig davon wie P konstruiert ist, immer $\text{Greedy}'(P) = \text{OPT}(P)$, und Lemma 3.21 ist trivialerweise erfüllt. Obwohl wir nicht wissen, wie die Problem Instanz P aussieht, können wir dennoch die Kosten angeben, die die Graph-Strategie Greedy' beim Lösen von P verursacht:

$$\begin{aligned} \text{Greedy}'(P) &= (d_{G_0}(v_0, t) - d_{G_0}(v_1, t)) + (d_{G_1}(v_1, t) - d_{G_1}(v_2, t)) + \dots \\ &\quad + (d_{G_{m-1}}(v_{m-1}, t) - d_{G_{m-1}}(v_m, t)) + d_{G_m}(v_m, t) \\ &= (y_0 - x_1) + (y_1 - x_2) + \dots + (y_{m-1} - x_m) + y_m \\ &= w_m - l + y_m. \end{aligned} \tag{3}$$

Unabhängig von dem genauen Aufbau der Problem Instanz P können wir eine Problem Instanz $P^{\leq}$ angeben, die wie in Abbildung 3.18 aufgebaut ist und für die gilt:

- $\text{Greedy}'(P^{\leq}) = \text{Greedy}'(P)$
- $\text{OPT}(P^{\leq}) = \text{OPT}(P) = l$
- Greedy' trifft beim Lösen von $P^{\leq}$ ebenfalls auf genau m ausfallende Kanten
- $y_i^{\leq} = y_i$, und $w_i^{\leq} = w_i$, für alle $i \in \{0, \dots, m\}$
- $x_i^{\leq} = x_i$, für alle $i \in \{1, \dots, m\}$

Abbildung 3.18: Aufbau der Probleminstanz $P^{\leq}$

Wie Abbildung 3.18 zeigt, führen unter anderem die Pfade $(s, v_1, \dots, v_m, t)$ und $\{s, t\}$ vom Start- zum Zielknoten, wobei $\{s, t\}$ die optimale Lösung mit Kosten l darstellt. Offensichtlich ist $|(s, v_1, \dots, v_m, t)| = \text{Greedy}'(P)$, denn es gilt:

$$\begin{aligned} |(s, v_1, \dots, v_m, t)| &= (y_0^{\leq} - x_1^{\leq}) + (y_1^{\leq} - x_2^{\leq}) + \dots + (y_{m-1}^{\leq} - x_m^{\leq}) + y_m^{\leq} \\ &= (y_0 - x_1) + (y_1 - x_2) + \dots + (y_{m-1} - x_m) + y_m && \text{nach Konstruktion} \\ &= \text{Greedy}'(P). && \text{nach (3)} \end{aligned}$$

Es muss jedoch auch sichergestellt sein, dass die Graph-Strategie *Greedy'* tatsächlich dem Pfad $(s, v_1, \dots, v_m, t)$ folgt. Da auf Grund der *Greedy'*-Charakteristik bereits in P für alle $i \in \{0, \dots, m\}$ die Ungleichung $y_i \leq w_i$ erfüllt sein muss, gilt entsprechend in $P^{\leq}$ die Ungleichung $y_i^{\leq} \leq w_i^{\leq}$. Die Gleichheit $w_i^{\leq} = w_i$ wird in $P^{\leq}$ zwar nicht wie $x_i^{\leq} = x_i$, $y_i^{\leq} = y_i$ und $\text{OPT}(P^{\leq}) = \text{OPT}(P)$ explizit festgelegt, sie ergibt sich aber implizit aus letzteren beiden Gleichungen. Falls für alle $i \in \{0, \dots, m\}$ $y_i^{\leq} < w_i^{\leq}$ und $y_i^{\leq} < tx_i^{\leq}$ gilt, folgt *Greedy'* (genau wie es *Greedy* tun würde) dem Pfad $(s, v_1, \dots, v_m, t)$, denn dann existiert an jedem Knoten v_i (einschließlich $v_0 = s$) nur genau ein Pfad in G_i nach t , der die Länge y_i hat. Falls aber für ein $i \in \{0, \dots, m\}$ mindestens eine der Gleichungen $y_i^{\leq} = w_i^{\leq}$ bzw. $y_i^{\leq} = tx_i^{\leq}$ erfüllt ist, existiert an v_i mehr als ein Pfad in G_i nach t , der die Länge y_i hat. Dann folgt *Greedy'* (im Unterschied zu *Greedy*) gemäß Definition 3.20 jedoch weiterhin eindeutig dem Pfad $(s, v_1, \dots, v_m, t)$. Damit ist sichergestellt, dass die Graph-Strategie *Greedy'* in $P^{\leq}$ stets dem Pfad $(s, v_1, \dots, v_m, t)$ folgt, dabei auf insgesamt m ausfallende Kanten trifft und die Kosten $\text{Greedy}'(P^{\leq}) = \text{Greedy}'(P)$ verursacht. Die roten Kanten in Abbildung 3.18 stellen sicher, dass der durch Entfernen von m Kanten entstehende Restgraph ein t -Spanner des Ursprungsgraphen ist.

Es sei $P^=$ eine Problemistanz, die wie $P^{\leq}$ aus Abbildung 3.18 aufgebaut ist, sich jedoch durch folgende Eigenschaften auszeichnet:

- $\text{OPT}(P^=) = \text{OPT}(P^{\leq}) = l$
- *Greedy'* trifft beim Lösen von $P^=$ auf genau m ausfallende Kanten (analog zu $P^{\leq}$)
- $P^=$ erfüllt für alle $i \in \{0, \dots, m\}$ die Gleichung $y_i^= = w_i^=$
- $P^=$ erfüllt für alle $i \in \{1, \dots, m\}$ die Gleichung $y_i^= = tx_i^=$

Zum Beweis von Lemma 3.21 werden wir zunächst folgende Aussage zeigen:

Lemma 3.22. *Es sei $i \in \{0, \dots, m\}$. Erfüllt eine Problemistanz $P^{\leq}$, die wie in Abbildung 3.18 aufgebaut ist, für alle j mit $0 \leq j < i$ die Gleichung $y_j^{\leq} = w_j^{\leq}$ und für alle j mit $0 < j < i$ die Gleichung $y_j^{\leq} = tx_j^{\leq}$, dann gibt es eine Problemistanz $\tilde{P}^{\leq}$, die sogar für alle j mit $0 \leq j < i+1$ die Gleichung $\tilde{y}_j^{\leq} = \tilde{w}_j^{\leq}$ und für alle j mit $0 < j < i+1$ die Gleichung $\tilde{y}_j^{\leq} = t\tilde{x}_j^{\leq}$ erfüllt, mit*

$$\text{Greedy}'(\tilde{P}^{\leq}) \geq \text{Greedy}'(P^{\leq}).$$

Dabei bezeichnet $\tilde{P}^{\leq}$ eine Probleminstance mit $\text{OPT}(\tilde{P}^{\leq}) = \text{OPT}(P^{\leq})$, in der Greedy' ebenfalls auf m ausfallende Kanten trifft, bevor t erreicht wird.

Beweis. Wir unterscheiden in diesem Beweis die Fälle $i = 0$ und $i \in \{1, \dots, m\}$.

Es sei zunächst also $i = 0$, und wir betrachten die in Abbildung 3.19 angedeutete Probleminstance $P^{\leq}$, von der wir wissen, dass sie $y_0^{\leq} \leq w_0^{\leq}$ erfüllt. Es sei $\tilde{P}^{\leq}$ eine Probleminstance,

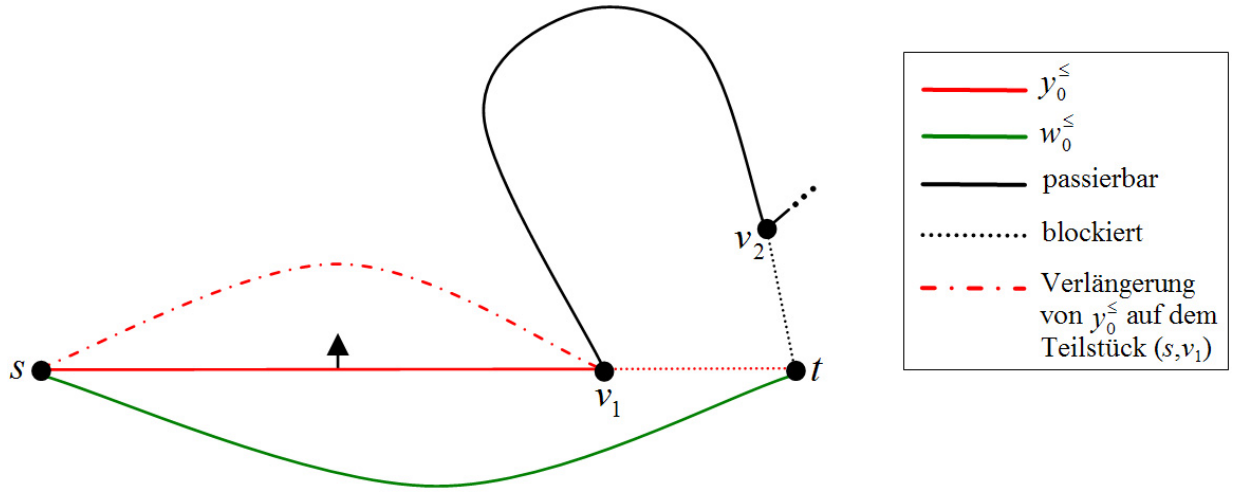


Abbildung 3.19: Konstruktion von $\tilde{P}^{\leq}$ aus $P^{\leq}$ im Fall $i = 0$

die mit Ausnahme der Kantenlänge von $\{s, v_1\}$ genauso aufgebaut ist wie $P^{\leq}$. Die Kante $\{s, v_1\}$ ist in $\tilde{P}^{\leq}$ genau $w_0^{\leq} - y_0^{\leq}$ länger als in $P^{\leq}$. Man beachte den Fall $w_0^{\leq} = y_0^{\leq}$. Hier bleibt die Kante $\{s, v_1\}$ in $\tilde{P}^{\leq}$ gegenüber $P^{\leq}$ unverändert, was sich dadurch erklärt, dass $P^{\leq}$ hier bereits die Eigenschaften von $\tilde{P}^{\leq}$ aus Lemma 3.22 erfüllt. Auf jeden Fall gilt aber für $\tilde{P}^{\leq}$:

$$\begin{aligned} \tilde{y}_0^{\leq} &= y_0^{\leq} + (w_0^{\leq} - y_0^{\leq}) && \text{wegen Verlängerung der Kante } \{s, v_1\} \\ &= w_0^{\leq} \\ &= \tilde{w}_0^{\leq}. && \text{die Kante } \{s, t\} \text{ bleibt unverändert} \end{aligned}$$

Die Probleminstance $\tilde{P}^{\leq}$ erfüllt nun also die Gleichung $\tilde{y}_0^{\leq} = \tilde{w}_0^{\leq}$. Weil sich für alle $j > 0$ einzig die Werte $\tilde{w}_j^{\leq}$ ändern (sie werden höchstens größer), bleiben die Bedingungen $\tilde{y}_j^{\leq} \leq \tilde{w}_j^{\leq}$ und $\tilde{y}_j^{\leq} \leq t\tilde{x}_j^{\leq}$ erfüllt. Darum trifft Greedy' auch in $\tilde{P}^{\leq}$ auf insgesamt m ausfallende Kanten. Da die Kante $\{s, v_1\}$ dort höchstens länger ist als in $P^{\leq}$, gilt $\text{Greedy}'(\tilde{P}^{\leq}) \geq \text{Greedy}'(P^{\leq})$. Für die Kosten der optimalen Lösungen gilt $\text{OPT}(\tilde{P}^{\leq}) = \text{OPT}(P^{\leq})$. Wir haben mit $\tilde{P}^{\leq}$ also eine Probleminstance gefunden, die Lemma 3.22 erfüllt.

Es sei nun $i > 0$ beliebig. Die folgende Argumentation lässt sich auf jedes $i \in \{1, \dots, m\}$ anwenden. Aus Lemma 3.22 wissen wir, dass in der Probleminstance $P^\leq$ für alle $j < i$ die Gleichungen $y_j^\leq = w_j^\leq$ und $y_j^\leq = tx_j^\leq$ erfüllt sind und für i die Ungleichungen $y_i^\leq \leq w_i^\leq$ und $y_i^\leq \leq tx_i^\leq$ gelten. Wir unterscheiden nun die folgenden beiden Fälle:

- $y_i^\leq \leq w_i^\leq$ und $y_i^\leq < tx_i^\leq$ (Fall I)
- $y_i^\leq \leq w_i^\leq$ und $y_i^\leq = tx_i^\leq$ (Fall II)

Beide Fälle (I und II) sind in Abbildung 3.20 dargestellt. Fall II liegt entweder von vornherein vor oder aber nach Ausführung von Fall I.

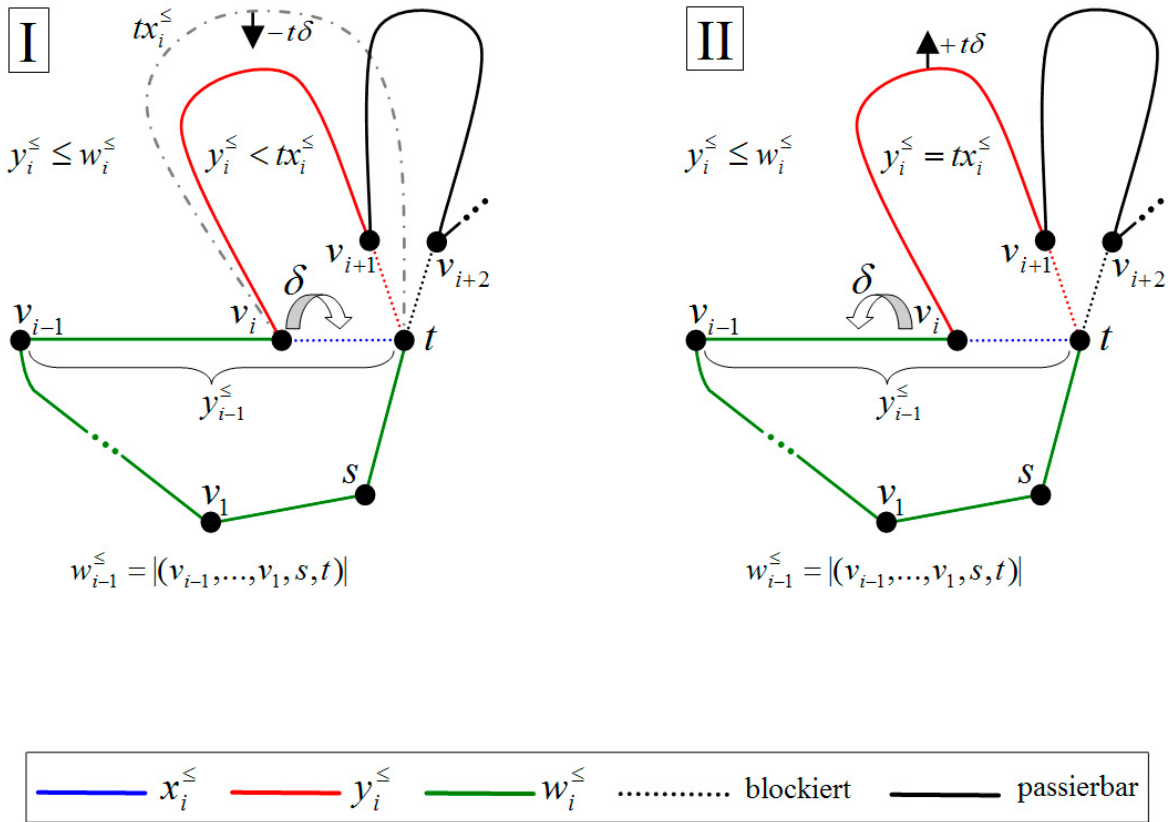


Abbildung 3.20: Konstruktion von $\tilde{P}^\leq$ aus $P^\leq$ im Fall $i > 0$

Fall I: Es gelte $y_i^\leq \leq w_i^\leq$ und $y_i^\leq < tx_i^\leq$.

In diesem Fall betrachten wir eine Probleminstance $\tilde{P}^\leq$, die mit Ausnahme der Kantenlängen von $\{v_{i-1}, v_i\}$ und $\{v_i, t\}$ genauso aufgebaut ist wie $P^\leq$. Die Kante $\{v_i, t\}$ ist in $\tilde{P}^\leq$ genau um

$$\delta := \frac{tx_i^\leq - y_i^\leq}{t}, \text{ mit } 0 < \delta < x_i^\leq,$$

kürzer als in $P^\leq$. Der Wertebereich von δ folgt aus $y_i^\leq \neq 0$ (sonst hätte *Greedy'* bereits den Zielknoten erreicht) und $y_i^\leq < tx_i^\leq$. Daher gilt in $\tilde{P}^\leq$

$$\tilde{x}_i^\leq = x_i^\leq - \delta,$$

woraus folgende Schlussfolgerung gezogen werden kann:

$$\begin{aligned} t\tilde{x}_i^\leq &= t(x_i^\leq - \delta) && \text{wegen } \tilde{x}_i^\leq = x_i^\leq - \delta \\ &= t(x_i^\leq - \frac{tx_i^\leq - y_i^\leq}{t}) && \text{nach Definition von } \delta \\ &= tx_i^\leq - tx_i^\leq + y_i^\leq \\ &= y_i^\leq \\ &= \tilde{y}_i^\leq. && \text{der Pfad } (v_i, v_{i+1}, t) \text{ bleibt unverändert} \end{aligned}$$

Damit die Veränderung der Problemistanz $\tilde{P}^\leq$ gegenüber $P^\leq$ nur lokale Auswirkungen hat, wird die Kante $\{v_{i-1}, v_i\}$ um δ verlängert. Weil damit auch $\tilde{w}_i^\leq$ um δ wächst, bleibt $\tilde{y}_i^\leq \leq \tilde{w}_i^\leq$ erfüllt (präziser gilt nun sogar $\tilde{y}_i^\leq < \tilde{w}_i^\leq$) und gleichzeitig ist $\tilde{y}_{i-1}^\leq = \tilde{w}_{i-1}^\leq$ sichergestellt. Anschaulich könnte man sagen, in $\tilde{P}^\leq$ ist lediglich v_i gegenüber $P^\leq$ um δ in Richtung t verschoben. Wie oben gezeigt wurde, erfüllt die Problemistanz $\tilde{P}^\leq$ nun also $\tilde{y}_i^\leq < \tilde{w}_i^\leq$ und $\tilde{y}_i^\leq = t\tilde{x}_i^\leq$. Für $j \neq i$ ändern sich gegenüber $P^\leq$ einzig die Werte $\tilde{w}_j^\leq$, mit $j > i$. Da diese jedoch größer werden, bleiben für alle $0 \leq j \leq m$ die Bedingungen $\tilde{y}_j^\leq \leq \tilde{w}_j^\leq$ und $\tilde{y}_j^\leq \leq t\tilde{x}_j^\leq$ erfüllt. Darum trifft *Greedy'* auch in $\tilde{P}^\leq$ auf insgesamt m ausfallende Kanten. Und weil der von *Greedy'* benutzte Pfad in $\tilde{P}^\leq$ um δ länger ist als in $P^\leq$ (da $\tilde{w}_i^\leq = w_i^\leq + \delta$), gilt wegen $\delta > 0$: $\text{Greedy}'(\tilde{P}^\leq) \geq \text{Greedy}'(P^\leq)$. Für die Kosten der optimalen Lösungen gilt $\text{OPT}(\tilde{P}^\leq) = \text{OPT}(P^\leq)$. Darüber hinaus gilt $y_i^\leq < w_i^\leq$ und $y_i^\leq = tx_i^\leq$ und wir fahren mit Fall II fort.

Fall II: Es gelte $y_i^\leq \leq w_i^\leq$ und $y_i^\leq = tx_i^\leq$.

In diesem Fall betrachten wir eine Problemistanz $\tilde{P}^\leq$, die mit Ausnahme der Kantenlängen von $\{v_{i-1}, v_i\}$, $\{v_i, t\}$ und $\{v_i, v_{i+1}\}$ genauso aufgebaut ist wie $P^\leq$. Es sei

$$\delta := \frac{w_i^\leq - y_i^\leq}{t + 1}, \text{ mit } 0 \leq \delta < \overbrace{|\{v_{i-1}, v_i\}|}^{\text{in } P^\leq}.$$

Man beachte den Spezialfall $y_i^\leq = w_i^\leq$ (im Allgemeinen gilt $y_i^\leq \leq w_i^\leq$). Hier erfüllt $P^\leq$ bereits alle Anforderungen von $\tilde{P}^\leq$ aus Lemma 3.22, und es gilt daher $\delta = 0$. Dass $\delta < |\{v_{i-1}, v_i\}|$ ist lässt sich folgendermaßen zeigen: Nach Voraussetzung sind in der Problemistanz $P^\leq$ für alle $j < i$ die Gleichungen $y_j^\leq = w_j^\leq$ und $y_j^\leq = tx_j^\leq$ erfüllt, woraus insbesondere $y_{i-1}^\leq = w_{i-1}^\leq$ folgt. Zudem können wir, wie zu Beginn des Beweises erläutert,

$t > 1$ voraussetzen. Damit lässt sich δ wie folgt abschätzen:

$$\begin{aligned}
\delta &= \frac{w_i^{\leq} - y_i^{\leq}}{t+1} && \text{nach Definition von } \delta \\
&\leq \frac{w_i^{\leq} - y_i^{\leq}}{2} && \text{wegen } t > 1 \text{ und } w_i^{\leq} - y_i^{\leq} \geq 0 \\
&< \frac{w_i^{\leq} - x_i^{\leq}}{2} && \text{wegen } y_i^{\leq} > x_i^{\leq} \text{ (folgt aus } y_i^{\leq} = tx_i^{\leq} \text{ und } t > 1) \\
&= \frac{|\{v_{i-1}, v_i\}| + w_{i-1}^{\leq} - x_i^{\leq}}{2} && |\{v_{i-1}, v_i\}| = w_i^{\leq} - w_{i-1}^{\leq}, \text{ vgl. Abbildung 3.20} \\
&= \frac{|\{v_{i-1}, v_i\}| + w_{i-1}^{\leq} + |\{v_{i-1}, v_i\}| - y_{i-1}^{\leq}}{2} && |\{v_{i-1}, v_i\}| = y_{i-1}^{\leq} - x_i^{\leq}, \text{ vgl. Abbildung 3.20} \\
&= \frac{2|\{v_{i-1}, v_i\}|}{2} && \text{wegen } y_{i-1}^{\leq} = w_{i-1}^{\leq} \\
&= |\{v_{i-1}, v_i\}|.
\end{aligned}$$

Verglichen mit $P^{\leq}$, verändern wir nun in $\tilde{P}^{\leq}$ für jeden δ -Wert die Längen oben genannter Kanten wie folgt:

- $\{v_{i-1}, v_i\}$ wird um δ verkürzt
- $\{v_i, t\}$ wird um δ verlängert
- $\{v_i, v_{i+1}\}$ wird um $t\delta$ verlängert

Für den Fall $i = m$ wird abweichend statt der Kante $\{v_i, v_{i+1}\}$ die Kante $\{v_i, t\}$ um $t\delta$ verlängert, da es in $\tilde{P}^{\leq}$ keinen Knoten v_{m+1} gibt, vergleiche Abbildung 3.18. Die beschriebenen Änderungen der Kantenlängen haben auf die Größen $\tilde{w}_i^{\leq}$, $\tilde{x}_i^{\leq}$ und $\tilde{y}_i^{\leq}$ folgende Auswirkungen:

- $\tilde{w}_i^{\leq} = w_i^{\leq} - \delta$
- $\tilde{x}_i^{\leq} = x_i^{\leq} + \delta$
- $\tilde{y}_i^{\leq} = y_i^{\leq} + t\delta$

Für das Verhältnis zwischen $\tilde{y}_i^{\leq}$ und $\tilde{w}_i^{\leq}$ ergibt sich daraus:

$$\begin{aligned}
\tilde{w}_i^{\leq} &= w_i^{\leq} - \delta && \text{nach Konstruktion von } \tilde{P}^{\leq} \text{ (s. o.)} \\
&= w_i^{\leq} - \frac{w_i^{\leq} - y_i^{\leq}}{t+1} && \text{nach Definition von } \delta \\
&= \frac{tw_i^{\leq} + w_i^{\leq} - w_i^{\leq} + y_i^{\leq}}{t+1} \\
&= \frac{tw_i^{\leq} - ty_i^{\leq}}{t+1} + \frac{(t+1)y_i^{\leq}}{t+1} \\
&= y_i^{\leq} + t \frac{w_i^{\leq} - y_i^{\leq}}{t+1} \\
&= y_i^{\leq} + t\delta && \text{nach Definition von } \delta \\
&= \tilde{y}_i^{\leq}. && \text{nach Konstruktion von } \tilde{P}^{\leq} \text{ (s. o.)}
\end{aligned}$$

Und für das Verhältnis zwischen $\tilde{y}_i^{\leq}$ und $t\tilde{x}_i^{\leq}$ gilt:

$$\begin{aligned}
\tilde{x}_i^{\leq} &= x_i^{\leq} + \delta && \text{nach Konstruktion von } \tilde{P}^{\leq} \text{ (s. o.)} \\
\Rightarrow t\tilde{x}_i^{\leq} &= tx_i^{\leq} + t\delta \\
&= y_i^{\leq} + t\delta && \text{nach Voraussetzung gilt } y_i^{\leq} = tx_i^{\leq} \\
&= \tilde{y}_i^{\leq}. && \text{nach Konstruktion von } \tilde{P}^{\leq} \text{ (s. o.)}
\end{aligned}$$

Anschaulich könnte man sagen, in $\tilde{P}^{\leq}$ ist lediglich v_i gegenüber $P^{\leq}$ um δ in Richtung v_{i-1} verschoben (weshalb $\tilde{y}_i^{\leq} = \tilde{w}_i^{\leq}$ gilt) und $\{v_i, v_{i+1}\}$ gerade soviel länger als in $P^{\leq}$ (nämlich genau $t\delta$), dass $\tilde{y}_i^{\leq} = t\tilde{x}_i^{\leq}$ erfüllt bleibt. Die „Verschiebung“ von v_i hat keinen Einfluss auf $\tilde{w}_j^{\leq}$, $\tilde{x}_j^{\leq}$ und $\tilde{y}_j^{\leq}$, mit $j < i$. Für $j > i$ ändern sich gegenüber $P^{\leq}$ einzig die Werte $\tilde{w}_j^{\leq}$. Da diese jedoch höchstens größer werden, bleiben für alle $0 \leq j \leq m$ die Bedingungen $\tilde{y}_j^{\leq} \leq \tilde{w}_j^{\leq}$ und $\tilde{y}_j^{\leq} \leq t\tilde{x}_j^{\leq}$ erfüllt. Darum trifft *Greedy'* auch in $\tilde{P}^{\leq}$ auf insgesamt m ausfallende Kanten. Weil der von *Greedy'* benutzte Pfad in $\tilde{P}^{\leq}$ insgesamt um $(t-1)\delta$ länger ist als in $P^{\leq}$ (da $\tilde{w}_i^{\leq} = w_i^{\leq} - \delta$ und $\tilde{y}_i^{\leq} = y_i^{\leq} + t\delta$), gilt wegen $t, \delta > 0$: $\text{Greedy}'(\tilde{P}^{\leq}) \geq \text{Greedy}'(P^{\leq})$. Für die Kosten der optimalen Lösungen gilt $\text{OPT}(\tilde{P}^{\leq}) = \text{OPT}(P^{\leq})$. Wir haben mit $\tilde{P}^{\leq}$ also eine Problemistanz gefunden, die Lemma 3.22 erfüllt.

Damit haben wir gezeigt, dass sich zu jeder Problemistanz $P^{\leq}$, die für alle j mit $0 \leq j < i$ die Gleichung $y_j^{\leq} = w_j^{\leq}$ und für alle j mit $0 < j < i$ die Gleichung $y_j^{\leq} = tx_j^{\leq}$ erfüllt, eine Problemistanz $\tilde{P}^{\leq}$ angeben lässt, die sogar für alle j mit $0 \leq j < i+1$ die Gleichung $\tilde{y}_j^{\leq} = \tilde{w}_j^{\leq}$ und für alle j mit $0 < j < i+1$ die Gleichung $\tilde{y}_j^{\leq} = t\tilde{x}_j^{\leq}$ erfüllt und für die $\text{Greedy}'(\tilde{P}^{\leq}) \geq \text{Greedy}'(P^{\leq})$ gilt. Dabei bezeichnet $\tilde{P}^{\leq}$ eine Problemistanz mit $\text{OPT}(\tilde{P}^{\leq}) = \text{OPT}(P^{\leq})$, in der *Greedy'* ebenfalls auf m ausfallende Kanten trifft, bevor t erreicht wird. Daraus folgt die Korrektheit von Lemma 3.22.

□

Mittels Lemma 3.22 können wir durch vollständige Induktion auch die folgende Aussage zeigen, aus der schließlich die Korrektheit von Lemma 3.21 folgt:

Lemma 3.23. *Es sei $i \in \{0, \dots, m+1\}$. Sind in der Problem Instanz $P^{\leq}$, die wie in Abbildung 3.18 aufgebaut ist, für alle j mit $0 \leq j < i$ die Gleichung $y_j^{\leq} = w_j^{\leq}$ und für alle j mit $0 < j < i$ die Gleichung $y_j^{\leq} = tx_j^{\leq}$ erfüllt, dann gilt:*

$$\text{Greedy}'(P^{\leq}) \leq \text{Greedy}'(P^=).$$

Dabei bezeichnet $P^=$ die auf Seite 80 definierte Problem Instanz.

Beweis. Es sei $A(i)$ die Aussage aus Lemma 3.23 für ein $i \in \{0, \dots, m+1\}$. Wir werden nun per von $m+1$ ausgehender rückwärts gerichteter Induktion zeigen, dass $A(i)$ für alle $i \in \{0, \dots, m+1\}$ gilt.

Der Induktionsanfang ist mit $i = m+1$ trivial, denn dann sind $P^{\leq}$ und $P^=$ identische Problem Instanzen.

Wir wissen also als Induktionsannahme, dass für jede Problem Instanz $P^{\leq}$, die für alle j mit $0 \leq j \leq i$ die Gleichung $y_j^{\leq} = w_j^{\leq}$ und für alle j mit $0 < j \leq i$ die Gleichung $y_j^{\leq} = tx_j^{\leq}$ erfüllt, $\text{Greedy}'(P^{\leq}) \leq \text{Greedy}'(P^=)$ gilt.

Im Induktionsschritt müssen wir zeigen, dass wir die gleiche Schlussfolgerung auch schon ziehen können, wenn wir nur wissen, dass in $P^{\leq}$ für alle j mit $0 \leq j < i$ die Gleichung $y_j^{\leq} = w_j^{\leq}$ und für alle j mit $0 < j < i$ die Gleichung $y_j^{\leq} = tx_j^{\leq}$ erfüllt ist. Wir können nach Lemma 3.22 eine Problem Instanz $\tilde{P}^{\leq}$ finden, die die beiden Gleichungen $\tilde{y}_j^{\leq} = \tilde{w}_j^{\leq}$ und $\tilde{y}_j^{\leq} = t\tilde{x}_j^{\leq}$ bis $j = i$ erfüllt und für die $\text{Greedy}'(\tilde{P}^{\leq}) \geq \text{Greedy}'(P^{\leq})$ und $\text{OPT}(\tilde{P}^{\leq}) = \text{OPT}(P^{\leq})$ gilt. Nach Induktionsannahme wissen wir aber, dass $\text{Greedy}'(P^=) \geq \text{Greedy}'(\tilde{P}^{\leq})$ und $\text{OPT}(P^=) = \text{OPT}(\tilde{P}^{\leq})$ ist. Damit ist bewiesen, dass $\text{Greedy}'(P^{\leq}) \leq \text{Greedy}'(P^=)$ bereits dann erfüllt ist, wenn wir nur wissen, dass in $P^{\leq}$ für alle j mit $0 \leq j < i$ die Gleichung $y_j^{\leq} = w_j^{\leq}$ und für alle j mit $0 < j < i$ die Gleichung $y_j^{\leq} = tx_j^{\leq}$ erfüllt ist. Wir haben also $A(i+1)$ aus $A(i)$ gefolgert und damit die Korrektheit von Lemma 3.23 bewiesen.

□

Damit haben wir folgendes gezeigt: Zu jeder beliebigen Problem Instanz P mit $\text{OPT}(P) = l$, in der *Greedy'* auf insgesamt $m \in \{1, \dots, k\}$ ausfallende Kanten trifft, bevor t erreicht wird, können wir eine Problem Instanz $P^{\leq}$ angeben, wie sie oben beschrieben wurde. Für die von *Greedy'* verursachten Kosten gilt nach Lemma 3.23 die Abschätzung

$$\text{Greedy}'(P) = \text{Greedy}'(P^{\leq}) \leq \text{Greedy}'(P^=) = \text{Greedy}'(P^{\max}),$$

wobei $P^=$ die oben beschriebene zu $P^{\leq}$ gehörige Problem Instanz darstellt, die der in Lemma 3.21 beschriebenen Problem Instanz $P^{\max}$, hinsichtlich der von *Greedy'* verursachten

sowie optimalen Lösungskosten, entspricht. Daraus folgt die Korrektheit von Lemma 3.21.

□

Da sich im letzten Teil des Beweises die Größen w_i , x_i und y_i stets eindeutig auf die Problem Instanz $P^{\max}$ beziehen, verzichten wir von nun an der Lesbarkeit wegen auf die differenziertere Schreibweise $w_i^{\max}$, $x_i^{\max}$ und $y_i^{\max}$.

Betrachten wir nun eine beliebige Problem Instanz $P \in \Pi$, zu deren Lösung *Greedy'* auf insgesamt $m \in \{1, \dots, k\}$ ausfallende Kanten trifft, bevor t erreicht wird. Die optimalen Lösungskosten seien $\text{OPT}(P) =: l$. Nach Lemma 3.21 sind die Kosten von *Greedy'* dann beschränkt durch:

$$\text{Greedy}'(P) \leq \text{Greedy}'(P^{\max}) = w_m - l + y_m \stackrel{y_i = tx_i}{=} w_m - l + tx_m. \quad (4)$$

Dabei sei $P^{\max} \in \Pi$ wie in Lemma 3.21 definiert. Der letzte Schritt besteht nun darin, die Kosten $\text{Greedy}'(P^{\max})$ konkret zu berechnen. Da nach Lemma 3.21 für die Problem Instanz $P^{\max}$ $y_i = w_i$ und $y_i = tx_i$ ist, gilt insbesondere

$$tx_i = w_i, \text{ für alle } i \in \{1, \dots, m\}. \quad (5)$$

Zudem gilt nach Lemma 3.21 $y_0 = w_0$. Damit können wir x_1 wie folgt berechnen:

$$\begin{aligned} tx_1 &= w_1 && \text{nach (5)} \\ &= w_0 + y_0 - x_1 && \text{nach Definition von } w_1 \\ &= 2w_0 - x_1 && y_0 = w_0 \text{ (nach Lemma 3.21)} \\ &= 2l - x_1 && w_0 = d_{G_k}(s, t) = l \text{ (nach Definition)} \\ \Rightarrow x_1 &= \frac{2l}{t+1}. \end{aligned}$$

Allgemein gilt für alle $i \in \{0, \dots, m-1\}$:

$$\begin{aligned} tx_{i+1} &= w_{i+1} && \text{nach (5)} \\ &= w_i + y_i - x_{i+1} && \text{nach Definition von } w_{i+1} \\ &= w_i + tx_i - x_{i+1} && y_i = tx_i \text{ (nach Lemma 3.21)} \\ &= 2tx_i - x_{i+1} && \text{nach (5)} \\ \Rightarrow x_{i+1} &= \frac{2tx_i}{t+1}. \end{aligned}$$

Diese Rekursion lässt sich mit Hilfe des oben berechneten Startwertes $x_1 = \frac{2l}{t+1}$ in folgende geschlossene Formel überführen:

$$x_i = \frac{l}{t} \left(\frac{2t}{t+1} \right)^i.$$

Setzen wir diese Formel in Gleichung (5) ein, so erhalten wir:

$$w_i = l \left(\frac{2t}{t+1} \right)^i.$$

Setzen wir die Werte für x_i und w_i in Ungleichung (4) ein, so folgt daraus:

$$\begin{aligned} \text{Greedy}'(P) &\leq \text{Greedy}'(P^{\max}) \\ &= w_m - l + tx_m \\ &= l \left(\frac{2t}{t+1} \right)^m - l + t \frac{l}{t} \left(\frac{2t}{t+1} \right)^m \\ &= 2l \left(\frac{2t}{t+1} \right)^m - l. \end{aligned}$$

Allgemein können wir daraus schließen: Trifft *Greedy'* beim Lösen einer beliebigen Problem Instanz P mit $\text{OPT}(P) =: l$ auf insgesamt $m \in \{1, \dots, k\}$ ausfallende Kanten, bevor t erreicht wird, so gilt für die Kosten der Graph-Strategie *Greedy'*:

$$\begin{aligned} \text{Greedy}'(P) &\leq 2l \left(\frac{2t}{t+1} \right)^m - l \\ &= (2 \left(\frac{2t}{t+1} \right)^m - 1) \cdot l \\ &= (2 \left(\frac{2t}{t+1} \right)^m - 1) \cdot \text{OPT}(P). \end{aligned}$$

Nach Definition 3.20 sind die Kosten $\text{Greedy}'(P)$ für den Fall mehrdeutiger kürzester Wege maximal unter den Kosten aller *Greedy*-Graph-Strategien. Sind kürzeste Wege stets eindeutig, verursachen alle *Greedy*-Graph-Strategien identische Kosten zum Lösen von P . Wir können die Kosten der *Greedy*-Strategie also wie folgt abschätzen:

$$\text{Greedy}(P) \leq \text{Greedy}'(P) \leq (2 \left(\frac{2t}{t+1} \right)^m - 1) \cdot \text{OPT}(P).$$

Weil *Greedy* auf höchstens k ausfallende Kanten treffen kann, bevor t erreicht wird, haben wir folgende Aussage bewiesen: Werden in einem ungerichteten, gewichteten Graphen G maximal k Kanten so entfernt, dass der Restgraph G_k ein t -Spanner von G ist, so ist die Strategie *Greedy* auf G für alle $t \geq 1$

$$(2 \left(\frac{2t}{t+1} \right)^k - 1)\text{-kompetitiv.}$$

Zusammen mit der unteren Schranke von Seite 76 folgt daraus die Korrektheit von Satz 3.19.

□

Um die Aussage von Satz 3.19 etwas greifbarer zu machen, möchten wir noch darlegen, wie der kompetitive Faktor konkret für bestimmte Stretch-Faktoren t aussieht. Dazu betrachten wir das folgende Korollar, in dem die kompetitiven Faktoren der *Greedy*-Strategie für ausgewählte t -Werte angegeben sind:

Korollar 3.24. *Werden in einem ungerichteten, gewichteten Graphen G maximal k Kanten so entfernt, dass der Restgraph G' ein t -Spanner von G ist, so ist die Strategie Greedy auf G genau*

- 1-kompetitiv für $t = 1$,
- $(2(\frac{4}{3})^k - 1)$ -kompetitiv für $t = 2$,
- $(2(\frac{20}{11})^k - 1)$ -kompetitiv für $t = 10$,
- $(2^{k+1} - 1)$ -kompetitiv für „ t beliebig groß“.

Den kompetitiven Faktor 1 für $t = 1$ kennen wir bereits aus Satz 3.18, und der kompetitive Faktor $2^{k+1} - 1$ für beliebig große t ist uns bereits aus Kapitel 3.2.2 bekannt. Schließlich ist ein beliebig großer Stretch-Faktor gleichbedeutend damit, dass keine Einschränkungen an den Restgraphen gestellt werden.

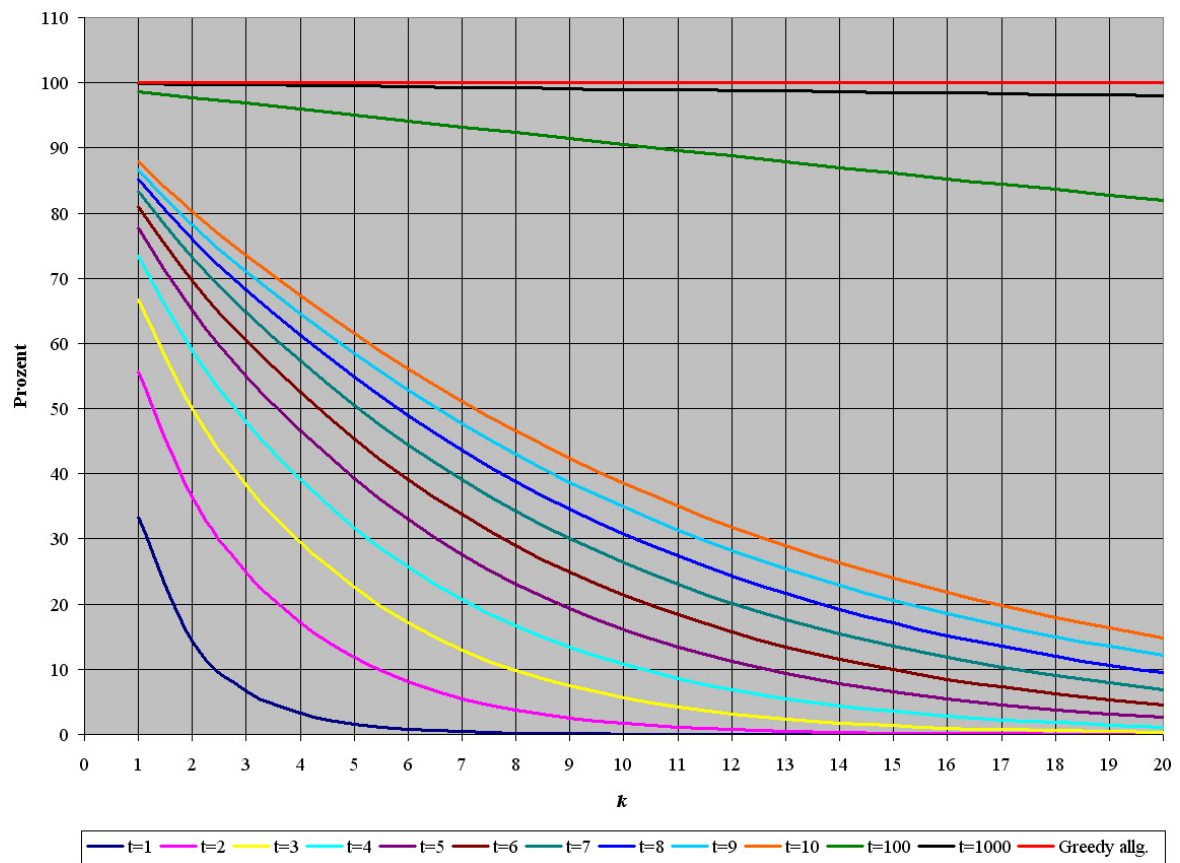
Insgesamt lässt sich festhalten, dass der kompetitive Faktor der *Greedy*-Strategie deutlich sinkt, wenn der Stretch-Faktor des Restgraphen stark eingeschränkt ist. Dabei wird der kompetitive Faktor umso kleiner, je mehr Kanten potentiell blockiert sind. Legen wir den kompetitiven Faktor von *Greedy* auf allgemeinen Graphen auf 100% fest, so können wir an Abbildung 3.21 ablesen, wie dieser Wert für abnehmende t und steigende k fällt. Beispielsweise liegt der kompetitive Faktor auf allgemeinen Graphen für $k = 1$ bei

$$2^{k+1} - 1 \stackrel{\text{Satz 3.3}}{=} 2^{1+1} - 1 = 3.$$

Beschränken wir den Stretch-Faktor des Restgraphen auf $t = 2$, so liegt der kompetitive Faktor bei

$$2\left(\frac{2t}{t+1}\right)^k - 1 \stackrel{\text{Satz 3.19}}{=} 2\left(\frac{2 \cdot 2}{2+1}\right)^1 - 1 \approx 1,6.$$

Das macht nur noch rund 55,5% des Ursprungswertes aus, vergleiche Abbildung 3.21. Mit wachsendem k sinkt der kompetitive Faktor im Vergleich zum uneingeschränkten Restgraphen zunehmend stark ab. Bereits für $k = 12$ (und $t = 2$) liegt dieser bei unter 1% des uneingeschränkten Falls. Die Grafik zeigt weiter, dass das Einsparpotential besonders für kleine Stretch-Faktoren mit der Anzahl möglicher Kantenblockaden stark ansteigt. Für große t hingegen ist der Unterschied des kompetitiven Faktors zwischen beiden Fällen gering und hängt zunehmend weniger von k ab.

Abbildung 3.21: Veränderung des kompetitiven Faktors in Abhängigkeit von t und k

Backtrack oder Greedy - welche Strategie ist im worst-case schneller?

Zusammenfassend wissen wir nach den letzten beiden Abschnitten über die Kompetitivität der Strategien *Backtrack* und *Greedy* folgendes: Dürfen nur solche Kanten blockiert werden, dass der Restgraph ein t -Spanner des Ursprungsgraphen ist,

- hat das auf den kompetitiven Faktor von *Backtrack* für $t > 1$ keinen Einfluss,
- verkleinert sich der kompetitive Faktor von *Greedy* auf $2\left(\frac{2t}{t+1}\right)^k - 1$.

Daher möchten wir nun der Frage nachgehen, ob die *Greedy*-Strategie der *Backtrack*-Strategie für gewisse Stretch-Faktoren hinsichtlich des kompetitiven Faktors nicht sogar überlegen ist. Zur Erinnerung sei noch einmal darauf hingewiesen, dass im allgemeinen Fall, wenn also beliebige Kanten blockiert werden dürfen, nach Kapitel 3.2 die folgenden Aussagen gelten:

- Der kompetitive Faktor der *Backtrack*-Strategie ist $2k + 1$.
- Der kompetitive Faktor der *Greedy*-Strategie ist $2^{k+1} - 1$.

Im worst-case ist *Greedy* also die deutlich unterlegene Strategie. Abbildung 3.22 zeigt, wie sich die kompetitiven Faktoren von *Backtrack* (schwarzer Funktionsgraph) und *Greedy* (roter Funktionsgraph) für steigende k -Werte entwickeln. Es ist deutlich zu erkennen, dass der *Greedy*-Graph für alle $k > 1$ über dem der *Backtrack*-Strategie liegt. Weiterhin sind in der Abbildung die Funktionsgraphen der kompetitiven Faktoren der *Greedy*-Strategie für $t \in \{1, \frac{5}{4}, \frac{3}{2}, \frac{7}{4}, 2, 3, 5, 100\}$ dargestellt. Dabei wird deutlich, dass der jeweilige Graph umso langsamer steigt, je kleiner t ist. Einige Abschnitte dieser Funktionsgraphen verlaufen sogar unterhalb des *Backtrack*-Funktionsgraphen. Das ist interessant, denn es bedeutet, dass die *Greedy*-Strategie für bestimmte Kombinationen aus t und k im worst-case schneller ist als *Backtrack*. Daher wollen wir uns jetzt überlegen, für welche t und k dies gilt. Offenbar ist der kompetitive Faktor von *Greedy* genau dann kleiner als der von *Backtrack*, wenn gilt:

$$2\left(\frac{2t}{t+1}\right)^k - 1 < 2k + 1.$$

Lösen wir diese Ungleichung nach t auf, erhalten wir:

$$t < \frac{\sqrt[k]{k+1}}{2 - \sqrt[k]{k+1}}, \text{ für } k > 1. \quad (*)$$

Angenommen, es sei eine Instanz des k -Canadian Traveller Problem gegeben und es sei bekannt, dass der Restgraph (aus dem bis zu k Kanten entfernt wurden), in Bezug auf den Ursprungsgraph, ein t -Spanner ist. Um die worst-case-Komplexität der Lösung des k -CTP unter Verwendung der Strategien *Backtrack* und *Greedy* zu minimieren, könnte man beide Strategien nach folgendem Schema einsetzen:

- t und k erfüllen die Ungleichung $(*)$
 $\Rightarrow$ Benutze die *Greedy*-Strategie.

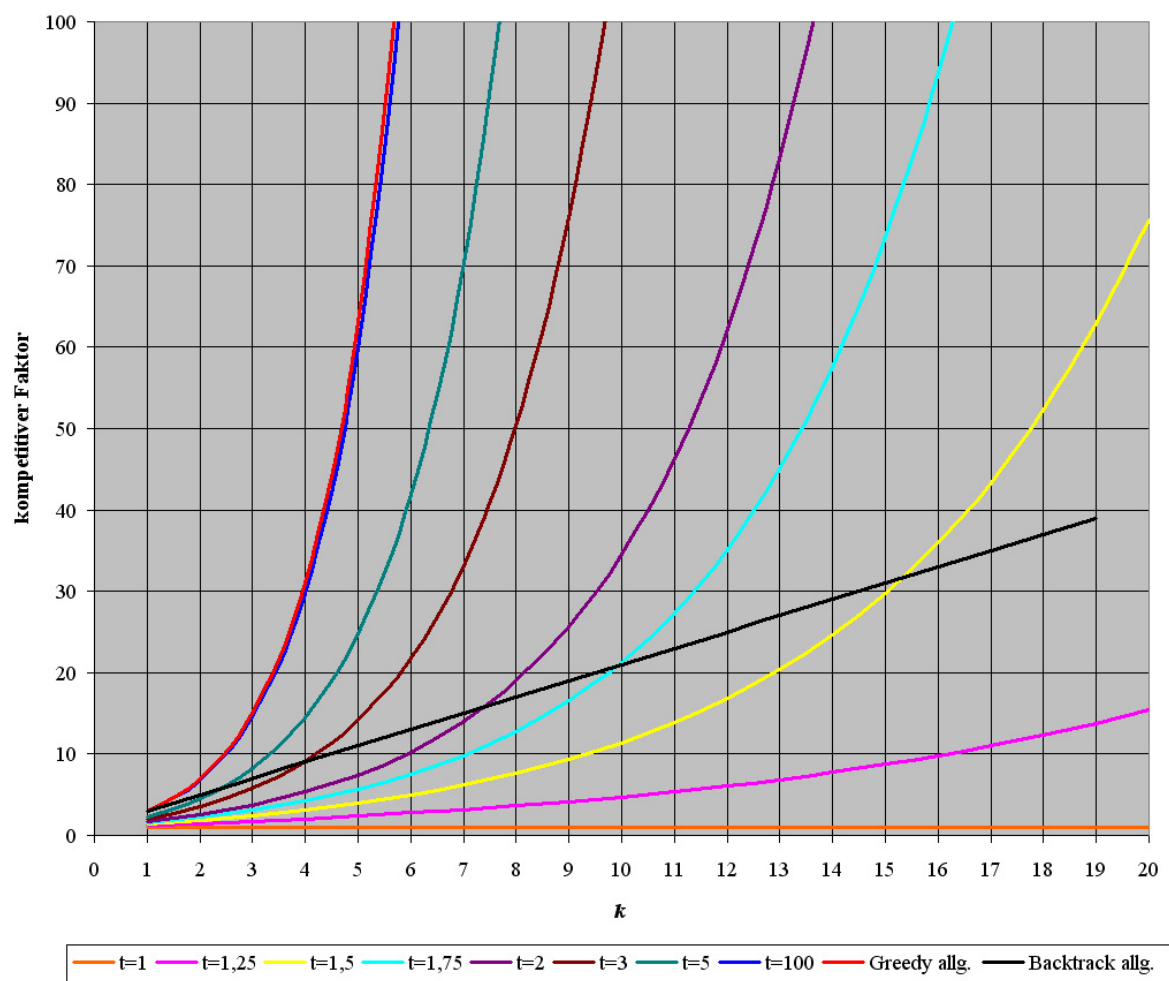
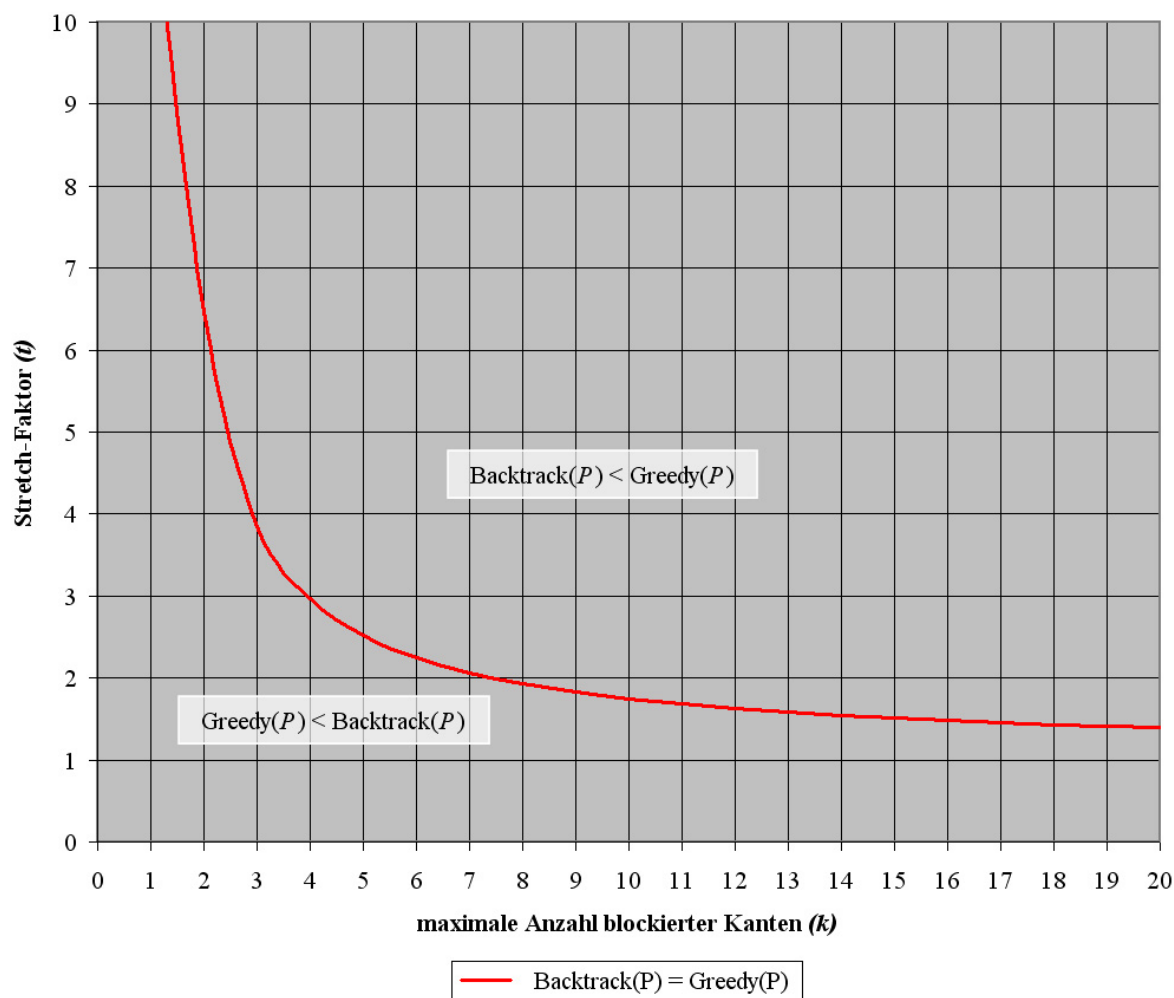


Abbildung 3.22: Kompetitivität von *Backtrack* und *Greedy* in Abhängigkeit von t und k

- t und k erfüllen *nicht* die Ungleichung (*)
 $\Rightarrow$ Benutze die *Backtrack*-Strategie.

Der Funktionsgraph in Abbildung 3.23 repräsentiert genau die Werte von t und k , für die *Backtrack* und *Greedy* die gleichen kompetitiven Faktoren haben. Die *Greedy*-Strategie ist also für alle Kombinationen von t und k , die unterhalb des Funktionsgraphen liegen, zu präferieren. Oberhalb des Graphen ist jedoch die *Backtrack*-Strategie die bessere Wahl. Für kleine k -Werte besteht also durchaus die realistische Chance, dass *Greedy* im worst-



Abbildungung 3.23: Vergleich der kompetitiven Faktoren von *Backtrack* und *Greedy*

case schneller ist als *Backtrack*, zumindest dann, wenn ein relativ kleiner Stretch-Faktor für den Restgraph garantiert werden kann. Ist beispielsweise bekannt, dass aus einem Graphen ausschließlich Kanten entfernt werden, so dass sich die Länge des kürzesten Pfades zwischen einem beliebigen Knotenpaar höchstens verdoppelt, so ist *Greedy* im worst-case für alle $k \leq 7$ schneller als *Backtrack*. Für $k \rightarrow \infty$ strebt der maximal zulässige Stretch-Faktor, für den *Greedy* im worst-case noch schneller ist als *Backtrack*, gegen 1. Für $k = 1$ hingegen

strebt der maximal zulässige Stretch-Faktor erwartungsgemäß gegen ∞ . Schließlich wissen wir aus Kapitel 3.2, dass die kompetitiven Faktoren von *Backtrack* und *Greedy* identisch sind, wenn höchstens eine Kante aus dem Ursprungsgraphen entfernt werden darf.

3.4.2 (t, l) -Self-Spanner

In diesem Abschnitt betrachten wir die Situation aus einem anderen Blickwinkel. Sind wir beim t -Spanner davon ausgegangen, dass wir nur gewisse Teilgraphen betrachten, die unseren Reglementierungen entsprechen, betrachten wir nun eine verwandte Graphklasse, die sich jedoch dadurch auszeichnet, dass sie bereits durch ihre Definition über ein gefordertes Maß an Robustheit verfügt.

Unter einem (t, l) -Self-Spanner verstehen wir dabei einen Graphen, bei dem bis zu l beliebige Kanten ausfallen dürfen und dabei ein Umweg maximal die t -fachen Ursprungskosten verursacht. Formal bedeutet dies:

Definition 3.25. Gegeben sei ein ungerichteter Graph $G = (V, E)$, eine reelle Zahl $t \geq 1$ und eine nicht-negative Zahl $l \in \mathbb{N}$. G ist ein **(t, l) -Self-Spanner**, wenn für jeden Teilgraphen $G' = (V, E')$ von G mit $|E'| \geq |E| - l$ und $E' \subseteq E$ gilt:

$$d_{G'}(u, v) \leq t \cdot d_G(u, v), \quad \text{für alle } u, v \in V, \text{ die in } G' \text{ über einen Pfad verbunden sind.}$$

Den kleinsten Parameter t , so dass obige Ungleichung gilt, bezeichnen wir wieder als *Stretch-Faktor*.

Wir werden in den folgenden Abschnitten untersuchen, inwieweit die Einschränkung auf die Klasse der (t, l) -Self-Spanner, mit $t \geq 1$, die Güte der Strategien *Backtrack* und *Greedy* verbessern kann.

Die Strategie Backtrack

Bei einem Stretch-Faktor $t = 1$ sind keine Umwege zwischen zwei Knoten $u, v \in V$ zulässig und eine mehrdeutige Strategie *Backtrack* kann nicht zu Umwegen gezwungen werden. Es lässt sich jedoch auf Grund der Mehrdeutigkeit dieser Strategie nicht beweisen, dass sich *Backtrack* in diesem Fall stets optimal verhält:

Lemma 3.26. Jede *Backtrack*-Strategie ist auf $(1, l)$ -Self-Spannern $(2k + 1)$ -kompetitiv. Jede eindeutige *Backtrack*-Strategie ist zudem nicht c -kompetitiv für alle $c < 2k + 1$. Es existiert jedoch zu jedem Szenario $P \in \Pi$ eine eindeutige Graph-Strategie, so dass $\text{Backtrack}(P) = \text{OPT}(P)$ gilt.

Beweis. Gegeben sei ein ungerichteter, gewichteter Graph $G = (V, E)$, eine Konstante $k \in \mathbb{N}$, die die maximale Anzahl an ausfallenden Kanten angibt, und feste Start- und Zielknoten $s, t \in V$. Damit *Backtrack* zu einem kompetitiven Faktor > 1 gezwungen werden kann, muss die Strategie mindestens einmal zur Umkehr zum Startknoten s gezwungen

werden. Dies ist jedoch bei einer mehrdeutigen Strategie nicht möglich, da ein Gegenspieler keine Kenntnis über den tatsächlich gegangenen Weg hat. Angenommen ein Gegenspieler kennt die angewandte tie-breaking-rule nicht und will dennoch die Strategie *Backtrack* an einem Knoten $v \in V$ durch eine Blockade einer an diesem Knoten inzidenten Kante $e \in E$ zur Rückkehr zwingen, so entsteht folgende Problematik: Wir wollen annehmen, dass *Backtrack* direkt vor Ankunft am Knoten v den Knoten $u \in V$ besucht hat. Dann gibt es am Knoten v die ausgehende Kante $\{u, v\}$, über die *Backtrack* bereits gelaufen ist, und es existieren n weitere ausgehende Kanten $\{v, v_i\} \in E$. Er weiß nun, dass über eine dieser Kanten ein kürzester Pfad zum Zielknoten t führt. Gibt es mehrere solcher Kanten, über die ein kürzester Weg von v zu t führt, so weiß er nicht, welche er blockieren muss, um *Backtrack* zur Rückkehr zum Startknoten s zu zwingen. Angenommen es gibt m viele solche Kanten: Wir gehen davon aus, dass auch noch mindestens m viele durch einen Gegenspieler blockiert werden dürfen. Wäre dies nicht der Fall, ließe sich erst recht nicht garantieren, dass *Backtrack* zur Umkehr gezwungen wird. Angenommen es gibt m viele Kanten und ein Gegenspieler dürfte mindestens m viele blockieren, so müsste er alle m Kanten blockieren, über die ein kürzester Pfad von v zu t führt. Es sei $G' = (V, E')$ mit $E' \subseteq E$ der Pfad, der entsteht, wenn alle Kanten am Knoten v , über die ein kürzester Weg zum Knoten t führt, entfernt wurden. Dann muss jedoch ein Knoten $v_i \in V$ existieren, für den $d_{G'}(v, v_i) > 1 \cdot d_G(v, v_i)$ gilt. Gäbe es diesen Knoten v_i nicht, wäre t von s aus in G' nicht mehr erreichbar, was wir eingangs ausgeschlossen haben. Damit kann aber G kein $(1, l)$ -Self-Spanner sein.

Blockieren der Gegenspieler hingegen nur einen Teil der m Kanten so können er umgekehrt nicht garantieren, dass die mehrdeutige Strategie *Backtrack* sich gemäß ihrer tie-breaking-rule für eine der unblockierten Kanten zur Traversierung entschieden hat. Er kann also selbst bei einer unblockierten Kante, über die ein kürzester Weg von s nach t führt, nicht garantieren, dass *Backtrack* diesen verfolgt, wenn zugleich ein weiterer kürzester Weg über eine Kante e_i zwischen den Knoten u und v existiert, wobei die e_i blockiert ist. Aus diesem Grund lässt sich auch keine Optimalität garantieren. Es gilt daher die obere Schranke des allgemeinen Graphen von $2k + 1$.

Im Falle mehrerer gleich langer kürzester Wege, kann ein Determinismus nur dadurch erreicht werden, dass eine tie-breaking-rule eingeführt wird. Diese lässt sich als Funktion darstellen, die beispielsweise die Elemente der Kantenmenge aufsteigend durchnummeriert. Wir wollen diese Kantennummerierung als Ordnungszahl bezeichnen. Falls nun zu einem Zeitpunkt der Berechnung mehrere kürzeste Wege berechnet wurden, könnte man einen Determinismus folgendermaßen garantieren: man traversiert stets den kürzesten Pfad dessen Ordnungszahl der ersten Kante unter allen ersten Kanten aller kürzesten Pfade minimal ist. Da eine tie-breaking-rule beliebig permutierbar ist, existiert stets eine eindeutige Graphstrategie aus der Menge aller *Backtrack*-Strategien, die das Problem optimal lösen.

□

Für höhere Werte von t lässt sich jedoch sehr wohl ein Szenario konstruieren, das die Strategie zu Umwegen zwingt. Wir betrachten erneut einen ungerichteten gewichteten Graphen $G = (V, E)$, eine Konstante $k \in \mathbb{N}$, die die Anzahl maximal ausfallender Kanten angibt, sowie zwei Knoten $s, t \in V$, die einen festen Start- sowie Zielknoten darstellen. Im Folgenden werden wir zeigen, welche Auswirkungen die Einschränkung auf (t, l) -Self-Spanner für die untere Schranke der Strategie *Backtrack* nach sich ziehen und betrachten zur Begründung die Konstruktion aus Abbildung 3.24. Es sei die Anzahl $l \in \mathbb{N}$ an maximal ausfallenden Kanten gemäß Definition 3.25 fest vorgegeben. Wir möchten an dieser Stelle noch einmal bemerken, dass die Kardinalität l der Ausfallmenge nicht von der maximalen Kantenausfallzahl k abhängt. Der Wert von l ist lediglich beschränkt durch die Kantenzahl $|E|$ des betrachteten Graphen G . Für jedes $k \in \mathbb{N}$ existiert eine Probleminstanz $P_k \in \Pi$, die ein worst-case-Szenario für die Strategie *Backtrack* repräsentiert. Für alle $k \in \mathbb{N}$, kon-

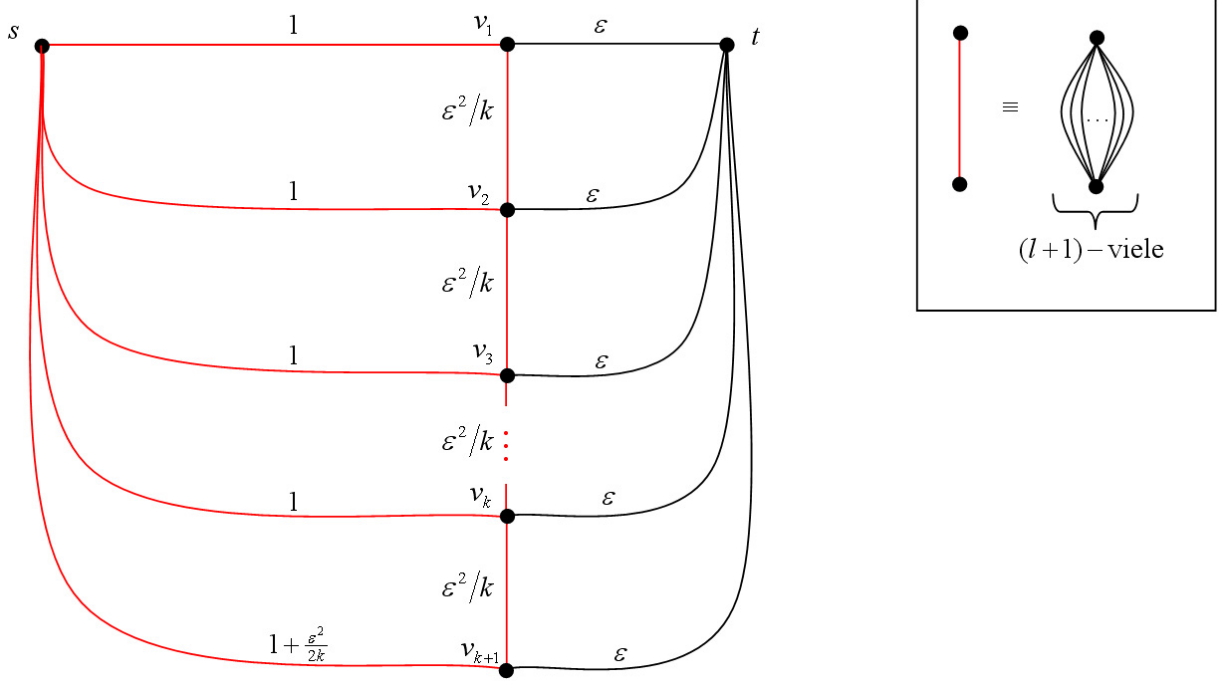


Abbildung 3.24: worst-case-Szenario für die *Backtrack*-Strategie auf (t, l) -Self-Spannern

struieren wir nachfolgend eine konkrete Probleminstanz P_k . Als Orientierung dient das worst-case-Szenario des allgemeinen Graphen, an dem wir einige Änderungen vornehmen müssen. Wir verwenden dazu am Startknoten s ausgehende k -viele roten Kanten $\{s, v_i\}$, mit $|\{s, v_i\}| := 1$ für alle $i \in \{1, \dots, k\}$. Alle diese Kanten $\{s, v_i\}$ werden durch eine Kante $\{v_i, t\}$, die mit $|\{v_i, t\}| := \varepsilon$ gewichtet ist, zu einem s - t -Pfad der Länge $|\{s, v_i, t\}| = 1 + \varepsilon$ mit $i \in \{1, \dots, k\}$ verlängert. Darüber hinaus verwenden wir noch eine weitere rote Kante $\{s, v_{k+1}\}$, die mit $|\{s, v_{k+1}\}| := 1 + \frac{\varepsilon^2}{2k} > 1$ gewichtet ist. Desweiteren fügen wir einen einfachen Pfad $(v_1, \dots, v_{k+1})$ aus roten Kanten $e_i := \{v_i, v_{i+1}\}$, die mit $|e_i| := \frac{\varepsilon^2}{k}$ gewichtet sind,

hinzu. Die roten Kanten werden, wie in Abbildung 3.24 angedeutet, durch $(l + 1)$ -viele schwarze Kanten mit jeweils identischen Kostenmaßen bezüglich der roten Kante ersetzt. Sie dienen ausschließlich der Übersicht, die bei der verwendeten Anzahl von Kanten in einer Zeichnung nicht mehr einzuhalten wäre. Von jedem Knoten v_i führt, wie zuvor bei der Konstruktion zur unteren Schranke für allgemeine Graphen, eine Kante $e_{k+1+i} = \{v_i, t\}$, mit Kosten $|e_{k+1+i}| := \varepsilon$, mit $i \in \{1, \dots, k + 1\}$, zum Zielknoten t . Insgesamt ist gewährleistet, dass stets l -viele Kanten, die nicht inzident zum Zielknoten t sind, ausfallen können ohne, dass hierdurch ein Umweg entsteht. Betrachten wir nun die Kosten des größten Umweges zwischen zwei Knoten $u, v \in V$, die bei einem Ausfall von l beliebigen Kanten $e \in E$ entstehen können: Diese entstehen für den Graphen G z. B. zwischen den Knoten v_1 und t , wodurch wir bei deren Ausfall den Graphen $G' = (V, E \setminus \{\{v_1, t\}, \dots, \{v_k, t\}\})$ erhalten. Sollte $l > k$ sein, so hat dies auf den größten Umweg zwischen zwei Knoten keine Auswirkungen, da im Graphen G' , so wie er oben angegeben ist, nur noch rote Kanten und die Kante $\{v_{k+1}, t\}$ nicht blockiert sind. Rote Kanten sind jedoch stets $(l + 1)$ -fach vorhanden und durch die Blockade der Kante $\{v_{k+1}, t\}$ würden Start- und Zielknoten in unterschiedlichen Zusammenhangskomponenten liegen. Dann sind die beiden zu betrachtenden Knoten im resultierenden Graphen aber nicht mehr miteinander verbunden und müssen für die Berechnung des Stretch-Faktors nicht berücksichtigt werden. In einem solchen Teilgraphen $G'' := (V, E \setminus \bigcup_{1 \leq i \leq k+1} \{v_i, t\})$ ist der Stretch-Faktor $t = 1$. Für den Stretch-Faktor t in unserem Szenario lässt sich nun folgender Wert ermitteln:

$$\begin{aligned} t &\leq \max\left\{ \frac{d_{G'}(u, v)}{d_G(u, v)} \mid u, v \in V, u \neq v, \text{ und } G' = (V, E') \text{ mit } E' \subseteq E \right\} \\ &= \frac{d_{G'}(v_1, t)}{d_G(v_1, t)} = \frac{k \cdot \frac{\varepsilon^2}{k} + \varepsilon}{\varepsilon} = 1 + \varepsilon. \end{aligned}$$

Durch die obige Konstruktion haben wir einen Graphen gefunden, der für ein beliebiges $k \in \mathbb{N}$ einen $((1 + \varepsilon), l)$ -Self-Spanner darstellt, mit $\varepsilon > 0$.

Betrachten wir nun die *Backtrack*-Strategie auf diesem Graphen G_k , so ist zu bemerken, dass zu keinem Zeitpunkt ein kürzester Weg vom Startknoten s zum Zielknoten t über eine mit $\frac{\varepsilon^2}{k}$ gewichtete Kante verläuft. Das vorliegende Szenario P_k ähnelt sehr stark dem worst-case-Szenario für allgemeine Graphen aus Kapitel 3.2.1. Die *Backtrack*-Strategie versucht sukzessive alle kürzesten Wege vom Startknoten s zum Zielknoten t zu traversieren. Dabei muss sie sich aufgrund der Konstruktion am Startknoten s k -mal für eine der k -vielen roten Kanten entscheiden, die allesamt mit identischen Kosten 1 gewichtet und zu den Knoten v_i , mit $i \in \{1, \dots, k + 1\}$, inzident sind. Da alle Pfade mit identischen Kosten gewichtet sind, ist ein kürzester Pfad nicht eindeutig. Jeder dieser roten Pfade repräsentiert $l + 1$ schwarze Pfade. Um an dieser Stelle eine eindeutige Strategie zu erhalten könnte man eine tie-breaking-rule einführen, die mittels einer injektiven Abbildung $\varphi : E \rightarrow \mathbb{N}$ jeder Kante $e_i \in E$ eine eindeutige Ordnungszahl zuweist. Dadurch wird ein Determinismus garantiert, der die Strategie eindeutig werden lässt. Die Strategie könnte dann nämlich z. B. im Falle von mehreren kürzesten Wegen die Kante mit der kleinsten

Ordnungszahl traversieren. Für das worst-case-Szenario ist dies jedoch unerheblich. Ein Gegenspieler blockiert im schlimmsten Fall die Kanten $\{v_1, t\}, \dots, \{v_k, t\}$. Die *Backtrack*-Strategie traversiert k mal einen Repräsentanten einer roten Kante und entdeckt dabei an den Knoten v_i , dass die Kante $\{v_i, t\}$ entfernt ist. Dadurch wird sie zum Rückweg zum Startknoten s gezwungen, von wo aus sie weitere rote Kanten $\{s, v_i\}$ mit $i \in \{1, \dots, k\}$ durch Wahl eines Repräsentanten ausprobiert. Nachdem sie alle k roten Kanten traversiert hat, ist lediglich der Pfad (s, v_{k+1}, t) unblockiert und stellt außerdem mit Kosten $|(s, v_{k+1}, t)| := 1 + \frac{\varepsilon^2}{2k} = \text{OPT}(P_k)$ die Kosten einer optimalen Lösung dar. Es lässt sich nun folgende Kostenabschätzung durchführen:

$$\begin{aligned} \forall k \in \mathbb{N} \exists P_k \in \Pi : \frac{\text{Backtrack}(P_k)}{\text{OPT}(P_k)} &= \frac{2 \sum_{i=1}^k 1 + 1 + \frac{\varepsilon^2}{2k} + \varepsilon}{1 + \frac{\varepsilon^2}{2k} + \varepsilon} \\ &= \frac{2k + 1 + \frac{\varepsilon^2}{2k} + \varepsilon}{1 + \frac{\varepsilon^2}{2k} + \varepsilon}. \end{aligned}$$

Da wir durch Skalierung des Graphen den optimalen Weg beliebig groß wählen können, ergibt sich nach Lemma 3.1 eine untere Schranke für einen kompetitiven Faktor auf $((1 + \varepsilon), l)$ -Self-Spannern von

$$\frac{2k + 1 + \frac{\varepsilon^2}{2k} + \varepsilon}{1 + \frac{\varepsilon^2}{2k} + \varepsilon}.$$

Dadurch ergibt sich mit der oberen Schranke für die *Backtrack*-Strategie auf allgemeinen Graphen aus Kapitel 3.2.1 folgender Satz:

Satz 3.27. *Die Strategie Backtrack ist für $\varepsilon > 0$ auf $((1 + \varepsilon), l)$ -Self-Spannern $(2k + 1)$ -kompetitiv und nicht c -kompetitiv für*

$$c < \frac{2k + 1 + \frac{\varepsilon^2}{2k} + \varepsilon}{1 + \frac{\varepsilon^2}{2k} + \varepsilon}.$$

Die Strategie Greedy

Gegenstand der Betrachtung ist erneut das k -Canadian Traveller Problem. Wir untersuchen in diesem Abschnitt, inwieweit die Graphklasse der (t, l) -Self-Spanner die Bedingungen für einen Gegenspieler einschränkt, der gegen die Strategie *Greedy* spielt, und werden dabei die enge Verbindung zwischen t -Spannern und (t, l) -Self-Spannern untermauern. Wir werden nachfolgend zeigen, dass sich die Ergebnisse des t -Spanners auf den (t, l) -Self-Spanner übertragen lassen, indem wir die untere Schranke für t -Spanner durch leichte Modifikation zu einer unteren Schranke für (t, l) -Self-Spanner transformieren. Dabei nutzen wir folgende Tatsache aus:

Ist $G = (V, E)$ ein (t, l) -Self-Spanner, mit $l \in \mathbb{N}$, so ist $G' = (V, E')$ ein t -Spanner, für alle $E' \subseteq E$, mit $|E'| \geq |E| - l$.

Für jedes $l \geq k$, wobei $k \in \mathbb{N}$, die Anzahl maximal ausfallender Kanten bezüglich des k -Canadian Traveller Problem angibt, lassen sich die (t, l) -Self-Spanner als Teilmenge der t -Spanner betrachten.

Zur Erinnerung: Ein Graph $G' \subseteq G$, mit $G' = (V, E')$, $G = (V, E)$ und $E' \subseteq E$, wird als t -Spanner bezeichnet, wenn die Distanz eines beliebigen Knoten $u, v \in V$, die in G' verbunden sind, die t -fache Distanz im Ursprungsgraphen G nicht überschreitet. Da die betrachteten Teilgraphen $G' \subseteq G$ durch Entfernen von maximal k Kanten entstehen, gewährleistet die Forderung $l \geq k$, dass *jeder* solche Teilgraph G' von G zugleich t -Spanner von G ist.

Um für jedes $k \in \mathbb{N}$ eine eigene Probleminstanz $P_k \in \Pi$ zu konstruieren, betrachten wir Abbildung 3.25, die im Wesentlichen die Konstruktion unseres worst-case-Graphen aus Abschnitt 3.4.1 zu t -Spannern darstellt. Um den Graphen aus Abbildung 3.17 jedoch in einen (t, l) -Self-Spanner zu transformieren, müssen wir garantieren, dass es möglich ist, beliebige l Kanten aus G zu entfernen, so dass die Kosten eines kürzesten Pfades zwischen zwei Knoten $u, v \in V$ ein t -faches bezüglich ihrer Distanz im Ursprungsgraphen nicht übersteigen, sofern sie jetzt immer noch über einen Pfad verbunden sind. Diesem Problem können elegant aus dem Weg gehen, indem wir an einigen Stellen $(l + 1)$ -viele Kopien einer Kante mit identischer Gewichtung verwenden. Diese Maßnahme treffen wir für jede Kante auf dem Pfad von *Greedy*, sowie dem einer optimalen Lösung. Zur besseren Übersicht benutzen wir in unserer Konstruktionsabbildung 3.25 rote Kanten, die erneut $(l + 1)$ -viele schwarze Kanten mit identischer Gewichtung repräsentieren. Somit wird der Weg, den die *Greedy*-Strategie verfolgt, nicht unnötig durch die Konstruktionsbedingungen für einen (t, l) -Self-Spanner in den Hintergrund gerückt. Es kann nun garantiert werden, dass sich zwischen zwei Knoten auf einem dieser beiden Pfade kein Umweg erzeugen lässt, wenn l beliebige Kanten ausfallen. Dies gilt auch für die Kanten $\{v_i, t\}$, mit $i \in \{1, \dots, k\}$, die durch den Gegenspieler entfernt werden. Die Konstruktion ist so ausgelegt, dass die Kosten für die Traversierung des Pfades $(v_i, v_{i-1}, \dots, v_1, s, t)$ das t -fache von $|\{v_i, t\}|$ nicht übersteigen mit $i \in \{1, \dots, k\}$. Die exakte Argumentation ist analog zu der für die Konstruktion der unteren Schranke auf t -Spannern und findet sich in Abschnitt 3.4.1 ab Seite 65.

Es wäre an dieser Stelle außerdem nicht sinnvoll, die Kanten zwischen den Knoten v_i und t , mit $i \in \{1, \dots, k+1\}$, durch $l+1$ Hilfskanten selbiger Gewichtung zu versehen, da sich dies zu Ungunsten der Strategie eines Gegenspielers auswirken würde. Die Begründung ist darin zu finden, dass alle $\{v_i, t\}$ -Kanten blockiert werden müssen, um die Greedy-Strategie daran zu hindern, einen kürzesten Pfad über eine unblockierte $\{v_i, t\}$ -Kante zum Zielknoten t zu nehmen. Eine Blockade von mehr als einer $\{v_i, t\}$ -Kante an einem Knoten v_i verringert jedoch die Anzahl maximal möglicher Umwege zu weiteren Knoten v_j . Dies verringert den maximalen Gesamtumweg und somit den maximal erreichbaren kompetitiven Faktor. Einen Sonderfall dieser Überlegung werden wir jedoch am Ende dieses Abschnittes zu einer alternativen Konstruktion eines worst-case-Szenarios betrachten.

Wir wählen als Kosten für den optimalen Weg $\text{OPT}(P_k) := 1$. Dies ist die rote Kante $\{s, t\}$, mit $|\{s, t\}| = 1$ aus Abbildung 3.25, die den Startknoten s und den Zielknoten t direkt miteinander verbindet. In der Absicht einen möglichst großen Umweg für *Greedy* zu erzeugen, konstruieren wir ein Pfadsegment vom Knoten s zu einem Knoten v_1 , an dem *Greedy* die erste ausgefallene Kante $\{v_1, t\}$ bemerken wird. Für diese gilt $|\{v_1, t\}| =: x_1$. Die beiden Pfadsegmente zusammen ergeben einen möglichen s - t -Pfad, der, um von *Greedy* gewählt zu werden, echt kleiner als die optimale Lösung vom Startknoten s zum Zielknoten t sein muss. Wir müssen garantieren, dass $|(s, v_1, t)| < 1 = \text{OPT}(P_k)$ gilt. Somit wählen wir die Kosten für die Kante $|\{s, v_1\}| = 1 - x_1 - \varepsilon$ für ein $\varepsilon > 0$. Dadurch ist sichergestellt, dass der Weg kürzer als der letztlich optimale Weg ist und somit von *Greedy* traversiert wird. Da ein Gegenspieler die Kante $\{v_1, t\}$ entfernt, muss sichergestellt sein, dass ein Pfad von v_1 zu t existiert, der Kosten $|(v_1, t)| \leq t \cdot x_1$ garantiert, um die (t, l) -Self-Spanner-Eigenschaft zu erfüllen. Dies ist durch den Pfad (v_1, s, t) gegeben. Dieser Pfad ist stets verfügbar, da $l + 1$ -fach vorkommt. Es ist daher durch Blockade von maximal l Kanten nicht möglich diesen Pfad einzuschränken. *Greedy* wird also im ersten Schritt gemäß unserer Konstruktion am Knoten v_1 ankommen und die blockierte Kante $\{v_1, t\}$ entdecken. Die Strategie berechnet nun den kürzesten v_1 - t -Pfad und folgt diesem bis sie entweder t erreicht oder an einem Knoten v_2 eine weitere blockierte Kante $\{v_2, t\}$ entdeckt. Der angesprochene v_1 - t -Pfad muss also echt kleiner als die Summe der Kosten für den Rückweg zu s und die Kosten $\text{OPT}(P_k)$ für den optimalen Weg sein. Es muss nun im Sinne einer Maximierung für die Kosten eines worst-case-Szenarios der Strategie *Greedy* $|(v_1, v_2, t)| < |(v_1, s, t)|$ gelten. Die Konstruktion kann nach diesem Schema beliebig erweitert werden. Wir haben somit unser worst-case-Szenario für t -Spanner aus Kapitel 3.4.1 auf einem Graphen G angegeben, welches nun außerdem die Eigenschaften eines (t, l) -Self-Spanners erfüllt.

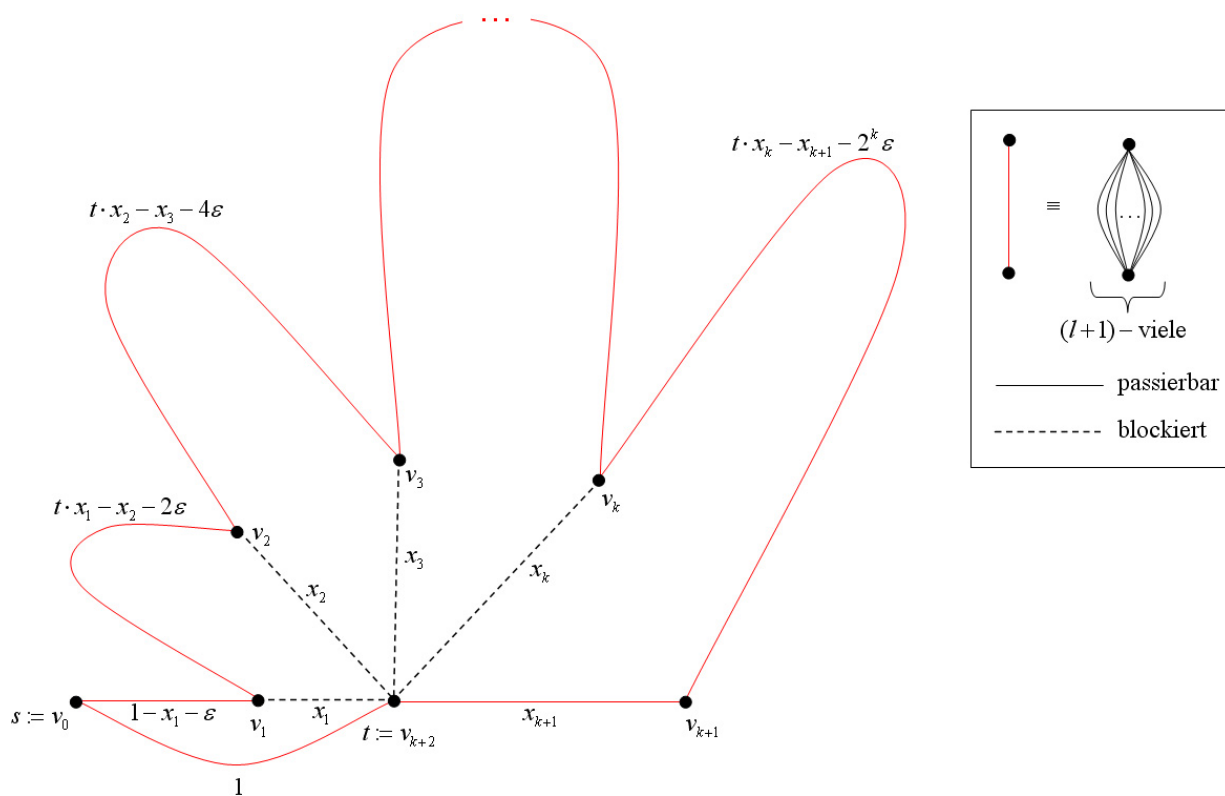
Die Strategie eines Gegenspielers kann nun die Strategie *Greedy* dazu zwingen, den Pfad $(s, v_1, \dots, v_{k+1}, t)$ zu traversieren. Es wurden somit alle gestrichelten Kanten $\{\{v_1, t\}, \dots, \{v_k, t\}\}$ mit $|\{v_i, t\}| = x_i$ aus Abbildung 3.25 blockiert, während sich ein optimaler s - t -Pfad mit Kosten 1 ergibt. Die exakten Kosten der Strategie *Greedy* ergeben sich aus der Wahl der Längen für die x_i . Somit gilt für den kompetitiven Faktor dieselbe untere Schranke wie auf t -Spannern. Da der (t, l) -Self-Spanner für $l \geq k$ überdies ein Sonderfall des t -Spanners ist, gilt die obere Schranke aus Satz 3.19 für t -Spanner ebenfalls für (t, l) -Self-Spanner. Aus beiden Aussagen ergibt sich:

Korollar 3.28. *Ist G ein (t, l) -Self-Spanner, so ist die Strategie *Greedy* auf G für alle $t \geq 1$ und für alle $l \geq k$ c -kompetitiv mit*

$$c = 2 \left(\frac{2t}{t+1} \right)^k - 1$$

und nicht c' -kompetitiv für alle $c' < c$.

Zusätzlich zur soeben vorgestellten unteren Schranke wollen wir eine alternative Konstruktion angeben, die die untere Schranke des kompetitiven Faktors der Strategie *Greedy*

Abbildung 3.25: worst-case-Szenario für die *Greedy*-Strategie auf (t, l) -Self-Spannern

auf (t, l) -Self-Spannern beeinflusst. Für sehr kleine Werte von l lässt sich nämlich ein Szenario angeben, in dem die untere Schranke für den kompetitiven Faktor gegenüber dem Ergebnis aus Korollar 3.28 deutlich angehoben wird. Die Differenz ist dabei antiproportional zum Stretch-Faktor t . Wir geben im Folgenden für jedes $k \in \mathbb{N}$ eine Instanz P_k an, die eine untere Schranke für die Strategie *Greedy* auf (t, l) -Self-Spannern darstellt. Hierzu betrachten wir Abbildung 3.26, die die Konstruktion verdeutlicht. Wir beschreiben nachfol-

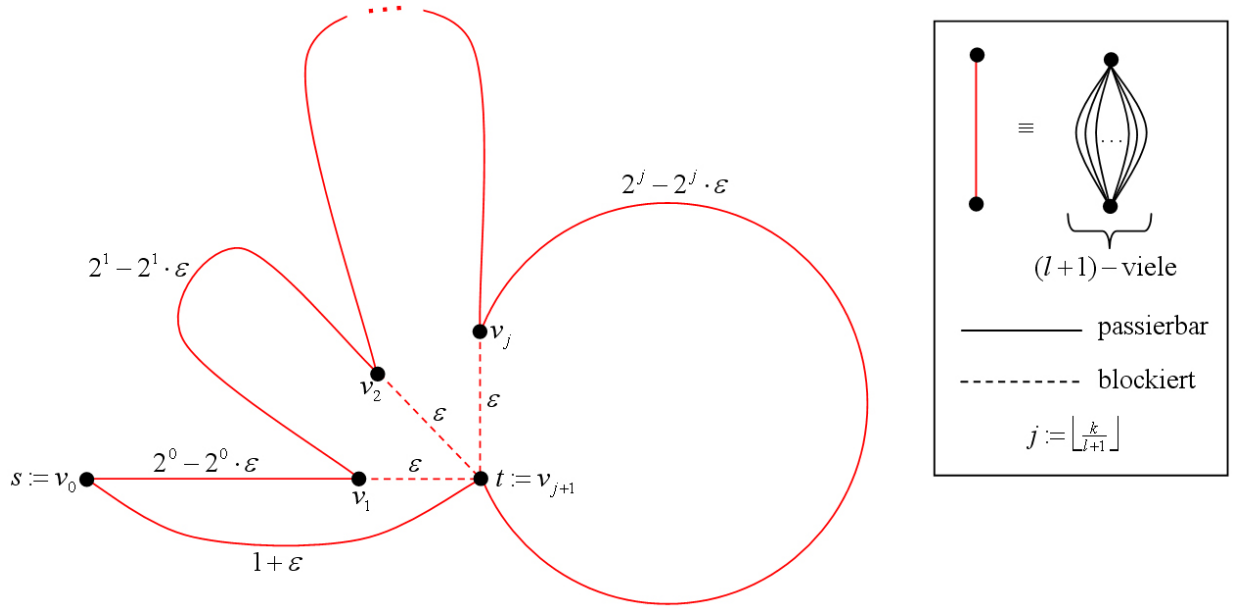


Abbildung 3.26: worst-case-Szenario für die *Greedy*-Strategie auf (t, l) -Self-Spannern (2)

gend den Aufbau eines ungerichteten, gewichteten Graphen $G = (V, E)$, zu einer gegebenen Konstante $k \in \mathbb{N}$ sowie eines Start- und Zielknotens $s, t \in V$. Dazu konstruieren wir eine Kante $\{s, t\}$ mit Gewichtung $|\{s, t\}| := 1 + \varepsilon$, die die *Greedy*-Strategie jedoch niemals betreten wird. Sie verfolgt nämlich den um $\varepsilon > 0$ kürzeren Pfad von s zu t , der aus den beiden roten Kanten $\{s, v_0\}$, mit $|\{s, v_0\}| := 2^0 - 2^0 \cdot \varepsilon$, und $\{v_0, t\}$, mit $|\{v_0, t\}| := \varepsilon$, besteht. Es existieren weitere Kanten vom Knoten v_i zum Knoten v_{i+1} , mit $|\{v_i, v_{i+1}\}| := 2^i - 2^i \cdot \varepsilon$, für alle $i \in \{1, \dots, \lfloor \frac{k}{l+1} \rfloor\}$. Es sei $s := v_0$ und $t := v_j$, mit $j := \lfloor \frac{k}{l+1} \rfloor + 1$. Die roten Kanten repräsentieren $(l+1)$ schwarze Kanten und dienen ausschließlich der Übersichtlichkeit. Ein Gegenspieler blockiert nun alle Kanten $\{v_i, t\}$, mit $i \in \{1, \dots, j\}$. Dadurch wird die *Greedy*-Strategie gezwungen den Pfad $(s, v_1, \dots, v_j, t)$ zu traversieren und verursacht dabei Kosten in Höhe von $\text{Greedy}(P_k) := \sum_{i=0}^{\lfloor \frac{k}{l+1} \rfloor} (2^i - 2^i \cdot \varepsilon)$. Die optimalen Kosten betragen jedoch nur

$\text{OPT}(P_k) := 1 + \varepsilon$. Damit ergibt sich folgende Kostenabschätzung:

$$\begin{aligned} \forall k \in \mathbb{N} \exists P_k \in \Pi : \frac{\text{Greedy}(P_k)}{\text{OPT}(P_k)} &= \frac{\sum_{i=0}^{\lfloor \frac{k}{l+1} \rfloor} (2^i - 2^i \cdot \varepsilon)}{1 + \varepsilon} \\ &= \frac{2^{\lfloor \frac{k}{l+1} \rfloor + 1} - 1 - (2^{\lfloor \frac{k}{l+1} \rfloor + 1} - 1) \cdot \varepsilon}{1 + \varepsilon} \end{aligned}$$

Dieser Term kann nun so umgeformt werden, dass für jedes $\delta > 0$ ein $\varepsilon > 0$ existiert, so dass gilt:

$$\begin{aligned} \forall k \in \mathbb{N} \exists P_k \in \Pi : \frac{\text{Greedy}(P_k)}{\text{OPT}(P_k)} &= \frac{2^{\lfloor \frac{k}{l+1} \rfloor + 1} - 1 - (2^{\lfloor \frac{k}{l+1} \rfloor + 1} - 1) \cdot \varepsilon}{1 + \varepsilon} \\ &\geq 2^{\lfloor \frac{k}{l+1} \rfloor + 1} - 1 - \delta. \end{aligned}$$

Da wir den Graphen G beliebig skalieren können, lässt sich ein beliebig hoher Wert für die optimalen Kosten erreichen. Damit gilt nach Lemma 3.1 eine untere Schranke für den kompetitiven Faktor von $2^{\lfloor \frac{k}{l+1} \rfloor + 1} - 1 - \delta$. Da $\delta > 0$ darüber hinaus beliebig nah an Null wählbar ist, gilt sogar eine untere Schranke von $2^{\lfloor \frac{k}{l+1} \rfloor + 1} - 1$. Wir erhalten somit für $l = 0$ den zu erwartenden Wert der unteren Schranke auf allgemeinen Graphen von $2^{k+1} - 1$. Dann ergibt sich mit der oberen Schranke der Strategie *Greedy* auf allgemeinen Graphen folgendes Korollar:

Korollar 3.29. *Ist G ein (t, l) -Self-Spanner, mit $l < k$, so ist die Strategie *Greedy* $(2^{k+1} - 1)$ -kompetitiv und nicht besser als c -kompetitiv, mit*

$$c = \max \left\{ 2 \left(\frac{2t}{t+1} \right)^k - 1, \quad 2^{\lfloor \frac{k}{l+1} \rfloor + 1} - 1 \right\}.$$

Es stellt sich nun die Frage, für welche Werte l diese untere Schranke einen höheren kompetitiven Faktor als den in Korollar 3.28 auf Seite 100 liefert. Dazu formen wir die Gleichungen etwas um:

$$\begin{aligned} 2 \left(\frac{2t}{t+1} \right)^k - 1 &< 2^{\lfloor \frac{k}{l+1} \rfloor + 1} - 1 \\ \Rightarrow \log_2 \left(2 \left(\frac{2t}{t+1} \right)^k \right) &< \left\lfloor \frac{k}{l+1} \right\rfloor + 1. \end{aligned}$$

Sei nun $k := n \cdot (l + 1)$:

$$\begin{aligned}
 &\Rightarrow (l + 1) \cdot (\log_2 \left(2 \left(\frac{2t}{t+1} \right)^k \right) - 1) < k \\
 &\Rightarrow (l + 1) \cdot \left(\underbrace{\log_2(2)}_{=1} + \log_2 \left(\frac{2t}{t+1} \right) \cdot k - 1 \right) < k \\
 &\Rightarrow k \cdot (l + 1) \cdot (\log_2(2t) - \log_2(t + 1)) < k \\
 &\Rightarrow (l + 1) \cdot (\log_2(2t) - \log_2(t + 1)) < 1 \\
 &\Rightarrow l < \frac{1}{\log_2(2t) - \log_2(t + 1)} - 1 \\
 &\Rightarrow l < \frac{1}{1 + \log_2(t) - \log_2(t + 1)} - 1.
 \end{aligned}$$

Abbildung 3.27 stellt dies noch einmal anschaulicher dar. Um den maximalen Wert für t zu

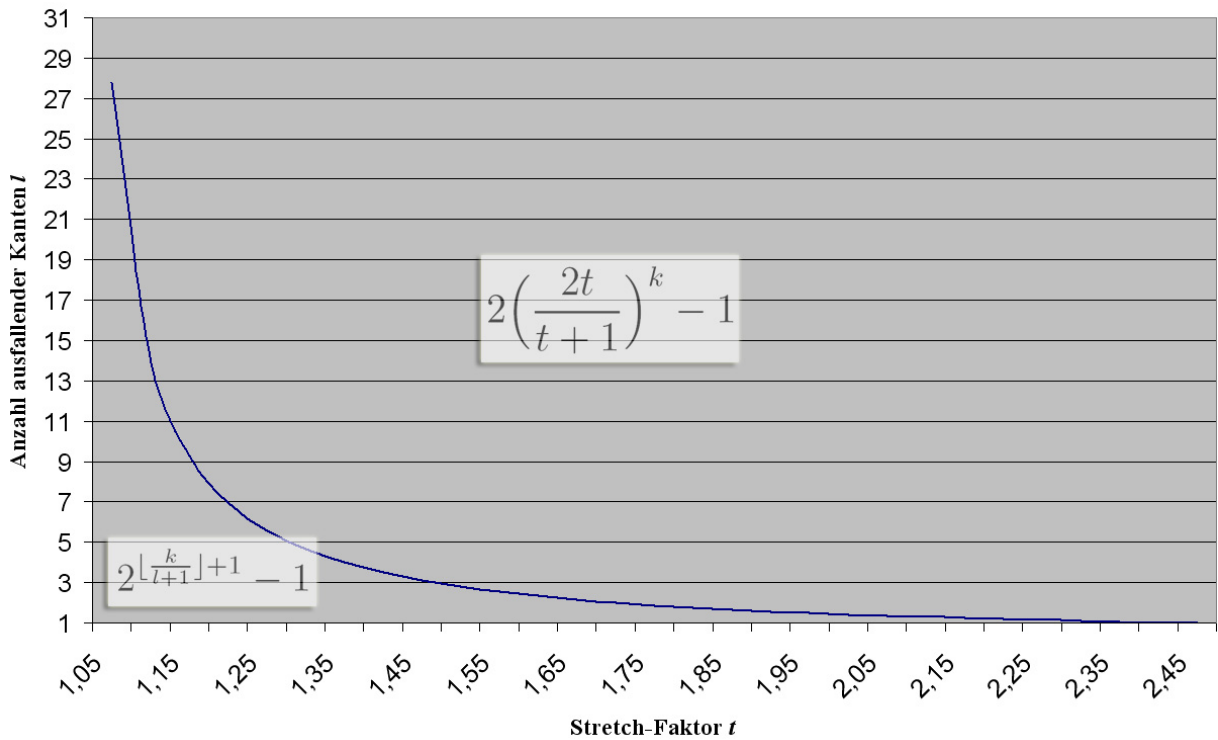


Abbildung 3.27: Gegenüberstellung beider Varianten für die untere Schranke der Strategie *Greedy*

berechnen, bei dem diese Schranke noch einen höheren kompetitiven Faktor liefert, formen wir unsere Gleichung erneut um.

Allgemein gilt:

$$\begin{aligned}
& 2\left(\frac{2t}{t+1}\right)^k - 1 < 2^{\lfloor \frac{k}{l+1} \rfloor + 1} - 1 \\
\Rightarrow & \frac{2t}{t+1} < \sqrt[k]{2^{\lfloor \frac{k}{l+1} \rfloor}} \\
\Rightarrow & t - \frac{t}{2} \cdot \sqrt[k]{2^{\lfloor \frac{k}{l+1} \rfloor}} < \frac{1}{2} \cdot \sqrt[k]{2^{\lfloor \frac{k}{l+1} \rfloor}} \\
\Rightarrow & t < \frac{\sqrt[k]{2^{\lfloor \frac{k}{l+1} \rfloor}}}{2 - \sqrt[k]{2^{\lfloor \frac{k}{l+1} \rfloor}}}.
\end{aligned}$$

Um eine bessere Vorstellung von diesem Wert zu bekommen berechnen wir exemplarisch für den speziellen Wert $l := 1$ den maximal möglichen Stretch-Faktor t , so dass sich für diese alternative Konstruktion eine höhere untere Schranke als in Korollar 3.28 auf Seite 100 ergibt. Für das feste $l = 1$ betrachten wir eine beliebige, jedoch gerade Anzahl $k := 2n$, für $n \in \mathbb{N}$, an ausfallenden Kanten. Durch diese nur sehr geringe Einschränkung lassen sich die unteren Gaußklammern ignorieren. Dann ergibt sich:

$$\begin{aligned}
t & < \frac{\sqrt[k]{2^{\lfloor \frac{k}{l+1} \rfloor}}}{2 - \sqrt[k]{2^{\lfloor \frac{k}{l+1} \rfloor}}} \stackrel{k=2n}{=} \frac{\sqrt{2}}{2 - \sqrt{2}} \\
& \approx 2,41421356.
\end{aligned}$$

Falls der Stretch-Faktor kleiner als $\frac{\sqrt{2}}{2-\sqrt{2}}$ ist, liefert das hier vorgestellte Szenario eine höhere untere Schranke für die Strategie *Greedy* auf $(t, 1)$ -Self-Spannern.

3.4.3 t -Self-Spanner

In diesem Abschnitt betrachten wir die Klasse der t -Self-Spanner, die eine enge Verwandtschaft mit den (t, l) -Self-Spannern aufweisen. Man kann sich einen t -Self-Spanner als (t, l) -Self-Spanner vorstellen, bei dem die maximale Ausfallgröße l an Kanten, die für den Stretch-Faktor t berücksichtigt werden muss genau der Kardinalität der Kantenmenge des Graphen entspricht. Es muss also bei einer beliebigen Ausfallmenge an Kanten gewährleistet sein, dass die Länge des kürzesten Weges zwischen zwei im Restgraphen noch über einen Pfad miteinander verbundenen Knoten das t -fache der Länge eines kürzesten Pfades im Ursprungsgraphen nicht übersteigt. Es ist keine triviale Aufgabe t -Self-Spanner mit kleinem t zu finden, so dass dieses Szenario ebenfalls als Grundlage für das k -Canadian Traveller Problem nutzbar ist, was in den folgenden Abschnitten verdeutlicht werden soll.

Definition 3.30. Gegeben sei ein ungerichteter Graph $G = (V, E)$ und eine reelle Zahl $t \geq 1$. G ist ein t -Self-Spanner, wenn für jeden Teilgraphen $G' = (V, E')$, mit $E' \subseteq E$ gilt:

$$d_{G'}(u, v) \leq t \cdot d_G(u, v), \text{ für alle } u, v \in V, \text{ die in } G' \text{ über einen Pfad verbunden sind.}$$

Den kleinsten Parameter t , so dass obige Ungleichung gilt, bezeichnen wir wieder als *Stretch-Faktor*.

Es lassen sich an dieser Stelle nicht die Ergebnisse des (t, l) -Self-Spanners aus Korollar 3.28 übertragen, da die Anzahl an maximal ausfallenden Kanten l im Vorhinein nicht bekannt ist.

Lemma 3.31. *Sei $G = (V, E)$ und $t \geq 1$. Wenn G ein t -Self-Spanner ist, dann hat jeder einfache Kreis C in G maximal $\lfloor t \rfloor + 1$ Kanten.*

Beweis. Sei $G = (V, E)$ ein ungerichteter, gewichteter t -Self-Spanner, mit $t \in \mathbb{R}_{\geq 1}$. Angenommen, es gäbe einen Kreis C mit mehr als $\lfloor t \rfloor + 1$ Kanten. Streiche alle Kanten aus G , die nicht zu C gehören und konstruiere so einen Graphen $G' = (V, E')$, mit $E' \subseteq E$. Wir entfernen nun auf diesem Kreis eine Kante mit minimalen Kosten. Dann wissen wir, dass die beiden inzidenten Knoten dieser soeben entfernten Kante noch über einen Pfad aus $t + 1$ Kanten miteinander verbunden sind. Da wir jedoch die Kante mit minimalen Kosten entfernt haben, muss die Überquerung jeder einzelnen Kante auf diesem Pfad mindestens die Kosten der Überquerung der entfernten Kante verursachen. Formaler betrachten wir: $G'' = (V, E' \setminus \{e_i\})$, mit $l := |e_i| = |\{u, v\}|$ und $l \leq |e_j|$ für alle $e_j \in E'$. Dann gilt:

$$d_{G''}(u, v) \geq (\lfloor t \rfloor + 1) \cdot l > t \cdot l \geq t \cdot d_G(u, v).$$

Dies ist jedoch ein Widerspruch zur Annahme, dass G ein t -Self-Spanner ist.

□

Die Strategie Backtrack

1-Self-Spanner

Wir bezeichnen das k -Canadian Traveller Problem mit erneut Π . Für jedes $k \in \mathbb{N}$ geben wir eine Instanz $P_k \in \Pi$ an, die ein worst-case-Szenario für *Backtrack* auf 1-Self-Spannern darstellt. Die Konstruktion einer Instanz P_k wie folgt: Es sei $G = (V, E)$ ein ungerichteter, gewichteter 1-Self-Spanner. In Abbildung 3.28 sehen wir ein Beispiel eines Graphen, der bis auf Mehrfachkanten zwischen zwei Knoten $u, v \in V$ nach Lemma 3.31 zyklensfrei sein muss. Existieren in G n Mehrfachkanten $e_i = \{u, v\}$, mit $i \in \{1, \dots, n\}$, zwischen zwei Knoten u und v , so gilt $|e_i| = |e_j|$, für alle $i, j \in \{1, \dots, n\}$. Auch ohne Mehrfachkanten, wie z. B. zwischen den Knoten s und v_1 in Abbildung 3.29 ließe sich durch Entfernen aller $m < n$ Kanten e_i , mit $|e_i| < |e_j|$ und $e_i = \{u, v\} = e_j$, für alle $i, j \in \{1, \dots, n\}$ ein Umweg zwischen den Knoten u und v erzeugen, so dass gilt $d_{G'}(u, v) > 1 \cdot d_G(u, v)$, womit G kein 1-Self-Spanner sein könnte. Bis auf Mehrfachkanten identischer Gewichtung zwischen zwei Knoten $u, v \in V$ haben wir es also mit einer Baumstruktur zu tun.

Definition 3.32. *Es sei $G = (V, E)$ ein ungerichteter Graph. Hat ein größter einfacher Kreis in G nicht mehr als 2 Kanten, so nennen wir G **quasi-zyklensfrei**.*

In zyklensfreien Graphen ist der kürzeste Weg von einem Startknoten $s \in V$ zu einem Zielknoten $t \in V$ stets eindeutig. In quasi-zyklensfreien Graphen ist dies nicht mehr

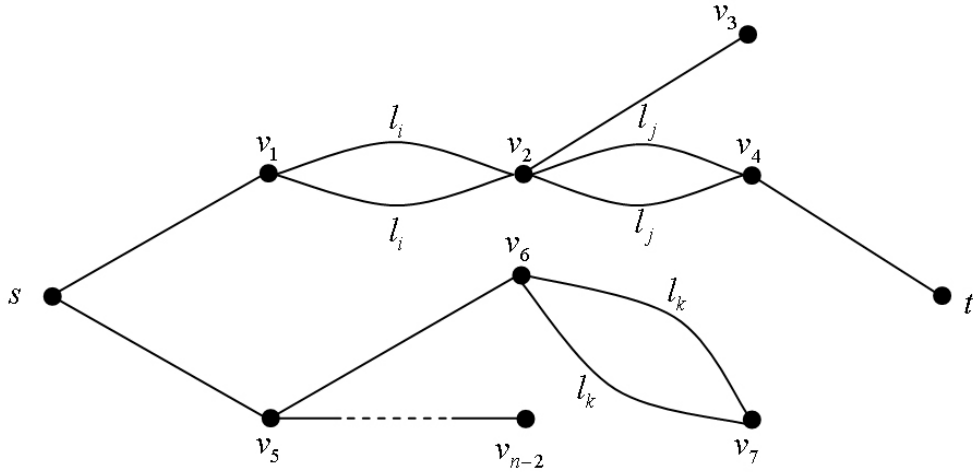
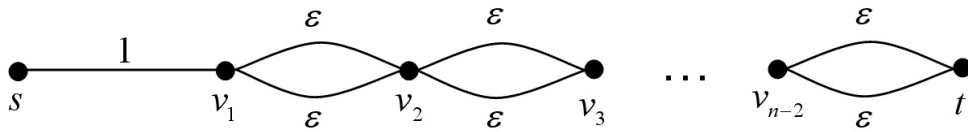


Abbildung 3.28: Beispiel eines 1-Self-Spanners

gewährleistet, da Mehrfachkanten zwischen zwei Knoten u und v existieren können. Der Graph G aus Abbildung 3.28 ist quasi-zyklenfrei. Wir betrachten zunächst zwei Knoten $u, v \in V$ zwischen denen Mehrfachkanten existieren. Wie bereits oben erwähnt, müssen alle diese Kanten identisch gewichtet sein.

Entfernt ein Gegenspieler nur eine echte Teilmenge dieser Mehrfachkanten, so wählt die mehrdeutige Strategie *Backtrack* eine dieser Kanten. Da alle Kanten identisch gewichtet sind, bedarf es einer tie-breaking-rule, um eine Eindeutigkeit für die *Backtrack*-Strategie zu erreichen. Der Gegenspieler hat Kenntnis über diese tie-breaking-rule und weiß somit für welche Kante sich die Strategie *Backtrack* im Falle mehrerer kürzester Wege entscheidet. Betrachten wir dazu den Extremfall aus Abbildung 3.29. Ein Gegenspieler entfernt immer

Abbildung 3.29: worst-case für die Strategie *Backtrack* auf 1-Self-Spannern

genau eine der beiden ε -Kanten zwischen zwei Knoten $v_i \in V$ und $v_{i+1} \in V$, mit $i \in \{1, \dots, n-1\}$. Dies seien die Kanten, die der genau der tie-breaking-rule entsprechen. In diesem Fall hat *Backtrack* sich immer für eine durch den Gegenspieler entfernte Kante entschieden und tritt auf Grund der Blockade den Rückweg zum Startknoten s an. Für k Blockaden ergibt das Reisekosten von $\text{Backtrack}(P_k) = 2 \sum_{i=0}^{k-1} (1 + i \cdot \varepsilon) + 1 + k \cdot \varepsilon$, die optimalen Kosten von $\text{OPT}(P_k) := 1 + k \cdot \varepsilon$ gegenüberstehen. In Fall einer eindeutigen

Strategie ist *Backtrack* also $(2k + 1)$ -kompetitiv und nicht besser. Es existiert jedoch für jedes Szenario eine Graph-Strategie, bei der *Backtrack* sich optimal verhält und somit 1-kompetitiv ist.

Blockiert ein Gegenspieler hingegen alle Mehrfachkanten zwischen zwei Knoten u und v , so zerfällt der Graph G in zwei Zusammenhangskomponenten $G_1 \ni s$ und $G_2 \ni t$, mit $G_1 \cap G_2 = \emptyset$. In diesem Fall kann kein Weg mit endlichen Kosten mehr gefunden werden. Andernfalls ergibt sich dieselbe Problematik. Betrachten wir nämlich zwei Knoten $u, v \in V$, die durch genau eine Kante miteinander verbunden sind, so stellt dies den Sonderfall von obiger Betrachtung mit Mehrfachkanten dar, wobei alle Kanten - das ist in diesem Fall genau eine - blockiert werden. Da hierbei jedoch kein kürzester Weg π von s nach t mit $|\pi(s, t)| < \infty$ existieren kann und somit die Betrachtung des k -Canadian Traveller Problem sinnlos wird, können wir davon ausgehen, dass solche Kanten nicht entfernt werden.

Korollar 3.33. *Es lässt sich für jede eindeutige Strategie *Backtrack* auf 1-Self-Spannern mit Mehrfachkanten eine obere Schranke angeben, so dass *Backtrack* $(2k + 1)$ -kompetitiv und nicht c -kompetitiv für $c < 2k + 1$ ist. Für jedes dieser Szenarien existiert jedoch eine Graph-Strategie aus der Menge der *Backtrack*-Strategien, die auf diesem Graphen 1-kompetitiv ist.*

$(1 + \varepsilon)$ -Self-Spanner

Wir werden im Folgenden ein Szenario angeben, das die Strategie *Backtrack* auf t -Self-Spannern, mit $t > 1$, zu maximalen Kosten zwingt. Betrachten wir das k -Canadian Traveller Problem in Verbindung mit der Strategie *Backtrack*, so gibt uns die obere Schranke des allgemeinen Graphen für einen kompetitiven Faktor mit $2k + 1$ den Berg an, den es für die untere Schranke zu erklimmen gilt. Wir werden eine Konstruktion angeben, die dies für t -Self-Spanner, mit $t > 1$, leistet. Es sei $k \in \mathbb{N}$ eine Konstante, die die Anzahl an maximal ausfallenden Kanten angibt. Wir geben nun für jedes $k \in \mathbb{N}$ die Konstruktion einer Instanz $P_k \in \Pi$ an, für die eine untere Schranke des kompetitiven Faktors von $2k + 1$ erreicht wird. Zum besseren Verständnis verweisen wir außerdem auf Abbildung 3.30, die die Konstruktion für beliebiges k andeutet.

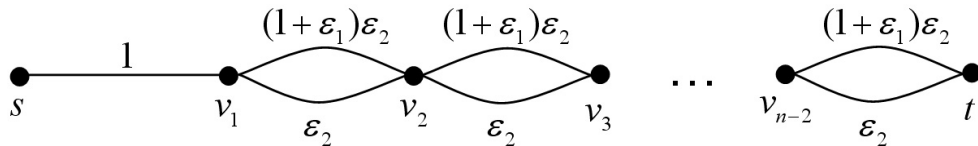


Abbildung 3.30: Beispiel eines $(1 + \varepsilon_1)$ -Self-Spanners

Es sei $G = (V, E)$ ein ungerichteter, gewichteter t -Self-Spanner. Weiter sei V die Knotenmenge mit $|V| := k + 2$. Wir definieren einen Startknoten $s \in V$, von dem eine erste Kante $e_0 := \{s, v_1\}$ mit Kosten $|e_0| := 1$ ausgeht. Vom Knoten $v_1 \in V$ und später von allen Knoten $v_i \in V$, mit $i \leq k + 1$, gehen stets zwei Kanten $e_{i,1} := |\{v_i, v_{i+1}\}|$ und $e_{i,2} := |\{v_i, v_{i+1}\}|$ mit Gewichtung $|e_{i,1}| := \varepsilon_2$ und $|e_{i,2}| := (1 + \varepsilon_1)\varepsilon_2$ zum nächsten Knoten $v_{i+1} \in V$. Letztlich führen auf diese Weise zwei Kanten $e_{k,1}$ und $e_{k,2}$ in den Zielknoten $t \in V$. Ein Streichen der ersten Kante e_0 aus dem Graphen G , würde für keine zwei Knoten $u, v \in V$, die jetzt noch in $G' := (V, E \setminus \{e_0\})$ verbunden sind, eine Verlängerung des Weg zwischen ihnen bedeuten.

Als nächstes untersuchen wir, was bei Entfernen einer der beiden Mehrfachkanten zwischen zwei Knoten v_i und v_{i+1} passiert. Es macht auch hier nur Sinn, die Entfernung der Kante mit den geringen Kosten zu betrachten. Andernfalls bliebe die minimale Distanz zwischen allen Knotenpaaren erneut konstant. Streichen wir also eine beliebige Kante $e_{i,1}$, mit $i \in \{1, \dots, k\}$. Wir erhalten damit einen Stretch-Faktor zwischen den Knoten v_i und v_{i+1} von

$$t = \frac{d_{G'}(v_i, v_{i+1})}{d_G(v_i, v_{i+1})} = \frac{(1 + \varepsilon_1)\varepsilon_2}{\varepsilon_2} = 1 + \varepsilon_1.$$

Ein weiteres Entfernen einer Kante $e_{j,1} \in E_1 := \{e_{1,1}, \dots, e_{k,1}\}$, mit $i \neq j$, ist unabhängig von allen zuvor entfernten Kanten zu betrachten. Der Stretch-Faktor des Graphen G wird dadurch nicht beeinflusst. Der Stretch-Faktor zwischen zwei beliebigen Knoten v_i und v_{i+m} , mit $m \geq 1$, wird nicht größer, wenn weitere Kanten e_j zwischen zwei Knoten v_j und v_{j+1} mit $|e_j| = \varepsilon$ blockiert werden. Der Stretch-Faktor ist nämlich durch $t \leq \frac{d_{G'}(v_i, v_{i+m})}{d_G(v_i, v_{i+m})} \leq \frac{m \cdot (\varepsilon_1)\varepsilon_2}{m \cdot \varepsilon_2} = 1 + \varepsilon_1$. Somit liefert obige Konstruktion tatsächlich einen t -Self-Spanner, der einen Wert $1 + \varepsilon_1$ beliebig nah an 1 annehmen kann.

Wir zeigen nun, wie sich die Strategie *Backtrack* auf diesem Graphen verhält und welche Kosten sie dabei verursacht. Sie startet wie gewohnt am Startknoten s und traversiert einen kürzesten Pfad in Richtung des Zielknotens t . Dieser Pfad führt sie gemäß ihrer aktuellen Berechnung eines kürzesten Weges von s zu t über die Kante e_0 und weiter über die Kanten $e_{i,1}$, mit $i \in \{1, \dots, k\}$, bis hin zum Zielknoten t . Am Knoten v_1 blockiert ein Gegenspieler jedoch die Kante $e_{i,1}$ und zwingt die Strategie *Backtrack* somit zum Rückweg zum Startknoten s . Mit jeder Iteration durchläuft sie einen weiteren Knoten v_i bis sie durch Blockade der Kante $e_{i,1}$, mit $i \in \{1, \dots, k\}$, über die der aktuelle kürzeste Pfad zu t führt, zur Umkehr gezwungen wird. Ein Gegenspieler darf insgesamt k Kanten entfernen und kann die Strategie somit genau k mal zur Rückkehr zwingen. Die daraus resultierenden Kosten lassen sich für ein beliebiges $k \in \mathbb{N}$ mit $\text{Backtrack}(P_k) := 2 \sum_{i=0}^{k-1} (1 + i \cdot (1 + \varepsilon_1)\varepsilon_2) + 1 + k \cdot (1 + \varepsilon_1)\varepsilon_2$ angeben. Die optimalen Kosten belaufen sich auf $\text{OPT}(P_k) := 1 + k \cdot (1 + \varepsilon_1)\varepsilon_2$. Damit

lässt sich folgende untere Schranke angeben:

$$\begin{aligned}
\forall k \in \mathbb{N} \exists P_k \in \Pi : \frac{\text{Backtrack}(P_k)}{\text{OPT}(P_k)} &= \frac{2 \sum_{i=0}^{k-1} (1 + i \cdot (1 + \varepsilon_1) \varepsilon_2) + 1 + k \cdot (1 + \varepsilon_1) \varepsilon_2}{1 + k \cdot (1 + \varepsilon_1) \varepsilon_2} \\
&= \frac{2k + 2 \frac{k \cdot (k-1)}{2} (1 + \varepsilon_1) \varepsilon_2 + 1 + k \cdot \varepsilon_2 + k \cdot \varepsilon_1 \varepsilon_2}{1 + k \cdot \varepsilon_2 + k \cdot \varepsilon_1 \varepsilon_2} \\
&= \frac{2k + (k^2 - k)(1 - \varepsilon_1) \varepsilon_2 + 1 + k \cdot \varepsilon_2 + k \cdot \varepsilon_1 \varepsilon_2}{1 + k \cdot \varepsilon_2 + k \cdot \varepsilon_1 \varepsilon_2} \\
&= \frac{2k + 1 + k^2 \cdot \varepsilon_2 - k^2 \cdot \varepsilon_1 \varepsilon_2 + 2k \cdot \varepsilon_1 \varepsilon_2}{1 + k \cdot \varepsilon_2 + k \cdot \varepsilon_1 \varepsilon_2}
\end{aligned}$$

Für jedes $t \geq 1 + \varepsilon_1$ lässt sich für jedes $\delta > 0$ ein $\varepsilon_2 > 0$ angeben, so dass gilt:

$$\begin{aligned}
\forall k \in \mathbb{N} \exists P_k \in \Pi : \frac{\text{Backtrack}(P_k)}{\text{OPT}(P_k)} &= \frac{2k + 1 + k^2 \cdot \varepsilon_2 + (2k - k^2) \cdot \varepsilon_1 \varepsilon_2}{1 + k \cdot \varepsilon_2 + k \cdot \varepsilon_1 \varepsilon_2} \\
&= \frac{2k + 1 + \varepsilon_2(k^2 - k^2 \cdot \varepsilon_1 + 2k \cdot \varepsilon_1)}{1 + \varepsilon_2(k + k \cdot \varepsilon_1)} \\
&\geq 2k + 1 - \delta.
\end{aligned}$$

Die Kosten der Kante e_0 und somit die Kosten des optimalen Weges, die in obiger Konstruktion mit $|e_0| = 1$ angegeben wurden, lassen sich beliebig hoch wählen, wodurch Lemma 3.1 anwendbar ist. Da außerdem der Wert für δ beliebig nah an Null wählbar ist, gilt sogar eine untere Schranke von $2k + 1$. Da für jeden größeren Wert von t der Gegenspieler höchstens mehr Möglichkeiten hat ein Szenario mit, für die Strategie möglichst ungünstigen Konditionen, zu konstruieren, kann der Wert der unteren Schranke höchstens steigen. Da jedoch für den Wert $t = 1 + \varepsilon_1$, mit $\varepsilon_1 > 0$ bereits der Wert der oberen Schranke des allgemeinen Graphen angenommen wird gilt folgende Überlegung. Der allgemeine Graph kann dabei als t -Self-Spanner mit $t \rightarrow \infty$ angesehen werden. Verknüpfen wir nun dieses Ergebnis mit der oberen Schranke des kompetitiven Faktors für allgemeine Graphen aus Satz 3.2, so erhalten wir folgenden Satz:

Satz 3.34. *Die Strategie Backtrack ist auf t -Self-Spannern mit Mehrfachkanten, für $t > 1$, $(2k + 1)$ -kompetitiv und nicht c -kompetitiv mit $c < 2k + 1$.*

Das oben konstruierte Szenario basiert auf der Annahme, dass der Graph $G = (V, E)$ Mehrfachkanten zwischen zwei Knoten $u, v \in V$ enthalten darf. Das bedeutet, es gibt ein $i \neq j$, so dass gilt $e_i = e_j$ für $e_i, e_j \in E$. Unter Berücksichtigung einer möglichen Anwendung des Canadian Traveller Problem, nämlich der fehlertoleranten Versendung von Informationen über ein Netzwerk $G = (V, E)$, bei dem die Kanten von G beispielsweise Glasfaserkabel repräsentieren, so ist aus ökonomischer Sicht die Annahme sinnvoll, dass jeweils nur *ein* solches Kabel zwei Serverknoten miteinander verbindet. Wir untersuchen daher im Folgenden die Eigenschaften der Strategie *Backtrack* für das Canadian Traveller Problem auf t -Self-Spannern ohne Mehrfachkanten.

Die Strategie Backtrack (auf schlichten Graphen)

Da alle t -Self-Spanner auf schlichten Graphen mit $t < 2$ zyklensfrei sind, ist die Strategie *Backtrack* auf diesen Graphen stets 1-kompetitiv. Wir beginnen daher unsere weitere Untersuchung mit $t = 2$. Nach Lemma 3.31 verfügt ein Kreis in einem 2-Self-Spanner stets über genau 3 Kanten.

Definition 3.35. Es sei $G = (V, E)$ ein ungerichteter, gewichteter Graph. Als **Größe eines Kreises** C in G bezeichnen wir die Anzahl seiner Kanten. Ist C der minimale Pfad $(v_1, \dots, v_n, v_1)$, so hat C die Größe n . Wir schreiben dies auch als $\#C = n$.

Sie bestehen also ausschließlich aus einer Konkatination von gleichseitigen Dreiecken - das sind Kreise C mit Kantenanzahl $\#C = 3$, bei denen alle Kanten identisch gewichtet sind - und Brücken - das sind einfache Pfade, die Dreiecke miteinander verbinden. Die einzelnen Kanten $e_i \in E$ dieser Brücken können eine beliebige Gewichtung $|e_i| > 0$ annehmen, da eine Blockade dieser Kanten den Graphen in zwei Zusammenhangskomponenten zerfallen ließe und demnach kein Stretch-Faktor zwischen den betreffenden Knoten mehr eingehalten werden müsste. Eine Konkatination zweier dieser beiden Grundbauelemente ist genau dann erlaubt, wenn durch diese kein einfacher Kreis $\#C > 3$ entsteht und jeder Kreis C ausschließlich aus Kanten identischer Gewichtung aufgebaut ist. Im Hinblick auf das k -Canadian Traveller Problem werden wir nun eine untere Schranke für die *Backtrack*-Strategie angeben. Dazu werden wir für die Strategie *Backtrack* die Konstruktion eines Szenarios P_k angeben, das für ein $k \in \mathbb{N}$ ein worst-case-Szenario auf 2-Self-Spannern darstellt. Abbildung 3.31 soll die Konstruktion der unteren Schranke eines 2-Self-Spanners

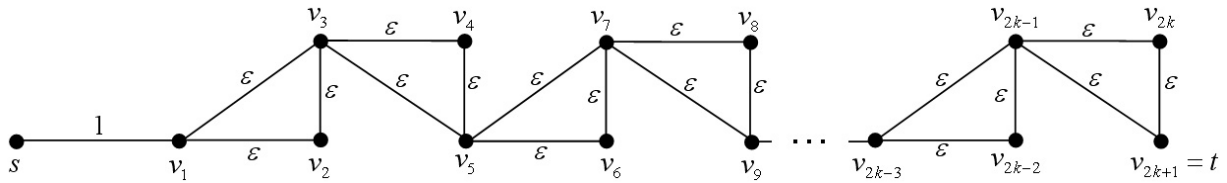


Abbildung 3.31: worst-case eines 2-Self-Spanners für die Strategie *Backtrack*

ohne Mehrfachkanten für die *Backtrack*-Strategie verdeutlichen. Wir konstruieren einen ungerichteten, gewichteten 2-Self-Spanner $G = (V, E)$ und erläutern die Konstruktion einer Instanz $P_k \in \Pi$. Da die Strategie *Backtrack* alle ausfallenden Kanten sieht, die zum Knoten $v_i \in V$ inzident sind, an dem sie sich befindet, beginnen wir mit einer Kante $e_0 := \{s, v_1\}$, die den Startknoten s mit einem ersten Knoten $v_1 \in V$ verbindet, an dem die Strategie die erste ausfallende Kante bemerkt. Von nun an werden nur noch Dreiecke verwendet, die die Strategie durch Blockade der einzelnen Kanten auf dem kürzesten Pfad zum Zielknoten t zur Umkehr zum Startknoten s zwingen. Eine Verwendung von Brücken würde die Strategie nicht zu höheren Kosten zwingen. Das zweite und alle weiteren Dreiecke werden jetzt so an das zuletzt hinzugefügte Dreieck angehängt, dass kein Knotengrad > 4 entsteht und sich der längste einfache Pfad zwischen zwei maximal entfernten Knoten stets verlängert.

Zur Konstruktion ist es wichtig, dass alle Kanten, die häufig durchlaufen werden, möglichst große Kosten verursachen, während Kanten, die seltener überschritten werden, mit geringen Kosten versehen werden. Die Kosten der ersten Kante e_0 sollen verglichen mit den übrigen Kanten möglichst hoch sein, da diese am häufigsten überquert wird. Wir wählen deshalb $|e_0| := 1$. Für alle übrigen Kanten $e_0 \neq e_i \in E$ wählen wir $|e_i| := \varepsilon$. Somit haben alle drei Kanten eines Dreiecks stets dieselbe Länge, wodurch wir die 2-Self-Spanner-Eigenschaft gewährleisten, da der Umweg zwischen zwei Knoten, die im betrachteten Teilgraphen verbunden sind, so maximal die doppelte Distanz bezüglich des Ursprungsgraphen aufweist. Dadurch ergibt sich die Konstruktion aus Abbildung 3.31. Ein Gegenspieler blockiert nun die Kanten $e_j = \{v_{2i-1}, v_{2i+1}\}$, mit $i \in \{1, \dots, k\}$, die zwei Knoten mit aufeinander folgenden ungeraden Knotenindizes miteinander verbindet. Dadurch wird die Strategie *Backtrack* dazu gezwungen an jedem Knoten v_{2i-1} , mit $i \in \{1, \dots, k\}$, an einer blockierten Kante zu stoppen und zum Startknoten zurückzukehren. Ein Gegenspieler darf bis zu k Kanten entfernen, wodurch die Strategie k mal zur unplanmäßigen Umkehr zum Startknoten s gezwungen werden kann. Der kürzeste Pfad von s zu t verläuft dabei genau über die Knoten s, v_{2i-1} und t , mit $i \in \{1, \dots, k\}$. Die Kosten eines optimalen Weges, der genau über die Kanten zwischen Knoten mit ungeradem Index verläuft, betragen $\text{OPT}(P_k) := 1 + 2k \cdot \varepsilon$. Somit ergibt sich für die Berechnung des kompetitiven Faktors der Strategie *Backtrack* auf 2-Self-Spannern folgende Abschätzung:

$$\begin{aligned} \forall k \in \mathbb{N} \exists P_k \in \Pi : \frac{\text{Backtrack}(P_k)}{\text{OPT}(P_k)} &= \frac{2 \sum_{i=0}^{k-1} (1 + i \cdot \varepsilon) + 1 + 2k \cdot \varepsilon}{1 + 2k \cdot \varepsilon} \\ &= \frac{2k + 1 + 2 \frac{k(k-1)}{2} \varepsilon + 2k \cdot \varepsilon}{1 + 2k \cdot \varepsilon} \\ &= \frac{2k + 1 + k^2 \varepsilon + k\varepsilon}{1 + 2k \cdot \varepsilon} \end{aligned}$$

Für jedes $\delta > 0$ lässt sich ein $\varepsilon > 0$ finden, so dass gilt:

$$\begin{aligned} \forall k \in \mathbb{N} \exists P_k \in \Pi : \frac{\text{Backtrack}(P_k)}{\text{OPT}(P_k)} &= \frac{2k + 1 + k^2 \varepsilon + k\varepsilon}{1 + 2k \cdot \varepsilon} \\ &\geq 2k + 1 - \delta. \end{aligned}$$

Da die Kosten des optimalen Weges von s zu t durch Skalierung der Kante e_0 beliebig groß gewählt werden können, ist nach Lemma 3.1 bewiesen ist, dass die Strategie *Backtrack* auf 2-Self-Spannern nicht c -kompetitiv für $c < 2k + 1 - \delta$ ist. Da der Wert für δ , wie oben beschrieben, beliebig nah an Null gewählt werden kann, gilt sogar eine untere Schranke von $2k + 1$. Mit der oberen Schranke für allgemeine Graphen, die nach Satz 3.2 gilt, können wir folgendes Resultat festhalten:

Satz 3.36. *Die Strategie Backtrack ist auf t -Self-Spannern ohne Mehrfachkanten, mit $t \geq 2$, $(2k + 1)$ -kompetitiv und nicht c -kompetitiv mit $c < 2k + 1$.*

Die Strategie Greedy

Die *Greedy*-Strategie lässt sich auf 2-Self-Spannern in der soeben vorgestellten Weise nicht zu Umwegen zwingen.

Satz 3.37. *Die Strategie Greedy ist auf t -Self-Spannern, mit $t < 3$, stets optimal, d.h. 1-kompetitiv.*

Beweis. Es sei Π das k -Canadian Traveller Problem auf t -Self-Spannern, mit $t < 3$. Weiter sei $G = (V, E)$ ein ungerichteter, gewichteter t -Self-Spanner, mit $t < 3$. Dann existiert nach Lemma 3.31 kein Kreis C in G mit $\#C \geq 4$. Wir zeigen, dass sich die Strategie *Greedy* auf einem Kreis C mit $\#C \leq 3$ und damit auf allen t -Self-Spannern mit $t < 3$ optimal verhält. Betrachten wir hierfür zunächst einen Kreis C , mit $\#C = 3$. Ein Kreis mit drei Kanten ist ein Dreieck, wie es in Abbildung 3.32 zu sehen ist. Dieses Dreieck sei aus den drei Knoten

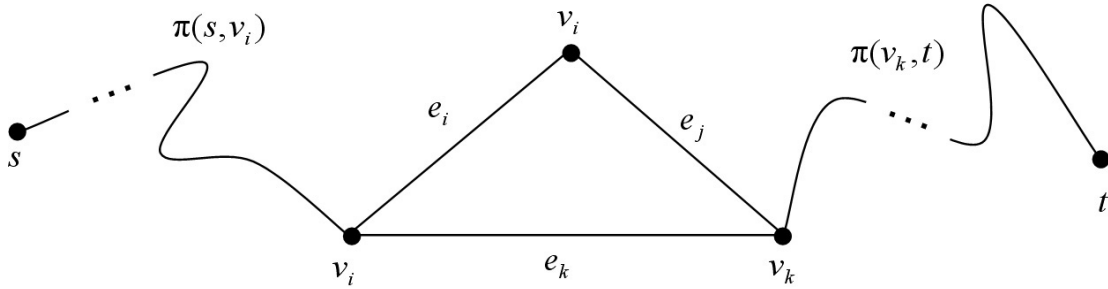


Abbildung 3.32: eindeutiger Pfad durch zwei Knoten eines Dreiecks

$\{v_i, v_j, v_k\} \subseteq V$ und den drei Kanten $\{e_i, e_j, e_k\} \subseteq E$ aufgebaut. Damit *Greedy* zu einem Umweg gezwungen werden kann, muss ein kürzester Weg über den Knoten v_i führen. Dann kann Greedy durch Blockade der Kanten e_j zur Rückkehr zum Knoten v_j gezwungen werden und hat dabei die Kante e_i doppelt traversiert, ohne sich dem Ziel genähert zu haben. Sie würde nun ihren Weg über die unblockierte Kante e_k fortsetzen und hätte so Mehrkosten gegenüber einer optimalen Lösung verursacht. Wir wollen nun erläutern, warum dies nicht möglich ist: Damit die Strategie *Greedy* überhaupt die Traversierung über den Knoten v_i in Betracht zieht, muss

$$|e_i| + |e_j| \leq |e_k| \quad (1)$$

gelten. Darüber hinaus gilt die t -Self-Spanner-Eigenschaft aus der

$$|e_k| + |e_i| < 3 \cdot |e_j| \quad (2)$$

$$|e_k| + |e_j| < 3 \cdot |e_i| \quad (3)$$

folgt. Wir können nun durch Umformung von Ungleichung (2) folgendes zeigen:

$$\begin{aligned} & |e_k| + |e_i| < 3 \cdot |e_j| \\ \Rightarrow & |e_k| + \underbrace{|e_i| + |e_j|}_{\leq |e_k|} < 3 \cdot |e_j| + |e_j| \end{aligned} \quad (2)$$

$$\begin{aligned} \Rightarrow & 2 \cdot |e_k| < 4 \cdot |e_j| \\ \Rightarrow & |e_j| > \frac{1}{2} |e_k| \end{aligned} \quad (4)$$

$$\text{analog gilt:} \quad |e_i| > \frac{1}{2} |e_k| \quad (5)$$

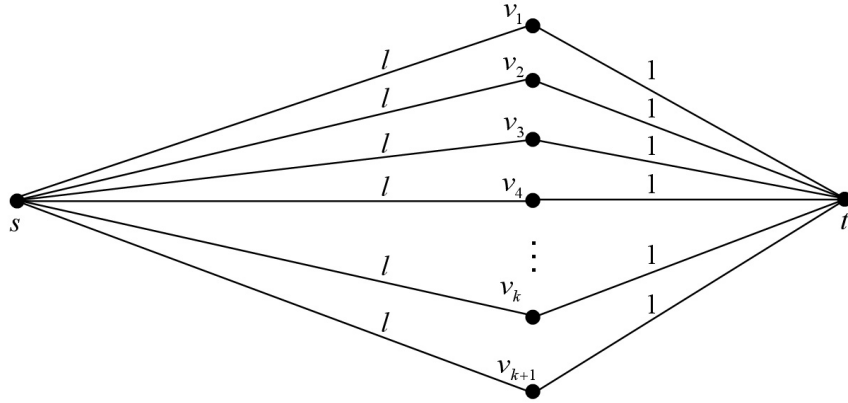
Nun betrachten wir Ungleichung (1) und zeigen, dass diese mit den Ungleichungen (4) und (5) zu einem Widerspruch geführt werden kann:

$$\underbrace{|e_i|}_{> \frac{1}{2} |e_k|} + \underbrace{|e_j|}_{> \frac{1}{2} |e_k|} \leq |e_k|$$

Damit ist gezeigt, dass die gewünschte Konstruktion nicht konstruierbar ist. Deshalb kann Greedy auf t -Self-Spanner mit $t < 3$ durch einen Kreis der Größe 3 nicht zu einem Umweg gezwungen werden. Für einen Kreis C , mit $\#C = 2$, oder eine Brücke zwischen zwei Knoten v_i und v_j , über die ein kürzester Pfad in G' führt, kann *Greedy* erst Recht nicht zu einem Umweg gezwungen werden, da der Strategie stets alle benötigten Informationen zur Verfügung stehen. Erreicht die Strategie Greedy nämlich einen Knoten v_i , an dem mehrere Pfade zum Zielknoten t führen, wobei nur Kreise C mit $\#C = 2$ verwendet werden, so hat die Strategie bei Erreichen von v_i Kenntnis über alle Kanten, die von v_i zu einem Knoten v_j führen und wählt eine kürzeste. Damit verhält sie sich ebenfalls optimal. Insgesamt gilt somit die Behauptung.

□

Wir untersuchen im Folgenden die *Greedy*-Strategie auf t -Self-Spannern mit größerem Stretch-Faktor: Dazu konstruieren wir eine Problemtellklasse $\Pi' \subset \Pi$, die das Problem ausschließlich auf t -Self-Spannern, mit $t \geq 3$, definiert. Gegeben sei daher ein ungerichteter, gewichteter Graph $G = (V, E)$ und eine Konstante $k \in \mathbb{N}$, die die Anzahl maximal ausfallender Kanten bezüglich des k -Canadian Traveller Problem angibt. Auf Grund der strengen Restriktion durch die Definition eines t -Self-Spanners existieren nicht viele Möglichkeiten ein Szenario anzugeben, das für die Strategie *Greedy* einem worst-case gleichkommt. Die Konstruktion, so wie sie in Abbildung 3.33 dargestellt ist, erfolgt dabei durch $k + 1$ disjunkte Pfade vom Startknoten $s \in V$ zum Zielknoten $t \in V$. Jeder dieser Pfade besteht aus zwei Kantensegmenten. Einem Kantensegment $e_i := \{s, v_i\}$, sowie einem Kantensegment $e_{k+1+i} := \{v_i, t\}$, mit $i \in \{1, \dots, k + 1\}$ und $e_i, e_{k+1+i} \in E$. Die Kanten von s zu den Knoten v_i sind dabei mit $|e_i| := l$ gewichtet, während die Traversierung der Kanten zwischen den Knoten v_i und t mit $|e_{k+1+i}| := 1$, für alle $i \in \{1, \dots, k + 1\}$, identische Kosten verursachen.

Abbildung 3.33: worst-case-Szenario für *Greedy* auf t -Self-Spannern

Der größtmögliche Stretch-Faktor t entsteht nun, wenn eine Kante e_{k+1+i} , mit $i \in \{1, \dots, k+1\}$ ausfällt. Betrug die Distanz zwischen v_i und t zuvor $|e_{k+1+i}| = 1$, für alle $i \in \{1, \dots, k+1\}$, so ergibt sich nach Ausfall obiger Kanten ein Stretch-Faktor von $t = \frac{d'_G(v_i, t)}{d_G(v_i, t)} = 2l + 1$, für ein $i \in \{1, \dots, k+1\}$. Bei allen anderen Kombinationen ist der Stretch-Faktor entweder nicht größer, oder die beiden Knoten sind in G' nicht mehr miteinander verbunden. Eine Distanz von $d'_G(u, v) = \infty$ zwischen zwei Knoten $u, v \in V$ muss jedoch nach Definition 3.30 nicht berücksichtigt werden muss.

Für die Strategie sind nun alle Pfade vom Startknoten s zum Zielknoten t gleich lang. Alternativ könnte man durch den Einsatz von ε den Graphen für jede Strategie *Greedy* eindeutig konstruieren, worauf wir an dieser Stelle zur besseren Übersicht verzichten wollen. Sie muss sich daher anhand einer tie-breaking-rule für einen kürzesten Pfad entscheiden und beginnt die Traversierung über eine Kante $e_i \in E$, mit $i \in \{1, \dots, k+1\}$, mit Kosten $|e_i| = l$. Ein Gegenspieler blockiert die auf diese Kante folgende Kante e_{k+1+i} und verhindert somit das Erreichen des Zielknotens t über den Pfad (s, v_i, t) . Da keine weiteren Kanten zum Knoten v_i inzident sind, ist die Strategie *Greedy* an dieser Stelle zum Rückweg gezwungen. Sie erreicht nun nach weiteren Kosten $|e_i| = l$ den Startknoten s , an dem sie erneut eine Entscheidung treffen muss. Die Strategie *Greedy* berechnet nun auf dem Restgraphen $G_1 := (V, E \subseteq \{e_{k+1+i}\})$ einen kürzesten Weg. Wir gehen also davon aus, dass die Strategie einen der k übrigen Pfade auswählt, um von s zu t zu gelangen. Ein Gegenspieler kann insgesamt k mal eine Kante $\{v_i, t\}$ entfernen und die Strategie zur Umkehr zum Startknoten zwingen. Erst im $(k+1)$ -ten Schritt ist der Graph mit all seinen blockierten Kanten bekannt. In diesem seien die Kanten e_j und e_{k+1+j} , für ein $j \in \{1, \dots, k+1\}$, unblockiert und die Strategie wählt final den übrig gebliebenen Pfad (s, v_j, t) vom Startknoten s zum Zielknoten

t und hat bis hierhin folgende Kosten verursacht:

$$\begin{aligned} \forall k \in \mathbb{N} \exists P_k \in \Pi : \text{Greedy}(P_k) &= 2 \sum_{i=0}^{k-1} l + l + 1 \\ &= (2k + 1)l + 1. \end{aligned}$$

Der optimale Weg verläuft ebenfalls über den Pfad (s, v_j, t) und seine Kosten belaufen sich auf $\text{OPT}(P_k) := l + 1$, $\forall P_k \in \Pi$. Damit ergibt sich eine untere Schranke des kompetitiven Faktors von:

$$\forall k \in \mathbb{N} \exists P_k \in \Pi : \frac{\text{Greedy}(P_k)}{\text{OPT}(P_k)} = \frac{(2k + 1)l + 1}{l + 1}.$$

Formen wir nun den Stretch-Faktor t nach l um, erhalten wir $l = \frac{t-1}{2}$. Setzen wir dies in obige Formel ein, erhalten wir:

$$\begin{aligned} \forall k \in \mathbb{N} \exists P_k \in \Pi : \frac{\text{Greedy}(P_k)}{\text{OPT}(P_k)} &= \frac{(2k + 1)\frac{t-1}{2} + 1}{\frac{t-1}{2} + 1} \\ &= \frac{(2k + 1)\frac{t-1}{2} + 1}{\frac{t+1}{2}} \\ &= (2k + 1)\frac{t-1}{t+1} + \frac{2}{t+1}. \end{aligned}$$

Da in unserer Konstruktion durch Skalierung des Graphen die Kosten für einen optimalen Weg von s zu t beliebig groß wählbar sind, erhalten wir nach Lemma 3.1 folgenden Satz:

Satz 3.38. *Die Strategie Greedy ist auf t -Self-Spannern, mit $t \geq 3$, $(2^{k+1} - 1)$ -kompetitiv und nicht c' -kompetitiv für alle $c' < c$ mit*

$$c = (2k + 1)\frac{t-1}{t+1} + \frac{2}{t+1}.$$

Als obere Schranke dient uns hier erneut die des allgemeinen Graphen, welche aus Satz 3.3 hervorgeht.

Für den speziellen Wert $t = 3$ können wir mit

$$c = (2k + 1)\frac{3-1}{3+1} + \frac{2}{3+1} = k + 1$$

außerdem folgendes festhalten:

Die Strategie *Greedy* ist auf 3-Self-Spannern nicht c -kompetitiv für alle $c < k + 1$.

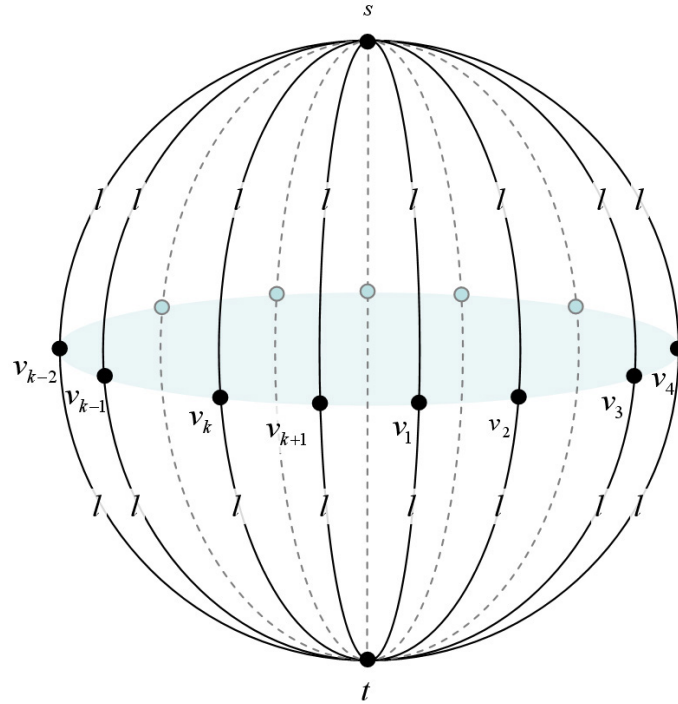


Abbildung 3.34: obere Schranke für 3-Self-Spanner

Abbildung 3.34 zeigt einen Graphen, der diese Konstruktion für beliebiges k zeigt. Diese Instanz P_k lässt sich anschaulich durch eine geometrische Interpretation als $k + 1$ Pfade vom Nordpol s zum Südpol t auf der Oberfläche einer Kugel mit Umfang $2l$ darstellen. Die Pfade sind dabei alle bis auf Start- und Zielknoten disjunkt und jeder Pfad besteht aus 2 Kanten der Länge l . Die Pfade bestehen dabei alle aus 2 Kantensegmenten die über Knoten $v_i \in V$ verlaufen, welche sämtlich auf dem Äquator liegen. Die Strategie *Greedy* verhält sich auf Grund der Konstruktion analog zur Strategie *Backtrack*. Sie muss sich, da alle Wege vom Startknoten s zum Zielknoten t dieselbe Länge haben, mittels einer tie-breaking-rule für einen ersten zu traversierenden Pfad entscheiden. Dieser besteht jedoch, wie alle anderen Pfade von s nach t auch, aus zwei Kanten, wovon der Gegenspieler die zu t inzidente sperrt und die *Greedy*-Strategie zum Rückweg zwingt.

Die Strategie *Greedy* entscheidet sich in den nun folgenden Traversierungsschritten für Kanten, die ihr noch unbekannt sind. Der Gegenspieler kann auf diese Weise die ersten k Traversierungsversuche eines kürzesten Pfades vom Startknoten s zum Zielknoten t verhindern und muss die Strategie frühestens im $(k + 1)$ -ten Versuch passieren lassen. Dann hat sie bis hierhin jedoch k mal eine Kante $\{s, v_i\}$ doppelt überquert. Der optimale Weg führt über die beiden unblockierten Kanten $\{s, v_j\}$ und $\{v_j, t\}$, für ein $j \in \{1, \dots, k + 1\}$. Seine Kosten betragen $\text{OPT}(P_k) := |\{s, v_j\}| + |\{v_j, t\}| = 2l$. Für die Kosten einer unteren Schranke der Strategie *Greedy* auf 3-Self-Spannern gilt, bereits aus der allgemeineren

unteren Schranke aus Satz 3.38 bekannt:

$$\begin{aligned} \forall k \in \mathbb{N} \exists P_k \in \Pi : \frac{\text{Greedy}(P_k)}{\text{OPT}(P_k)} &= \frac{\sum_{i=1}^k 2l + 2l}{2l} \\ &= \frac{2kl + 2l}{2l} \\ &= k + 1. \end{aligned}$$

Wir zeigen nun, dass $k + 1$ auch eine obere Schranke darstellt und somit folgender Satz gilt:

Satz 3.39. *Die Strategie Greedy ist auf 3-Self-Spannern $(k + 1)$ -kompetitiv und nicht c -kompetitiv für alle $c < k + 1$.*

Beweis. Wir betrachten das k -Canadian Traveller Problem auf t -Self-Spannern, mit $t = 3$. Gegeben sei ein ungerichteter, gewichteter 3-Self-Spanner $G = (V, E)$, sowie eine Konstante $k \in \mathbb{N}$, die die Anzahl maximal ausfallender Kanten bezüglich des k -Canadian Traveller Problem angibt. Es seien ferner ein Startknoten $s \in V$ und ein Zielknoten $t \in V$ gegeben. Für unsere Argumentation ist $P \in \Pi$ eine Instanz des k -Canadian Traveller Problem auf 3-Self-Spannern. Alle Pfade, die am Startknoten s beginnen und zum Zielknoten t führen können sich nicht beliebig unterscheiden. Die 3-Self-Spanner-Eigenschaft schränkt die Möglichkeiten hierzu sehr stark ein. Immer dann, wenn sich zwei Pfade an einem Knoten v_i trennen und an einem Knoten v_j wieder zusammenführen, kann zwischen v_i und v_j kein Kreis mit mehr als 4 Kanten existieren. Knoten an denen sich verschiedene s - t -Pfade trennen oder wieder zusammen finden bezeichnen wir im Folgenden mit $w_1, \dots, w_m$.

Definition 3.40. *Den Teilgraph von G , der aus allen Pfaden zwischen zwei Knoten w_i und w_{i+1} besteht, nennen wir **i -te Gruppe von G** und schreiben dies kurz als $\text{Gruppe}_i(G)$. Dabei sind die Knoten w_i und w_{i+1} so gewählt, dass alle Pfade der i -ten Gruppe mit Ausnahme von w_i und w_{i+1} knotendiskunkt sind.*

Sowohl die Strategie *Greedy* als auch eine optimale Lösung traversieren die Knoten w_i in aufsteigender Reihenfolge $w_1, \dots, w_m$. Dabei fällt auf, dass nur die Gruppe, die den höchsten kompetitiven Faktor erzielt betrachtet werden muss. Eine Hinzunahme von weiteren Gruppen kann den kompetitiven Faktor nur noch verschlechtern. Es gilt nämlich

$$\frac{\text{Greedy}(P)}{\text{OPT}(P)} \leq \max_i \frac{\text{Greedy}(\text{Gruppe}_i(G))}{\text{OPT}(\text{Gruppe}_i(G))}.$$

Das Pfadsegment zwischen s und w_1 sowie zwischen w_m und t kann ebenfalls vernachlässigt werden, da die Pfadsegmente von *Greedy* und einer optimalen Lösung identisch sind und *Greedy* auf diesen Teilstücken nie zurück geht. Betrachten wir nun, was in einer einzelnen Gruppe im schlimmsten Fall passieren kann:

Wir betrachten die Gruppe, für die das Verhältnis aus dem durch die Strategie *Greedy* zurück gelegten Weg zum optimalen Weg maximal ist. Dabei nehmen wir an, es existieren in einer solchen i -ten Gruppe mindestens zwei Pfade $\pi_1(w_i, w_{i+1}) \neq \pi_2(w_i, w_{i+1})$ von w_i nach w_{i+1} . Anderenfalls ließe sich die *Greedy*-Strategie nicht zu Umwegen zwingen. Dann gibt es einen Knoten w_i , an dem sich die beiden Pfade trennen und einen Knoten w_{i+1} an dem sie wieder aufeinander treffen. Wir wissen nach Lemma 3.31, dass ein einfacher Kreis zwischen den Knoten w_i und w_{i+1} aus maximal 4 Kanten besteht. An einem einfachen Kreis mit 2 Kanten verhält sich die Strategie *Greedy* nach unseren bisherigen Überlegungen optimal. Wir werden im Folgenden die Fälle betrachten, in denen ein einfacher Kreis aus 3 bzw. 4 Kanten besteht. Dieser Kreis, interpretiert als Teilgraph, muss selbst wiederum ein 3-Self-Spanner sein. Betrachten wir zuerst Abbildung 3.35, in der der Fall für drei Kanten auf

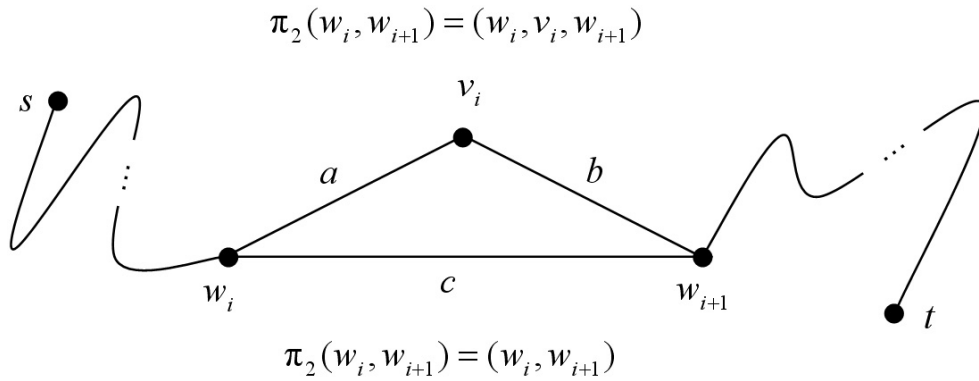


Abbildung 3.35: Einschränkung für einen 3-Self-Spanner mit Kreis aus 3 Kanten

einem einfachen Kreis dargestellt ist. Wir betrachten eine i -te Gruppe in einem Graphen $G = (V, E)$, so wie sie in Abbildung 3.35 dargestellt ist. Auf Grund der 3-Self-Spanner-Eigenschaft muss gelten:

$$|b| + |c| \leq 3 \cdot |a| \quad (1)$$

$$|a| + |c| \leq 3 \cdot |b| \quad (2)$$

$$|a| + |b| \leq 3 \cdot |c|. \quad (3)$$

Damit *Greedy* den alternativen Weg π_2 einschlägt, auf dem eine Kante blockiert werden kann, muss gelten:

$$|a| + |b| \leq |c|. \quad (4)$$

Wir zeigen nun, dass $|a| = |b|$ gilt. Angenommen, die Werte für $|a|$ und $|b|$ wären verschieden, so könnten wir auf Grund der Symmetrie ohne Beschränkung der Allgemeinheit

annehmen, dass $|a| > |b|$ ist. Dann ergibt sich für Ungleichung (3):

$$\begin{aligned} \overbrace{|c|}^{\geq |a|+|b|} &\leq 3 \cdot |b| - |a| \\ \Rightarrow |a| + |b| &\leq 3 \cdot |b| - |a| \\ \Rightarrow |a| &\leq |b|. \end{aligned}$$

Da dies zu einem Widerspruch führt gilt mit $|a| = |b|$ in Verbindung mit Ungleichung (4):

$$|a| = |b| \leq \frac{1}{2}|c|.$$

Für den Fall, dass der Kreis wie in Abbildung 3.36 aus genau 4 Kanten besteht, können wir wie folgt argumentieren: Wir betrachten nun eine Gruppe, bei der zwei Pfade

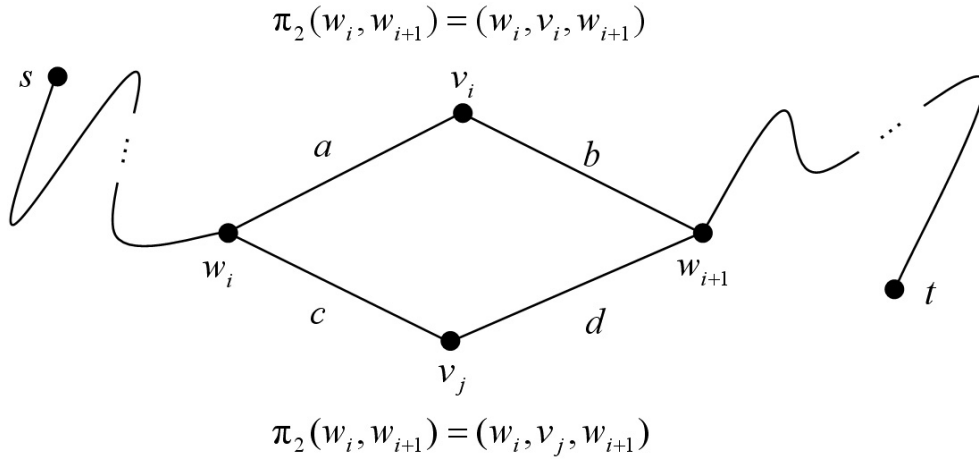


Abbildung 3.36: Einschränkung für einen 3-Self-Spanner mit Kreis aus 4 Kanten

$\pi_1(w_i, w_{i+1}) \neq \pi_2(w_i, w_{i+1})$ einen Knoten w_i mit einem Knoten w_{i+1} verbinden. Der erste Pfad $\pi_1(w_i, w_{i+1})$ verläuft dabei über die Kanten $c := \{w_i, v_j\}$ und $d := \{v_j, w_{i+1}\}$, während der zweite Pfad $\pi_2(w_i, w_{i+1})$ über die Kanten $a := \{w_i, v_i\}$ und $b := \{v_i, w_{i+1}\}$ verläuft. Wir wollen weiter annehmen, dass $\pi_1(w_i, w_{i+1})$ der optimale Pfad sei, und untersuchen im Folgenden, wie die Kanten $a, b \in E$ auf dem Pfad $\pi_2(w_i, w_{i+1})$ beschaffen sein müssen, damit die 3-Self-Spanner-Eigenschaft eingehalten wird. Es gilt in jedem Fall:

$$\begin{aligned} |b| + |c| + |d| &\leq 3 \cdot |a| \\ |a| + |c| + |d| &\leq 3 \cdot |b| \\ |a| + |b| + |d| &\leq 3 \cdot |c| \\ |a| + |b| + |c| &\leq 3 \cdot |d|. \end{aligned}$$

Damit Greedy sich überhaupt für den Pfad $\pi_2(w_i, w_{i+1})$ entscheiden kann, muss gelten:

$$|a| + |b| \leq |c| + |d|. \quad (1)$$

Wir können nun folgern, dass gilt:

$$\begin{aligned} |a| + |b| &\leq |c| + |d| \leq 3 \cdot |a| - |b| \\ \Rightarrow |a| + |b| &\leq 3 \cdot |a| - |b| \\ \Rightarrow |b| &\leq |a|. \end{aligned}$$

Nach symmetrischer Argumentation gilt $|a| \leq |b|$ und somit $|a| = |b|$.

Nach Ungleichung (1) gilt $|a| + |b| \leq |c| + |d|$. Damit der Umweg für die Strategie *Greedy* möglichst groß ausfällt, wählen wir $|a| + |b| = |c| + |d|$. In allen anderen Fällen verursacht *Greedy* höchstens geringere Kosten.

Wir konnten also für beide Fälle nachweisen, dass die zwei Kanten, über die die Strategie *Greedy* zu einem Umweg gezwungen werden soll, identische Kosten haben müssen, damit der Graph ein 3-Self-Spanner sein kann.

Eine Erweiterung des Graphen durch *eine* zusätzliche Kante am Knoten v_i ist nicht sinnvoll. Von hier aus wäre auf Grund der 3-Self-Spanner-Eigenschaft einzig eine Kante mit identischen Kosten der soeben blockierten Kante denkbar. Dann wäre der Effekt der Blockade jedoch gänzlich verloren gegangen. Eine Erweiterung des Graphen am Knoten v_i durch einen Pfad aus *zwei* Kanten $e, f \in E$, die zu Knoten w_{i+1} führen, so wie es in Abbildung 3.37 dargestellt wird, ist ebenfalls nicht möglich, da dann, wie folgt, die 3-Self-Spanner-Eigenschaft verletzt wird: Für die folgende Argumentation können wir annehmen,

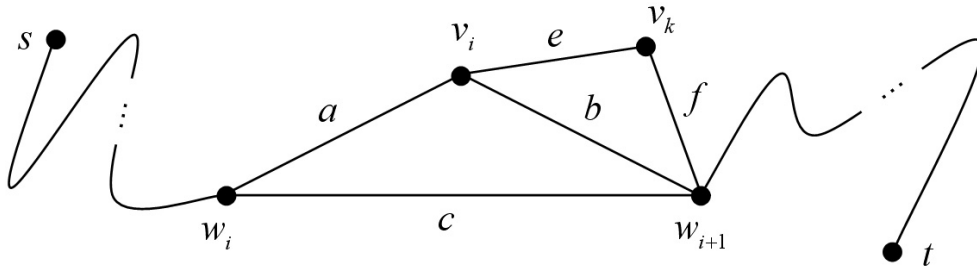


Abbildung 3.37: Erweiterungsmöglichkeiten des Graphen am Knoten v_i

dass $|c| \geq |a| + |b|$ sowie $|a| = |b|$ gelten. Wir betrachten den Graphen $G' := (V, E \setminus \{b, f\})$. In diesem Graphen existiert ein Pfad von v_k zu w_{i+1} mit $|\pi_{G'}^{\min}(v_k, w_{i+1})| = |e| + |a| + |c| \geq |e| + 3 \cdot |a|$. Da für die Kosten der Kante e nach Voraussetzung $|e| > 0$ gilt, muss für die Kosten der Kante f $|f| > |a|$ gelten. Andernfalls wäre G' mit

$$d_{G'}(v_k, w_{i+1}) = |e| + 3 \cdot |a| > 3 \cdot |a| \geq 3 \cdot d_G(v_k, w_{i+1})$$

kein 3-Self-Spanner. Analog muss für einen Graphen $G'' := (V, E \setminus \{b, e\})$ gelten, dass $|e| > |a|$ ist. Mit $|e| > |a|$ und $|f| > |a|$ gilt dann aber für den Graphen $\hat{G} := (V, E \setminus \{a, b\})$:

$$d_{\hat{G}}(w_i, v_i) = |e| + |f| + |c| > 4 \cdot |a| > 3 \cdot d_G(w_i, v_i).$$

Wir konnten also zeigen, dass *Greedy* nach Entdecken der ersten blockierten Kante $\{v_i, w_{i+1}\}$ zum Knoten w_i zurückkehrt. Wir bezeichnen mit G_i den Graphen, in dem i Kanten entfernt wurden. Dann können wir durch vollständige Induktion folgende Abschätzung durchführen:

$$\text{Greedy}(G_i) \leq 2 \cdot \frac{1}{2} \text{OPT}(G_k) + \text{Greedy}(G_{i+1})$$

Wie oben bereits gezeigt wurde, gilt die Behauptung für G_0 , den Graphen, in dem noch keine Kante entfernt wurde.

Der Induktionsschritt $i \mapsto i + 1$ gelingt durch folgende Überlegung: Die Strategie *Greedy* kann nur durch eine weitere blockierte Kante zu zusätzlichen Kosten gezwungen werden. Dies wird erreicht, falls die Strategie eine Kante $\{w_i, v'_i\}$ traversiert und so einen Knoten $v'_i \neq v_i$ erreicht. An diesem entdeckt sie eine weitere Blockade und kehrt zum Knoten w_i zurück. Dadurch hat sie Kosten $\leq 2 \cdot \frac{1}{2} \text{OPT}(G_k)$ verursacht. Damit ist der Induktionsschritt gezeigt und die Aussage gilt für alle $i \geq 0$.

Für eine maximale Ausfallmenge von k entfernten Kanten ergibt sich:

$$\text{Greedy}(G_i) \leq 2 \cdot \frac{1}{2} \text{OPT}(G_k) + \text{Greedy}(G_{i+1}).$$

Damit erhalten wir:

$$\begin{aligned} \text{Greedy}(G_0) &\leq \text{OPT}(G_k) + \text{Greedy}(G_1) \\ &\leq \text{OPT}(G_k) + \text{OPT}(G_k) + \text{Greedy}(G_2) \\ &\quad \vdots \\ &\leq \underbrace{\text{OPT}(G_k) + \dots + \text{OPT}(G_k)}_{k\text{-mal}} + \text{Greedy}(G_k) \\ &= k \cdot \text{OPT}(G_k) + \text{Greedy}(G_k) \end{aligned}$$

Im Graphen G_k in dem alle maximal entfernbaren Kanten bekannt sind, berechnet die Strategie *Greedy* einen kürzesten Weg und kann nicht durch weitere Blockaden zu Umwegen gezwungen werden. Sie verhält sich in G_k stets optimal, wodurch $\text{Greedy}(G_k) \leq \text{OPT}(G_k)$ gilt. Daraus folgt somit:

$$\begin{aligned} \text{Greedy}(G_0) &\leq k \cdot \text{OPT}(G_k) + \text{Greedy}(G_k) \\ &\leq k \cdot \text{OPT}(G_k) + \text{OPT}(G_k) \\ &\leq (k + 1) \cdot \text{OPT}(G_k) \end{aligned}$$

Wie wir oben gezeigt haben, existiert kein Graph G' , der die Strategie *Greedy* auf 3-Self-Spannern zu höheren Kosten zwingen kann. Es gilt:

$$\forall P \in \Pi : \frac{\text{Greedy}(P)}{\text{OPT}(P)} \leq k + 1.$$

□

Für einen Stretch-Faktor t , der gegen unendlich strebt, erhalten wir eine Graph-Klasse ohne Restriktionen und können problemlos das worst-case-Szenario für *Greedy* auf allgemeinen Graphen verwenden. Es ergibt sich die bereits bekannt obere Schranke aus Satz 3.3 von $2^{k+1} - 1$.

Damit sind unsere Untersuchungen zu speziellen Graphklassen für die beiden konkreten Strategien abgeschlossen. In Abbildung 6.1 auf Seite 151 werden die in diesem Kapitel beschriebenen Ergebnisse noch einmal zur besseren Übersicht in einer Tabelle zusammengetragen.

Kapitel 4

Untere Schranken deterministischer Strategien

In diesem Kapitel untersuchen wir das k -Canadian Traveller Problem in einem allgemeineren Zusammenhang. Wir lösen uns von den beiden in Kapitel 3.1 vorgestellten Strategien *Backtrack* und *Greedy* und untersuchen das Problem in Hinblick auf eine beliebige deterministische Strategie. Dabei verstehen wir unter einer Menge Σ die Menge aller Strategien, die das k -Canadian Traveller Problem korrekt lösen, und bezeichnen die Kosten einer Strategie $S \in \Sigma$ auf einer Instanz $P \in \Pi$ des k -CTP mit $S(P)$. Analog seien die Kosten einer optimalen Lösung mit $\text{OPT}(P)$ angegeben. Auch in diesem Kapitel weiten wir unsere Untersuchungen auf die aus Kapitel 3 bekannten Graphklassen aus.

4.1 Allgemeiner Graph

Für jede beliebige deterministische Strategie gilt auf allgemeinen Graphen der folgende Satz:

Satz 4.1. *Keine eindeutige deterministische Strategie ist auf allgemeinen Graphen mit k ausfallenden Kanten besser als $(2k + 1)$ -kompetitiv.*

Da die Beweisführung zu der für *Backtrack* auf allgemeinen Graphen identisch ist, wollen wir an dieser Stelle lediglich auf Abbildung 4.1 verweisen, die das Szenario noch einmal kurz in Erinnerung ruft. Jede Strategie kann k mal zur Rückkehr zum Startknoten s gezwungen werden und versucht damit Wegkosten von insgesamt $2kl+1+\varepsilon$, die Wegkosten von $l + \varepsilon$ gegenüberstehen. Dadurch ergibt sich obiger Satz.

Durch diese Kenntnis ist die *Backtrack*-Strategie bezüglich des kompetitiven Faktors im worst-case als optimale deterministische Online-Strategie zur Lösung des k -Canadian Traveller Problem auf allgemeinen Graphen zu betrachten. Dies bedeutet jedoch nicht, dass die *Greedy*-Strategie auf einer Teilmenge $\Pi' \subset \Pi$ von Problemen des k -Canadian Traveller Problem keine bessere Lösung finden kann.

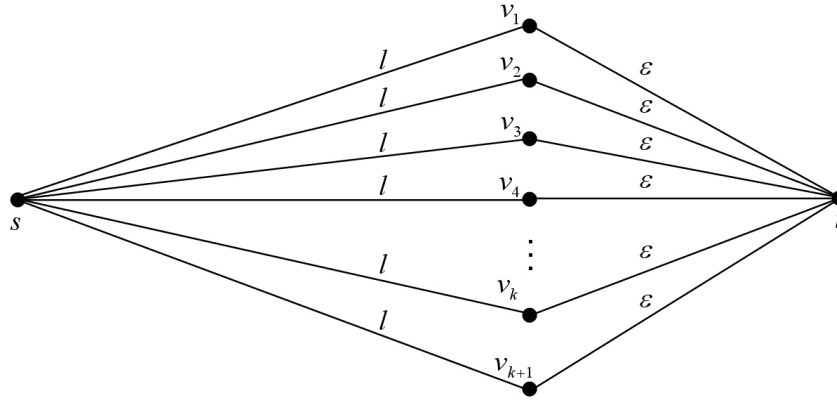


Abbildung 4.1: allgemeine untere Schranke

Nach Fertigstellung des Beweises zu Satz 4.1, wurden wir darauf aufmerksam, dass S. Westphal in seiner Arbeit [SW07], die im März dieses Jahres verfasst wurde, die gleiche Problemstellung untersucht und ebenfalls oben genanntes Ergebnis vorstellt.

4.2 Vollständiger Graph

Mit dem Wissen aus Abschnitt 4.1 können wir die Klasse der vollständigen Graphen relativ schnell abhandeln. Wir erweitern den Graphen aus Abbildung 4.1 zu einem vollständigen Graphen, indem wir für alle Knoten, die noch nicht direkt durch eine Kante verbunden sind, eine neue Kante mit Gewicht *unendlich* einfügen. Den Beweis zu Satz 4.1 können wir dann uneingeschränkt auf das neue Szenario anwenden. Sollte eine Strategie eine der neuen Kanten verwenden, um ins Ziel zu gelangen, kann sie auf Grund des hohen Kantengewichts sicher nicht besser als $(2k + 1)$ -kompetitiv sein. Es gilt also der folgende Satz:

Satz 4.2. *Keine eindeutige deterministische Strategie ist auf vollständigen Graphen mit k ausfallenden Kanten besser als $(2k + 1)$ -kompetitiv.*

Weil Satz 4.2 für jede beliebige deterministische Strategie gilt, kennen wir mit *Backtrack* bereits eine bezüglich des kompetitiven Faktors im worst-case optimale Online-Strategie zur Lösung des k -Canadian Traveller Problem auf vollständigen Graphen. Daraus folgt unmittelbar, dass sich keine größere untere Schranke als die aus Satz 4.2 finden lässt.

4.3 Vollständiger (m, n) -Gittergraph

Für die aus Kapitel 3.3.2 bekannten vollständigen (m, n) -Gittergraphen gilt hinsichtlich einer beliebigen deterministischen Strategie der folgende Satz:

Satz 4.3. *Keine eindeutige deterministische Strategie ist auf vollständigen (m,n) -Gittergraphen mit k ausfallenden Kanten besser als c -kompetitiv mit*

$$c = \lfloor \sqrt{k} \rfloor + \frac{1}{2} + \frac{1}{2\lfloor \sqrt{k} \rfloor}.$$

Beweis. Es sei Π das k -Canadian Traveller Problem auf vollständigen (m,n) -Gittergraphen, Σ die Menge aller deterministischen Strategien, die jede Instanz $P \in \Pi$ korrekt lösen und $k \in \mathbb{N}$ die Anzahl maximal ausfallender Kanten. Um eine untere Schranke des kompetitiven Faktors einer beliebigen deterministischen Strategie $S \in \Sigma$ zum Lösen der Problemklasse Π anzugeben, konstruieren wir für jedes k eine worst-case-Instanz $P_k \in \Pi$. Gegeben sei ein gewichteter vollständiger (m,n) -Gittergraph $G = (V, E)$, mit $m = n = \lfloor \sqrt{k} \rfloor + 1$ und $|e| = 1$, für alle $e \in E$. Den Knoten in der i -ten Zeile und j -ten Spalte bezeichnen wir mit $v_{i,j}$, für $i, j \in \{1, \dots, \lfloor \sqrt{k} \rfloor + 1\}$. Start- und Zielknoten seien durch $s := v_{1,x}$ und $t := v_{m,x'}$ gegeben, wobei x und x' folgende Werte repräsentieren:

- $x = \lfloor \frac{\lfloor \sqrt{k} \rfloor}{2} \rfloor + 1$
- $x' = \lceil \frac{\lfloor \sqrt{k} \rfloor}{2} \rceil + 1$

Ist $\lfloor \sqrt{k} \rfloor$ gerade, so gilt $x = x'$. Dann befinden sich s und t also in derselben Spalte. Der Startknoten markiert exakt die Mitte der ersten Zeile, der Zielknoten die Mitte der letzten Zeile (vgl. Abbildung 4.2). Ist $\lfloor \sqrt{k} \rfloor$ jedoch ungerade, existiert kein Knoten in G , der exakt

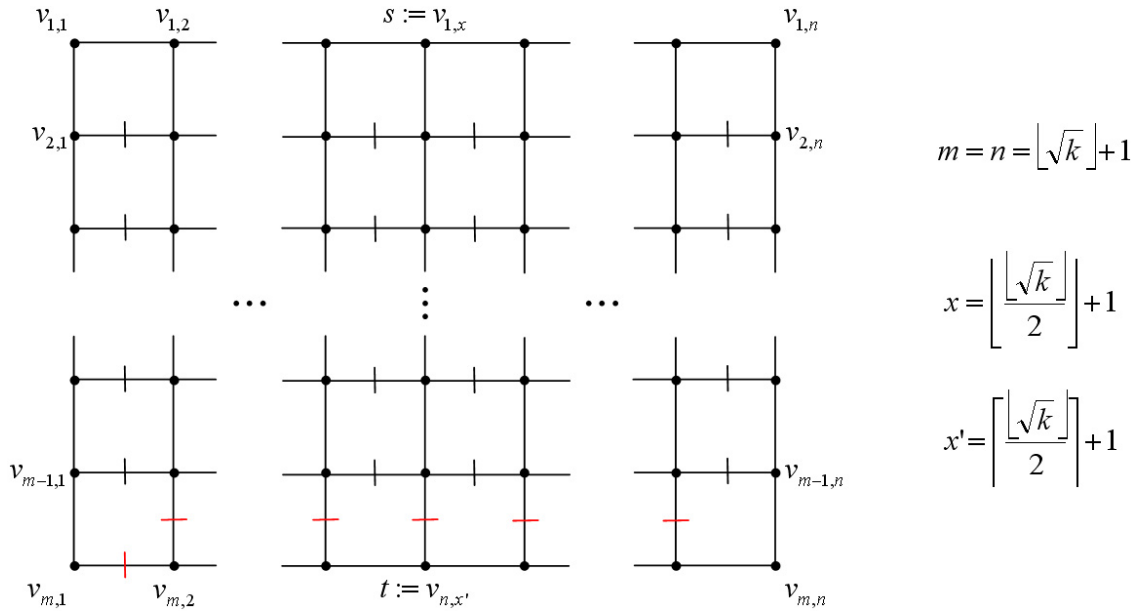


Abbildung 4.2: $\lfloor \sqrt{k} \rfloor$ gerade $\Rightarrow s$ und t sind in derselben Spalte

die Mitte einer Zeile markiert. Definitionsgemäß gilt dann $x = x' - 1$, woraus folgt, dass s in der ersten Zeile gegenüber t in der letzten Zeile um eine Spalte versetzt ist, siehe Abbildung 4.3.

Es sei $P_1 \in \Pi$, mit $P_1 = P_2 = P_3$, die in Abbildung 4.4 dargestellte Problemistanz. Jede deterministische Strategie $S \in \Sigma$ hat zu Beginn einzig die Möglichkeit, einer der Kanten $\{s, v_{1,2}\}$ oder $\{s, v_{2,1}\}$ zu folgen, um sich dem Zielknoten t zu nähern. In Abhängigkeit von der Präferenz der Strategie S blockiert ein Gegenspieler entweder die Kante $\{v_{1,2}, t\}$ oder $\{v_{2,1}, t\}$. Abbildung 4.4 zeigt also den Fall, in dem S zunächst die Kante $\{s, v_{2,1}\}$ traversiert und ein Gegenspieler darauf durch Blockieren der Kante $\{v_{2,1}, t\}$ reagiert. Für die von S verursachten Kosten vom Start bis ins Ziel gilt in jedem Fall:

$$S(P_1) \geq 4 = (2\lfloor\sqrt{k}\rfloor + 1)\lfloor\sqrt{k}\rfloor + 1.$$

Für die Kosten einer optimalen Lösung gilt jedoch:

$$\text{OPT}(P_1) = 2 = 2\lfloor\sqrt{k}\rfloor.$$

Wir geben hier die Kosten $S(P_1)$ und $\text{OPT}(P_1)$ bereits zusätzlich in der allgemeinen Schreibweise an. Wie wir später sehen werden, behalten diese Formeln auch bei den worst-case-Szenarien für $k \geq 4$ Gültigkeit.

Kommen wir nun zur Komplexitätsanalyse der übrigen Fälle. Die Problemistanzen $P_k \in \Pi$, mit $k \geq 4$, sind dann wie in den Abbildungen 4.2 und 4.3 aufgebaut. Da ein Gegenspieler a priori alle Kanten $\{v_{i,j}, v_{i,j+1}\}$, mit $i \in \{2, \dots, m-1\}$ und $j \in \{1, \dots, n-1\}$, blockiert (in den Abbildungen durch schwarze Markierungen dargestellt), können wir davon ausgehen, dass eine beliebige deterministische Strategie S gemäß ihrer Präferenz zunächst einem der Pfade $(s, \dots, v_{1,j}, \dots, v_{m,j}, \dots, t)$, mit $j \in \{1, \dots, n\}$, von s nach t folgt, um sich dem Ziel zu nähern. Da ein Gegenspieler die Schritte der Strategie kennt, kann er durch Blockieren einer der folgenden Kanten darauf reagieren:

- $\{v_{m,1}, v_{m,2}\}$, für $j = 1$
- $\{v_{m-1,j}, v_{m,j}\}$, für $j \in \{2, \dots, n-1\}$
- $\{v_{m,n-1}, v_{m,n}\}$, für $j = n$

Würde S nicht dem Pfad $(s, \dots, v_{1,j}, \dots, v_{m,j}, \dots, t)$ bis $v_{m-1,j}$ (für $j \in \{2, \dots, n-1\}$) bzw. $v_{m,j}$ (für $j \in \{1, n\}$) folgen, sondern bereits zu einem früheren Zeitpunkt umkehren, so würden sich die Wegkosten von S höchstens vergrößern. Denn dann hätte die Strategie bereits Kosten verursacht, ohne an neue Informationen gelangt zu sein, also ohne eine ausfallende Kante entdeckt zu haben. Erfährt die Strategie S an Knoten $v_{m-1,j}$ bzw. $v_{m,j}$ von der ausfallenden Kante, können wir annehmen, dass sie auf dem kürzesten Weg zum Knoten $v_{1,j}$ zurückkehrt. Denn unabhängig von der Vorgehensweise der Strategie ist zu diesem Zeitpunkt bereits bekannt, dass ein freier Weg von $v_{m-1,j}$ bzw. $v_{m,j}$ nach t nur über $v_{1,j}$ führen kann. Wenn wir im Sinne der Strategie S annehmen, dass sie keinen der Pfade $(v_{1,j}, v_{2,j}, \dots, v_{m,j})$ ein zweites Mal betritt, von dem sie bereits weiß, dass er in eine Sackgasse führt, kann ein Gegenspieler die Strategie nach obigem Schema genau $n-1$ mal zum Umkehren zwingen. Danach sind S alle ausfallenden Kanten bekannt und ein Gegenspieler

hat keine Möglichkeit mehr, die Strategie auf ihrem Weg nach t zu aufzuhalten. Dabei minimiert S die Wegkosten, wenn sie die insgesamt n Pfade $(v_{1,j}, v_{2,j}, \dots, v_{m,j})$ beispielsweise in der Reihenfolge $j = 1, 2, \dots, n$ besucht, da dann keine Kante öfter als nötig traversiert wird.

Unabhängig davon, nach welcher Vorgehensweise S eine Lösung einer Probleminstanz P_k ermittelt, gilt für die von der Strategie S verursachten Kosten:

$$\begin{aligned} S(P_k) &\geq 2(\lfloor \sqrt{k} \rfloor - 1)(\lfloor \sqrt{k} \rfloor - 1) + 2\lfloor \sqrt{k} \rfloor + 2\lfloor \sqrt{k} \rfloor - 1 + \lfloor \sqrt{k} \rfloor \\ &= (2\lfloor \sqrt{k} \rfloor + 1)\lfloor \sqrt{k} \rfloor + 1. \end{aligned}$$

Die Kosten $S(P_k)$ setzen sich aus folgenden Teilkosten zusammen, vergleiche Abbildung 4.2 und 4.3:

- Besuch von $2(\lfloor \sqrt{k} \rfloor - 1)(\lfloor \sqrt{k} \rfloor - 1)$ Kanten in der zweiten bis $(n - 1)$ -ten Spalte
- Besuch von $2\lfloor \sqrt{k} \rfloor$ Kanten in der ersten Spalte
- Besuch von $2\lfloor \sqrt{k} \rfloor - 1$ Kanten in der ersten und m -ten Zeile (*)
- Besuch von $\lfloor \sqrt{k} \rfloor$ Kanten in der n -ten Spalte

Ist $\lfloor \sqrt{k} \rfloor$ gerade, so ist die Anzahl der Kanten in (*) sogar $2\lfloor \sqrt{k} \rfloor$, nicht aber im allgemeinen Fall. Dieser Zusammenhang wird durch den Vergleich der Abbildungen 4.2 und 4.3 deutlich. Die Kosten einer optimalen Lösung lassen sich durch

$$\text{OPT}(P_k) \leq 2\lfloor \sqrt{k} \rfloor.$$

abschätzen und setzen sich wie folgt zusammen:

- Besuch von $\lfloor \sqrt{k} \rfloor$ Kanten in der ersten und m -ten Zeile
- Besuch von $\lfloor \sqrt{k} \rfloor$ Kanten in der n -ten Spalte

Vergleiche auch dazu wieder Abbildung 4.2 und 4.3. Insgesamt können wir also festhalten:

$$\begin{aligned} \forall k \in \mathbb{N} \exists P_k \in \Pi : \frac{S(P_k)}{\text{OPT}(P_k)} &\geq \frac{(2\lfloor \sqrt{k} \rfloor + 1)\lfloor \sqrt{k} \rfloor + 1}{2\lfloor \sqrt{k} \rfloor} \\ &= \lfloor \sqrt{k} \rfloor + \frac{1}{2} + \frac{1}{2\lfloor \sqrt{k} \rfloor}. \end{aligned}$$

In obiger Ungleichung können die Kosten einer optimalen Lösung durch Skalierung aller Kantengewichte mit demselben großen Faktor beliebig hoch gewählt werden. Nach Lemma 3.1 ist also gezeigt, dass keine deterministische Strategie $S \in \Sigma$ auf vollständigen (m, n) -Gittergraphen besser als c -kompetitiv ist, mit

$$c = \lfloor \sqrt{k} \rfloor + \frac{1}{2} + \frac{1}{2\lfloor \sqrt{k} \rfloor}.$$

Damit ist Satz 4.3 bewiesen. □

In den oben beschriebenen Szenarien zum Beweis der unteren Schranke bedient sich ein Gegenspieler, falls k keine Quadratzahl ist, stets weniger als k ausfallender Kanten. So gesehen nutzt er seine Möglichkeiten nicht immer vollständig aus. Das hat zur Folge, dass für bestimmte k , insbesondere wenn k dicht an der nächst größeren Quadratzahl liegt, Szenarien existieren, in denen die untere Schranke größer ist als in Satz 4.3 angegeben. Da die untere Schranke jedoch maßgeblich von den Dimensionen des Graphen bezüglich k abhängt, lässt sich für beliebige k nur sehr schwer ein allgemeingültiges worst-case-Szenario angeben. Zudem ist das Ergebnis aus Satz 4.3 auch für ungünstige k (wenn k dicht an der nächst größeren Quadratzahl liegt) nicht weit von der jeweiligen größten, uns bekannten unteren Schranke entfernt. Für den Fall, dass k eine Quadratzahl ist, ist kein Szenario bekannt, dass eine höhere untere Schranke als die aus Satz 4.3 liefert.

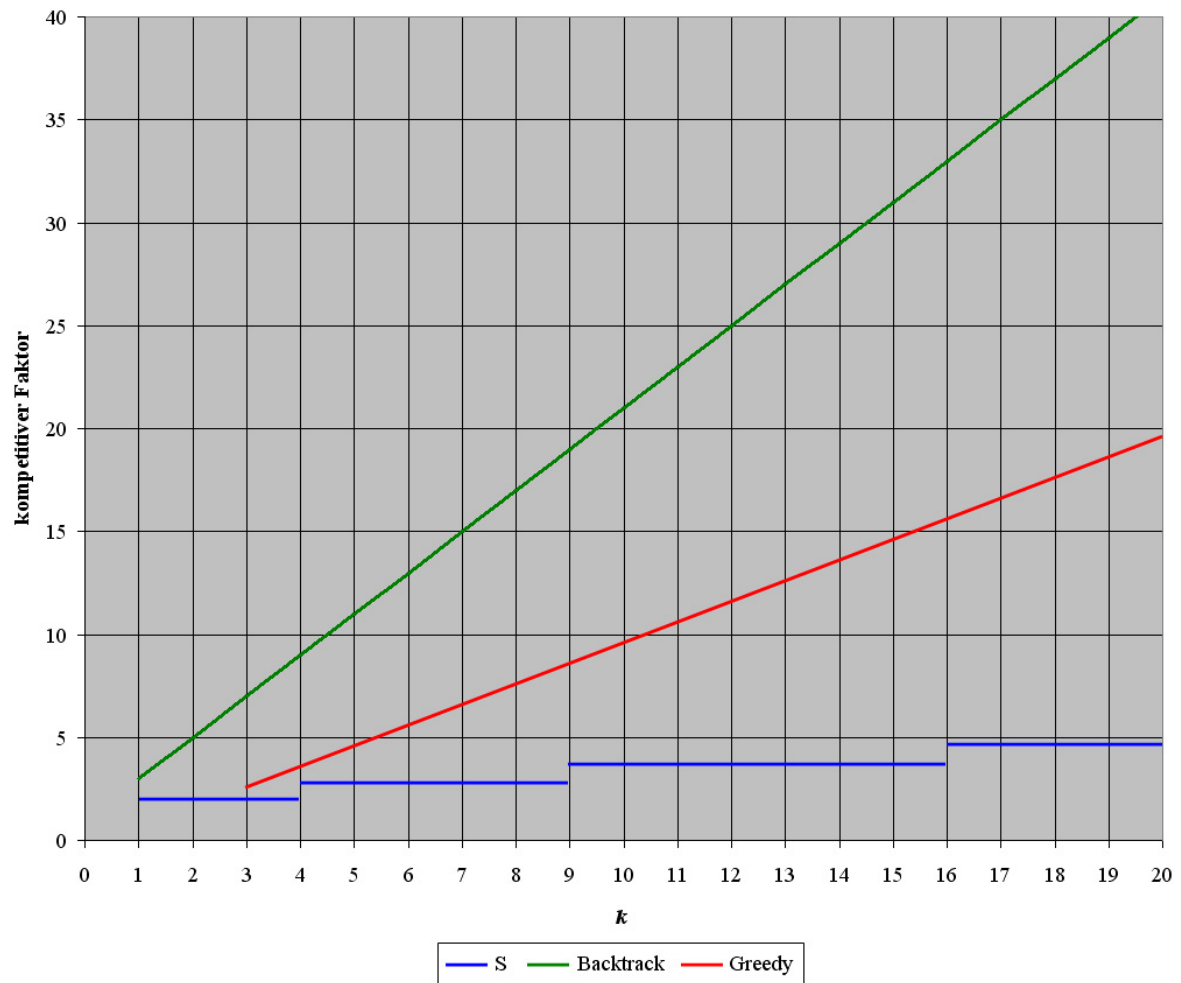
Abschließend wollen wir noch die untere Schranke einer beliebigen deterministischen Strategie $S \in \Sigma$ auf vollständigen (m, n) -Gittergraphen mit den aus Kapitel 3.3.2 bekannten unteren Schranken der Strategien *Backtrack* und *Greedy* auf dieser Graphklasse vergleichen. Dazu betrachten wir Abbildung 4.5. Der grüne Funktionsgraph repräsentiert den kompetitiven Faktor von *Backtrack* $(2k+1)$ in Abhängigkeit von k . Die unteren Schranken der kompetitiven Faktoren von *Greedy* $(k - \frac{1}{2})$ und $S (\lfloor \sqrt{k} \rfloor + \frac{1}{2} + \frac{1}{2\lfloor \sqrt{k} \rfloor})$ sind rot bzw. blau dargestellt. Auffällig ist das konstruktionsbedingte stufenweise Wachstum des blauen Funktionsgraphen. Betrachten wir für k zwei aufeinanderfolgende Quadratzahlen, so ist die untere Schranke von S in diesem Intervall für alle Werte konstant. Asymptotisch betrachtet wächst der Funktionsgraph von S für steigende k zunehmend langsamer als die Funktionsgraphen von *Backtrack* und *Greedy*.

4.4 Allgemeiner (m, n) -Gittergraph

In diesem Abschnitt wenden wir uns wieder den aus Kapitel 3.3.3 bekannten allgemeinen (m, n) -Gittergraphen zu. Für jede beliebige deterministische Strategie gilt auf dieser Graphklasse der folgende Satz:

Satz 4.4. *Keine eindeutige deterministische Strategie ist auf allgemeinen (m, n) -Gittergraphen mit k ausfallenden Kanten besser als $(2k + 1)$ -kompetitiv.*

Beweis. Es sei Π das k -Canadian Traveller Problem auf allgemeinen (m, n) -Gittergraphen, Σ die Menge aller deterministischen Strategien, die jede Instanz $P \in \Pi$ korrekt lösen und $k \in \mathbb{N}$ die Anzahl maximal auftretender Blockaden. Um eine untere Schranke des kompetitiven Faktors einer beliebigen deterministischen Strategie $S \in \Sigma$ zum Lösen der Problemklasse Π anzugeben, konstruieren wir für jedes k eine worst-case-Instanz $P_k \in \Pi$. Gegeben sei der gewichtete allgemeine (m, n) -Gittergraph $G = (V, E)$ aus Abbildung 4.6, mit $m = k + 1$, $n = \frac{1}{\varepsilon} + 2$ und $|e| = \varepsilon > 0$, für alle $e \in E$. Abbildung 4.6 zeigt darüber hinaus die Knoten $u_i, u'_i, v_i, w_i \in V$, für $i \in \{1, \dots, k + 1\}$, wobei Start- und Zielknoten durch $s := u_1$ und $t := w_1$ gegeben seien. Die maximale Anzahl ausfallender Kanten sei $k \in \mathbb{N}$. Für die Länge der Pfade $\pi_i(u_i, v_i) := (u_i, u'_i, \dots, v_i)$, mit $i \in \{1, \dots, k + 1\}$,

Abbildung 4.5: Kompetitivität von S , *Backtrack* und *Greedy* im Vergleich

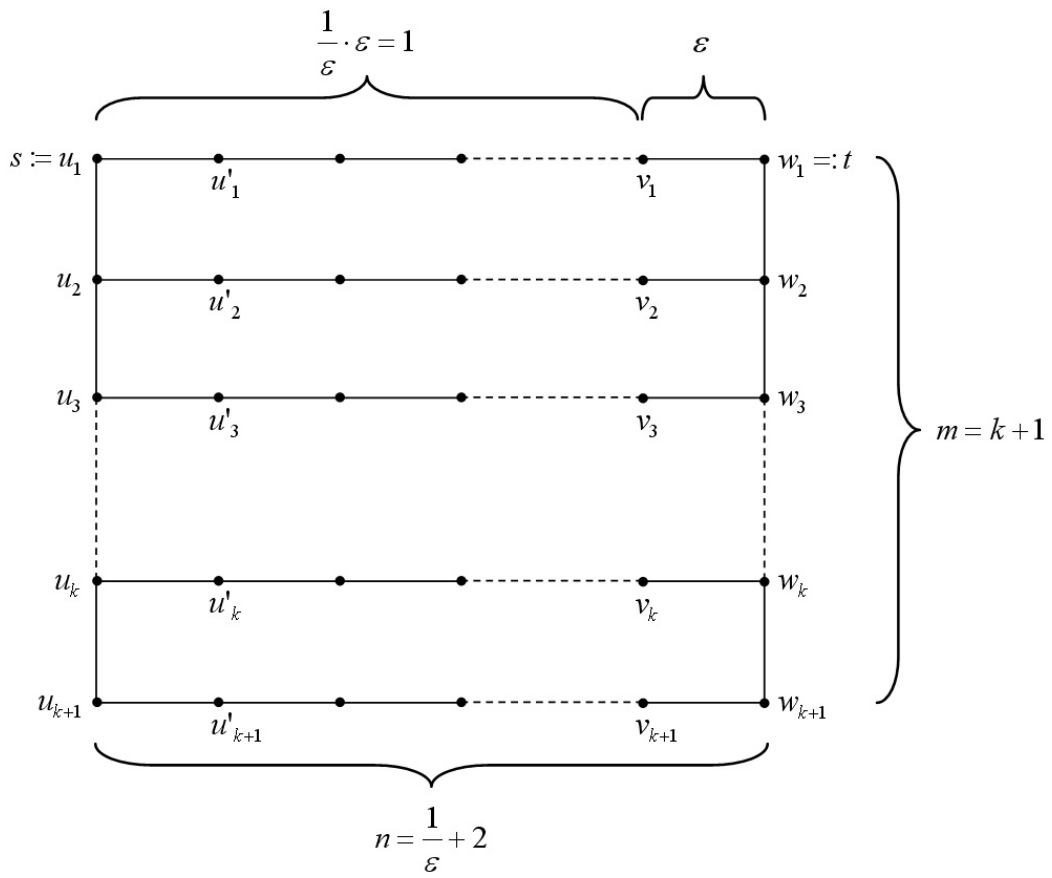


Abbildung 4.6: worst-case-Szenario für eine beliebige deterministische Strategie

gilt:

$$|\pi_i(u_i, v_i)| = 1.$$

Wir können nun davon ausgehen, dass eine beliebige deterministische Strategie S zunächst versucht, sich über einen der Pfade $(s, \dots, u_i) \circ \pi_i$ dem Zielknoten t zu nähern. Dieser Versuch endet allerdings an Knoten v_i , da ein Gegenspieler die Strategie S kennt und durch Blockieren der Kante $\{v_i, w_i\}$ darauf reagiert. Würde S nicht dem Pfad $(s, \dots, u_i) \circ \pi_i$ bis v_i folgen, sondern bereits zu einem früheren Zeitpunkt umkehren, so würden sich die Wegkosten von S höchstens vergrößern, da die Strategie dann bereits Kosten verursacht hätte, ohne eine noch unbekannte Blockade entdeckt zu haben. Erfährt die Strategie S an Knoten v_i von der ausfallenden Kante, können wir annehmen, dass sie auf dem kürzesten Weg zum Knoten u_i zurückkehrt. Denn unabhängig von der weiteren Vorgehensweise der Strategie ist zu diesem Zeitpunkt bereits bekannt, dass ein freier Weg von v_i nach t in jedem Fall über u_i führt. Wenn wir im Sinne der Strategie S annehmen, dass sie keinen Pfad $(u_i, u'_i, \dots, t)$ ein zweites Mal betritt, von dem sie bereits weiß, dass er blockiert ist, kann ein Gegenspieler die Strategie nach obigem Schema genau k mal in eine Sackgasse locken. Danach sind S alle ausfallenden Kanten bekannt und ein Gegenspieler hat keine Möglichkeit mehr die Strategie auf ihrem Weg nach t zu hindern. Dabei minimiert S die Wegkosten, wenn sie die insgesamt $k + 1$ Pfade $\pi_i(u_i, v_i)$ beispielsweise in der Reihenfolge $i = 1, \dots, k + 1$ besucht.

Unabhängig davon, nach welcher Vorgehensweise eine Strategie S eine Lösung der Problem Instanz P_k ermittelt, gilt für die von S verursachten Kosten:

$$S(P_k) \geq 2k + 1 + (2k + 1)\varepsilon \geq 2k + 1.$$

Die Kosten einer optimalen Lösung lassen sich wie folgt abschätzen:

$$\text{OPT}(P_k) \leq 1 + (2k + 1)\varepsilon.$$

Für jedes $\delta > 0$ existiert ein $\varepsilon > 0$, so dass gilt:

$$\begin{aligned} \forall k \in \mathbb{N} \exists P_k \in \Pi : \frac{S(P_k)}{\text{OPT}(P_k)} &\geq \frac{2k + 1}{1 + (2k + 1)\varepsilon} \\ &\geq 2k + 1 - \delta. \end{aligned}$$

In obiger Ungleichung können die Kosten einer optimalen Lösung beliebig hoch gewählt werden. Dazu skalieren wir einfach alle Kantengewichte mit demselben großen Faktor. Nach Lemma 3.1 ist also bewiesen, dass jede additive Konstante $\alpha > 0$ ausgestochen werden kann, womit keine deterministische Strategie $S \in \Sigma$ auf allgemeinen (m, n) -Gittergraphen besser als $(2k + 1 - \delta)$ -kompetitiv ist. Da der Wert für δ jedoch beliebig nah an 0 gewählt werden kann, existiert sogar keine Strategie, die auf allgemeinen (m, n) -Gittergraphen besser als $(2k + 1)$ -kompetitiv ist.

□

Weil Satz 4.4 für jede beliebige deterministische Strategie gilt, kennen wir mit *Backtrack* bereits eine bezüglich des kompetitiven Faktors im worst-case optimale Online-Strategie zum Lösen des k -Canadian Traveller Problem auf allgemeinen (m, n) -Gittergraphen. Das verschafft uns gleichzeitig die Gewissheit, in Satz 4.4 die größtmögliche untere Schranke für diese Graphklasse angegeben zu haben.

4.5 t -Spanner

In diesem Abschnitt geht es uns um die Bestimmung einer unteren Schranke für den kompetitiven Faktor einer beliebigen deterministischen Strategie zum Lösen des k -Canadian Traveller Problem. Dabei fordern wir für den Restgraph, der durch Streichen von maximal k Kanten entsteht, dass er ein t -Spanner seines Ursprungsgraphen ist, vergleiche Definition 3.15 auf Seite 65.

In Kapitel 3.4.1 haben wir bereits die beiden Strategien *Backtrack* und *Greedy* auf dieser Graphklasse untersucht und herausgefunden, dass *Backtrack* für $t > 1$ genau

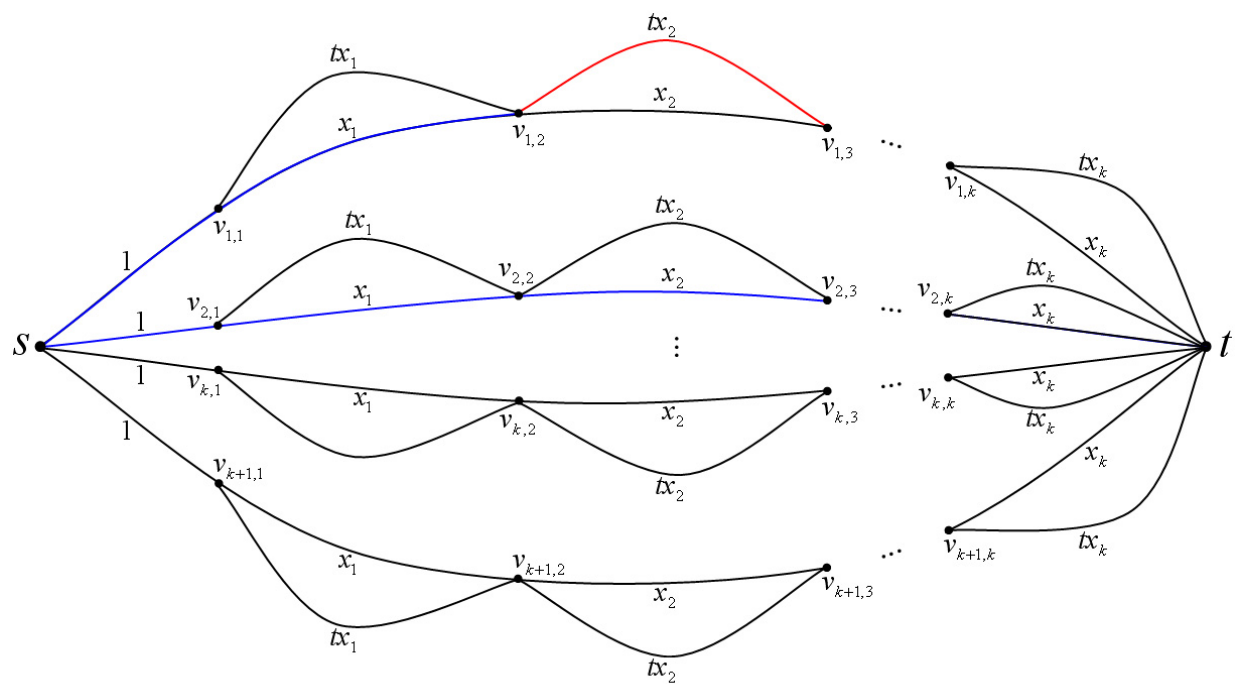
$$(2k + 1)\text{-kompetitiv}$$

und *Greedy* für $t \geq 1$ genau

$$\left(2 \left(\frac{2t}{t+1}\right)^k - 1\right)\text{-kompetitiv}$$

ist. Der Wert für *Backtrack* liefert uns die Erkenntnis, dass eine untere Schranke für den kompetitiven Faktor einer beliebigen Strategie selbst für beliebig große t keinesfalls $2k + 1$ überschreiten kann. Auf der anderen Seite stellt der Wert für *Greedy* sicher, dass eine untere Schranke für den kompetitiven Faktor einer beliebigen Strategie für $t \rightarrow 1$ in jedem Fall gegen 1 strebt. Dies sind die vorab bekannten Rahmenbedingungen.

Es sei Π das k -Canadian Traveller Problem auf allgemeinen Graphen, in denen Kanten ausschließlich der Art entfernt werden, dass die entstehenden Restgraphen t -Spanner ihrer Ursprungsgraphen sind. Mit Σ bezeichnen wir die Menge aller deterministischen Strategien, die jede Instanz $P \in \Pi$ korrekt lösen und mit $k \in \mathbb{N}$ die Anzahl maximal ausfallender Kanten. Um eine untere Schranke des kompetitiven Faktors einer beliebigen deterministischen Strategie $S \in \Sigma$ zum Lösen der Problemklasse Π anzugeben, konstruieren wir für jedes k eine worst-case-Instanz $P_k \in \Pi$. Wir beschreiben zunächst den Aufbau der Szenarien und erläutern welche Gleichungen die Kantenlängen erfüllen müssen, bevor wir zur Berechnung dieser Werte übergehen. Dazu betrachten wir Abbildung 4.7. Der dort dargestellte Graph $G = (V, E)$ spaltet sich am Startknoten $s \in V$ in $k + 1$ kürzeste Pfade $\pi_i(s, t)$ zum Zielknoten $t \in V$. Der i -te Pfad $\pi_i(s, t)$ beginnt in s , endet in t und führt über die Knoten $v_{i,j} \in V$, mit $j \in \{1, \dots, k + 1\}$ und $v_{i,k+1} = t$. Wie wir später sehen, blockiert ein Gegenspieler stets k der insgesamt $k(k + 1)$ Kanten $\{v_{i,j}, v_{i,j+1}\}$, mit $i \in \{1, \dots, k + 1\}$ und



$$x_j = \frac{2}{t+1} \left(\frac{t+1}{t-1} \right)^j \quad \boxed{tx_j} = w_j = 2(x_{j-1} + \dots + x_1 + 1) + x_j$$

Abbildung 4.7: worst-case-Szenario einer beliebigen deterministischen Strategie

$j \in \{1, \dots, k\}$. Damit der Restgraph $G' := (V, E')$ in jedem Fall ein t -Spanner von G ist, fügen wir in G für jede potentielle Ausfallkante $\{v_{i,j}, v_{i,j+1}\}$ der Länge x_j eine alternative Kante der Länge tx_j ein. Alle Kanten zwischen s und $v_{i,1}$, mit $i \in \{1, \dots, k+1\}$, haben die Länge $|\{s, v_{i,1}\}| = 1$. Für die übrigen Kantenlängen $x_1, \dots, x_k$ gilt:

$$tx_j = w_j, \text{ mit } w_j := 2(x_{j-1} + \dots + x_1 + 1) + x_j. \quad (1)$$

Um auf dem Pfad $\pi_i(s, t)$ von einem Knoten $v_{i,j}$ zum Knoten $v_{i,j+1}$ über die Kante der Länge tx_j zu gelangen, verursacht also höchstens die gleichen Kosten wie die Traversierung eines Pfades $(v_{i,j}, \dots, s, \dots, v_{i',j+1})$ von $v_{i,j}$ nach $v_{i',j+1}$, der auf dem Pfad $\pi_{i'}(s, t)$ liegt. Abbildung 4.7 veranschaulicht diesen Zusammenhang am Beispiel $tx_2 = w_2$ (die rote Kante hat genau die Länge des blauen Pfades).

Zunächst einmal werden wir die Parameter $x_1, \dots, x_k$ berechnen. Für x_1 gilt nach Gleichung (1):

$$\begin{aligned} tx_1 &= w_1 \\ \Leftrightarrow tx_1 &= 2 + x_1 \\ \Leftrightarrow x_1 &= \frac{2}{t-1}. \end{aligned}$$

Mit Hilfe von x_1 könnten wir nun auch x_2 bestimmen und so weiter. Deutlich einfacher wäre allerdings eine geschlossene Formel für x_j . Dann könnten wir direkt x_j berechnen, ohne zuvor $x_1, \dots, x_{j-1}$ berechnet zu haben. Neben Gleichung (1) existiert zwischen x_j und w_j auch der folgende Zusammenhang:

$$tx_{j+1} = x_j + w_j + x_{j+1}. \quad (2)$$

Dieser Zusammenhang folgt aus (1) und lässt sich an Abbildung 4.7 nachvollziehen. Wegen Gleichung (1) können wir in Gleichung (2) auf der rechten Seite w_j durch tx_j ersetzen:

$$tx_{j+1} = x_j + tx_j + x_{j+1}.$$

Durch Umformung erhalten wir für x_{j+1} folgende Rekursion:

$$x_{j+1} = \frac{t+1}{t-1}x_j.$$

Mit Hilfe des zuvor berechneten Startwertes für x_1 können wir die Rekursion in folgende geschlossene Formel überführen:

$$x_j = \frac{2}{t+1} \left(\frac{t+1}{t-1} \right)^j.$$

Eine beliebige deterministische Strategie $S \in \Sigma$ muss sich nun für einen der Pfade $\pi_i(s, t)$ entscheiden und diesem zunächst bis Knoten $v_{i,1}$ folgen. Ein Gegenspieler wird dann die Kante $\{v_{i,1}, v_{i,2}\}$ der Länge x_1 blockieren. Die Strategie S hat dann zwei Möglichkeiten. Entweder sie traversiert die noch freie Kante $\{v_{i,1}, v_{i,2}\}$ der Länge tx_1 , um den Knoten $v_{i,2}$ zu erreichen, oder sie tritt den Rückweg nach s an und wählt einen alternativen Pfad $\pi_{i'}(s, t)$, mit $i' \neq i$, und folgt diesem ungehindert bis zum Knoten $v_{i',2}$. Die Kosten für S , um von $v_{i,1}$ nach $v_{i',2}$ zu gelangen, würden mindestens $w_1 = 2 + x_1 = tx_1$ betragen. Für $i = i'$ würde S offensichtlich unnötige Mehrkosten verursachen. Da ein Gegenspieler bedacht ist, die Kanten der Länge x_j so spät wie möglich zu blockieren, lässt er die Kante $\{v_{i',1}, v_{i',2}\}$ der Länge x_1 frei und blockiert erst die Kante $\{v_{i',2}, v_{i',3}\}$ der Länge x_2 , sobald S den Knoten $v_{i',2}$ erreicht. Wegen $x_1 < x_2 < \dots < x_k$ zahlt sich diese Gegenspieler-Strategie aus, denn je länger eine blockierte Kante ist, desto mehr Kosten wird S verursachen. Wichtig ist jedoch für einen Gegenspieler, Kanten so zeitig zu blockieren, dass alle k Blockaden bereits eingesetzt wurden, bevor S den Zielknoten t erreicht. Würde die Strategie S im Pfad $\pi_{i'}(s, t)$ nicht bis $v_{i',2}$ vorrücken, würde sie Kosten verursachen, ohne neue Informationen zu erhalten, also ohne eine neue Blockade zu entdecken. Die Strategie S würde dann insgesamt höchstens höhere Kosten verursachen. Wir können nun also davon ausgehen, dass sich S an einem der Knoten $v_{i,2}$ oder $v_{i',2}$ befindet. Darauf reagiert ein Gegenspieler durch Blockieren der betreffenden Kante $\{v_{i,2}, v_{i,3}\}$ oder $\{v_{i',2}, v_{i',3}\}$ der Länge x_2 . Von $v_{i',2}$ aus gibt es für die Strategie S wieder die zwei hinsichtlich der Kosten gleichwertigen Möglichkeiten, die S bereits zu Beginn an Knoten $v_{i,1}$ hatte: Entweder sie traversiert die noch freie Kante $\{v_{i',2}, v_{i',3}\}$ der Länge tx_2 , um den Knoten $v_{i',3}$ zu erreichen, oder sie tritt den Rückweg zum Startknoten an, wählt einen alternativen Pfad $\pi_{i''}(s, t)$, mit $i'' \notin \{i', i\}$, und folgt diesem ungehindert bis zum Knoten $v_{i'',3}$. Für $i'' = i'$ oder $i'' = i$ würde S auch hier offensichtlich wieder unnötige Mehrkosten verursachen, da beide Pfade bereits blockierte Kanten enthalten. Möchte die Strategie S ihre Kosten minimieren, so gibt es von $v_{i,2}$ nur noch die Möglichkeit dem Pfad $\pi_i(s, t)$ weiter bis ins Ziel zu folgen. Der Grund dafür ist, dass die Kante $\{v_{i,1}, v_{i,2}\}$ blockiert ist und S damit um die Möglichkeit gebracht wird, von einem Knoten $v_{i,j}$ mit Kosten w_j zu einem Knoten $v_{i''',j}$, mit $i''' \neq i$, zu gelangen. Der Pfad $\pi_i(s, t)$ hätte dann allerdings die Länge $1 + tx_1 + tx_2 + \dots + tx_k$, da ein Gegenspieler die Kanten mit den Längen $x_1, x_2, \dots, x_k$ blockieren würde. Auch wenn die Strategie S statt von $v_{i,2}$ von $v_{i',2}$ wie oben angedeutet ihre Traversierung fortsetzt, verursacht S mindestens die Kosten $1 + tx_1 + tx_2 + \dots + tx_k$, um das Ziel zu erreichen. Denn dann liegt für die Strategie S bestenfalls einer der beiden nachfolgend beschriebenen Fälle vor.

Für eine Strategie gibt es genau zwei weitere Möglichkeiten, mit den Kosten $1 + tx_1 + tx_2 + \dots + tx_k$ von s nach t zu gelangen. Zum einen könnte eine Strategie S zunächst k verschiedenen Pfaden $\pi_i(s, t)$, mit $i \in \{1, \dots, k\}$, solange folgen, bis eine blockierte Kante entdeckt wird und dann zu s zurückkehren. Der $(k+1)$ -te Pfad kann dann keine Blockade mehr enthalten und führt die Strategie schließlich ohne weitere Umwege ins Ziel. Insgesamt würde S dann die Kosten $1 + w_1 + w_2 + \dots + w_k$ verursachen, die wegen Gleichung (1) den Kosten $1 + tx_1 + tx_2 + \dots + tx_k$ entsprechen. Alternativ könnte eine Strategie S zunächst

$j < k$ verschiedenen Pfaden $\pi_i(s, t)$, mit $i \in \{1, \dots, j\}$, bis zur Entdeckung einer blockierten Kante folgen, dann zu s zurückzukehren und dem Pfad $\pi_{j+1}(s, t)$ bis zum Zielknoten folgen. Da die Kanten $\{v_{j+1,j+1}, v_{j+1,j+2}\}, \dots, \{v_{j+1,k}, v_{j+1,k+1}\}$ mit den Längen $x_{j+1}, \dots, x_k$ blockiert sind, würde S dann die Kosten $1 + w_1 + w_2 + \dots + w_j + tx_{j+1} + \dots + tx_k$ verursachen, die wegen Gleichung (1) ebenfalls den Kosten $1 + tx_1 + tx_2 + \dots + tx_k$ entsprechen. Alle übrigen Vorgehensweisen einer Strategie würden insgesamt höhere Kosten verursachen.

Unabhängig davon, wie eine Strategie S ihre Lösung einer Problem Instanz $P_k \in \Pi$ ermittelt, existiert in G' immer mindestens ein Pfad von s nach t mit der Länge:

$$\begin{aligned}
 \text{OPT}(P_k) &= 1 + x_1 + x_2 + \dots + x_k \\
 &= \frac{w_k - x_k}{2} + x_k && \text{nach Definition von } w_k \\
 &= \frac{(t-1)x_k}{2} + x_k && \text{nach (1)} \\
 &= \frac{(t+1)x_k}{2} \\
 &= \frac{(t+1) \frac{2}{t+1} \left(\frac{t+1}{t-1}\right)^k}{2} \\
 &= \left(\frac{t+1}{t-1}\right)^k.
 \end{aligned}$$

Die von der Strategie S verursachten Kosten lassen sich wie folgt abschätzen:

$$\begin{aligned}
 S(P_k) &\geq 1 + tx_1 + tx_2 + \dots + tx_k \\
 &= t \cdot (\text{OPT}(P_k) - 1) + 1 \\
 &= t \cdot \text{OPT}(P_k) - (t-1) \\
 &= t \cdot \left(\frac{t+1}{t-1}\right)^k - (t-1).
 \end{aligned}$$

Wir erhalten folgende untere Schranke für den kompetitiven Faktor einer beliebigen deterministischen Strategie S :

$$\begin{aligned}
 \forall k \in \mathbb{N} \exists P_k \in \Pi : \frac{S(P_k)}{\text{OPT}(P_k)} &\geq \frac{t \cdot \left(\frac{t+1}{t-1}\right)^k - (t-1)}{\left(\frac{t+1}{t-1}\right)^k} \\
 &= t - \frac{t-1}{\left(\frac{t+1}{t-1}\right)^k} \\
 &= t - (t-1) \left(\frac{t-1}{t+1}\right)^k.
 \end{aligned}$$

In obiger Ungleichung können die Kosten einer optimalen Lösung beliebig hoch gewählt werden. Dazu skalieren wir einfach alle Kantengewichte mit demselben großen Faktor. Daher folgt aus Lemma 3.1 die Korrektheit folgender Aussage:

Satz 4.5. *Werden in einem ungerichteten, gewichteten Graphen G maximal k Kanten der Art entfernt, dass der Restgraph G' ein t -Spanner von G ist, so ist keine eindeutige deterministische Strategie auf G für alle $t \geq 1$ besser als c -kompetitiv mit*

$$c = t - (t - 1) \left(\frac{t - 1}{t + 1} \right)^k.$$

Nachfolgendes Korollar hebt die Aussage von Satz 4.5 noch einmal für zwei interessante Stretch-Faktoren t hervor:

Korollar 4.6. *Werden in einem ungerichteten, gewichteten Graphen G maximal k Kanten so entfernt, dass der Restgraph G' ein t -Spanner von G ist, so ist keine deterministische Strategie $S \in \Sigma$ auf G besser als*

- $(2 - \frac{1}{3^k})$ -kompetitiv für $t = 2$,
- $(2k + 1)$ -kompetitiv für „ t beliebig groß“.

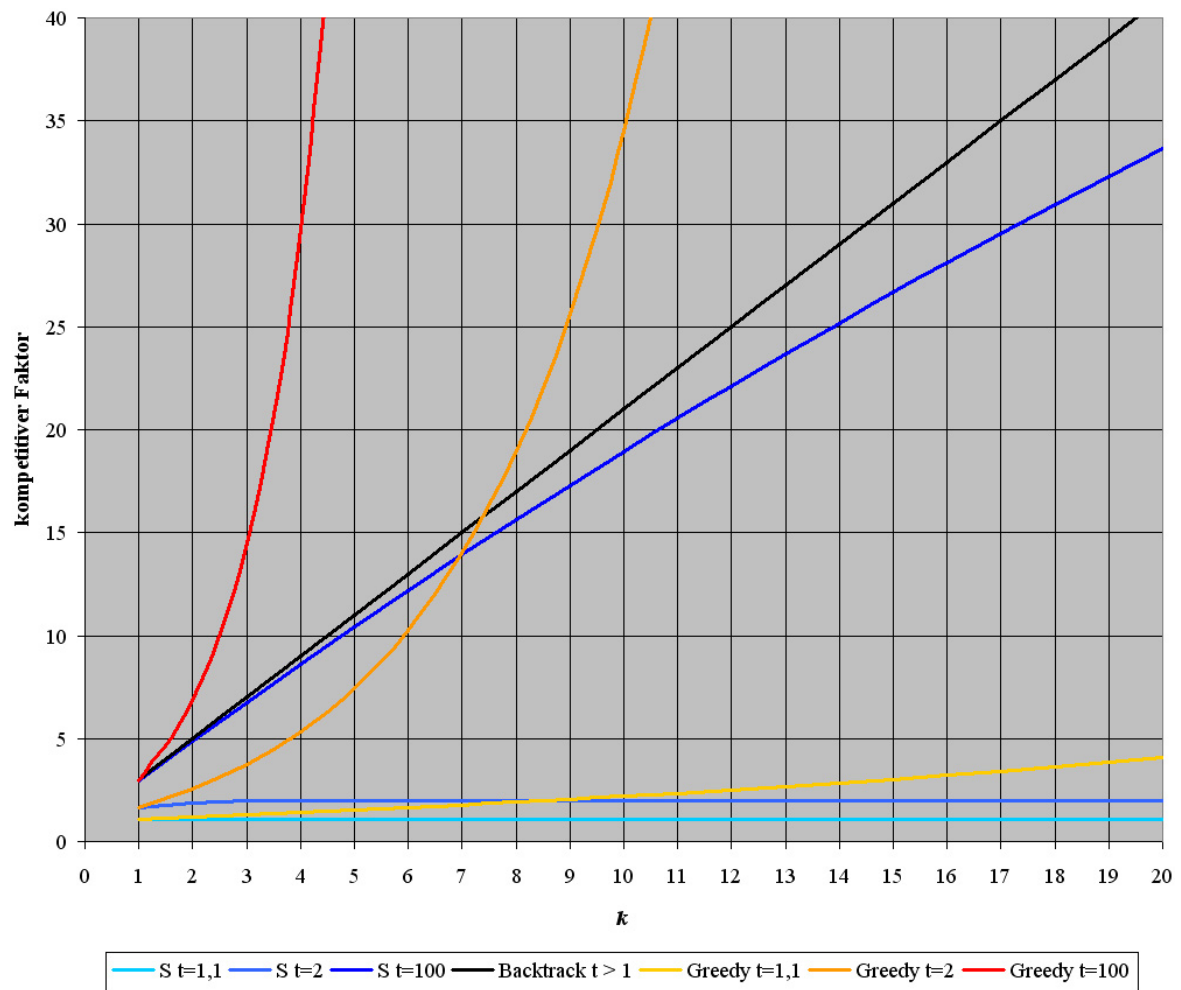
Der Grenzwert $2k + 1$, für $t \rightarrow \infty$, ist dem Bruch aus Satz 4.5 auf den ersten Blick nicht anzusehen. Das Ergebnis leitet sich wie folgt her:

$$\begin{aligned}
& t - (t - 1) \left(\frac{t - 1}{t + 1} \right)^k \\
&= t - \frac{(t - 1)^{k+1}}{(t + 1)^k} \\
&= \frac{t(t + 1)^k - (t - 1)^{k+1}}{(t + 1)^k} \\
&= \frac{t \left[\sum_{i=0}^k \binom{k}{i} \cdot t^{k-i} \cdot 1^i \right] - \left[\sum_{i=0}^{k+1} \binom{k+1}{i} \cdot t^{k+1-i} \cdot (-1)^i \right]}{\sum_{i=0}^k \binom{k}{i} \cdot t^{k-i} \cdot 1^i} \\
&= \frac{t \left[\binom{k}{0} t^k \cdot 1^0 + \binom{k}{1} t^{k-1} \cdot 1^1 + \dots \right] - \left[\binom{k+1}{0} t^{k+1} \cdot (-1)^0 + \binom{k+1}{1} t^k \cdot (-1)^1 + \dots \right]}{\binom{k}{0} t^k \cdot 1^0 + \binom{k}{1} t^{k-1} \cdot 1^1 + \dots} \\
&= \frac{t [t^k + k t^{k-1} + \dots] - [t^{k+1} - (k + 1) t^k + \dots]}{t^k + k t^{k-1} + \dots} \\
&= \frac{t^{k+1} + k t^k + \dots - t^{k+1} + (k + 1) t^k + \dots}{t^k + k t^{k-1} + \dots} \\
&= \frac{k t^k + k t^k + t^k + \dots}{t^k + k t^{k-1} + \dots} \\
&\rightarrow \frac{k + k + 1}{1}, \text{ für } t \rightarrow \infty \\
&= 2k + 1.
\end{aligned} \tag{3}$$

Den Grenzwert in Zeile (3) erhalten wir durch Koeffizientenvergleich der Terme mit den höchsten Exponenten. Alle in der Herleitung durch (...) angedeuteten Terme haben kleinere Exponenten als k und sind somit für die Grenzwertberechnung nicht von Bedeutung. Damit deckt sich der Wertebereich der unteren Schranke mit den zu Beginn dieses Abschnitts angestellten Überlegungen.

Bei genauerer Betrachtung der unteren Schranke aus Satz 4.5 fällt neben den oben genannten Eigenschaften auf, dass der Grenzwert für $k \rightarrow \infty$ gegen t strebt. Daraus ergibt sich, dass die angegebene untere Schranke für kein k oberhalb des Stretch-Faktors t liegt. Wir vermuten jedoch, dass das nicht immer zwingend der Fall sein muss und sich, zumindest für spezielle t -Werte, Szenarien finden lassen, die eine größere untere Schranke als t liefern.

Wir wollen abschließend das Ergebnis aus Satz 4.5 mit den beiden aus Kapitel 3.4.1 bekannten kompetitiven Faktoren der Strategien *Backtrack* und *Greedy* auf dieser Graphklasse vergleichen. Dazu betrachten wir Abbildung 4.8. Dieser Grafik lassen sich die unteren Schranken der kompetitiven Faktoren (Y-Achse) von S , *Backtrack* und *Greedy* für $k \in \{1, \dots, 20\}$ (X-Achse) und $t \in \{1.1, 2, 100\}$ entnehmen. Da der Funktionsgraph von *Backtrack* für alle $t > 1$ gleich ist, taucht er in der Grafik nur einmal auf. Weil die Funktionsgraphen zu S eine untere Schranke der kompetitiven Faktoren beliebiger deterministischer Strategien darstellen, ist klar, dass weder der *Backtrack*-, noch der *Greedy*-Funktionsgraph für korrespondierende t jemals unterhalb der zu S gehörenden Funktionsgraphen liegen kann. Für $t \rightarrow \infty$ strebt die untere Schranke für den kompetitiven Faktor einer beliebigen Strategie S gegen $2k + 1$, also gegen den kompetitiven Faktor der *Backtrack*-Strategie. Dieser Zusammenhang lässt sich anhand der Abbildung 4.8 bereits vermuten, denn der Funktionsgraph von S nähert sich schon für $t = 100$ stark dem *Backtrack*-Funktionsgraph an. Umgekehrt strebt die untere Schranke für den kompetitiven Faktor einer beliebigen Strategie S für $t \rightarrow 1$ gegen 1, und das natürlich schneller als der kompetitive Faktor der *Greedy*-Strategie. Dieser Zusammenhang lässt sich an den Funktionsgraphen für $t = 1, 1$ ablesen. Während sich der Funktionsgraph zu S für alle dargestellten k bereits sehr nah an 1 bewegt, lässt sich für *Greedy* mit steigendem k noch relativ deutlich ein Wachstum des kompetitiven Faktors feststellen.

Abbildung 4.8: Kompetitivität von S , Backtrack und Greedy in Abhängigkeit von t und k

4.6 (t, l) -Self-Spanner

In diesem Abschnitt werden wir eine untere Schranke für den kompetitiven Faktor einer beliebigen deterministischen Strategie auf der Graphklasse der (t, l) -Self-Spanner angeben. Es ist nun im Gegensatz zur Graphklasse der t -Spanner aus vorigem Abschnitt für jeden Teilgraphen $G' = (V, E') \subseteq G$, mit $E' \subseteq E$ und $|E'| \geq |E| - l$, ein Stretch-Faktor t einzuhalten. Hierzu können wir wieder auf den Trick der Kantenvervielfachung zurückgreifen. Zum besseren Verständnis wird dies in Abbildung 4.9 dargestellt.

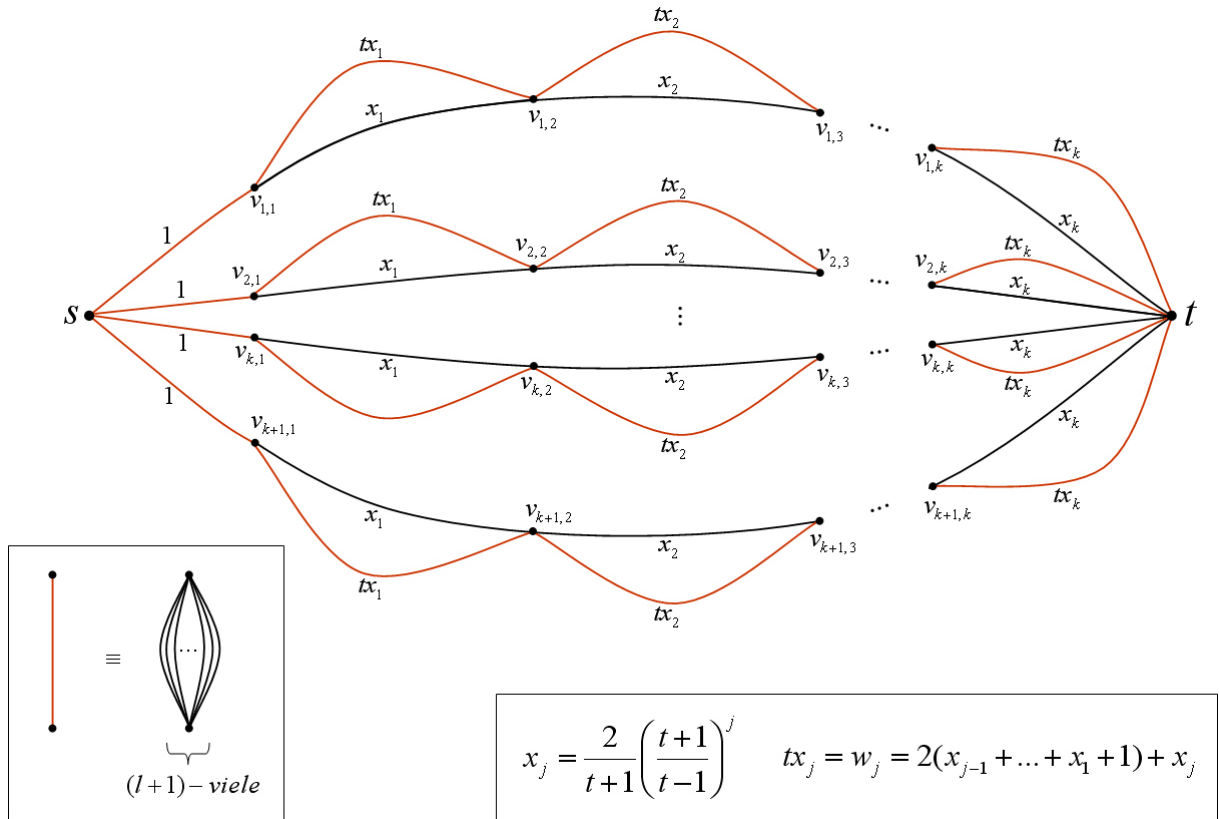


Abbildung 4.9: worst-case-Szenario einer beliebigen deterministischen Strategie auf (t, l) -Self-Spannern

Wir betrachten erneut mit Π das k -Canadian Traveller Problem. Für jedes $k \in \mathbb{N}$ können wir eine Instanz $P_k \in \Pi$ angeben, so dass eine beliebige deterministische Strategie S folgende Kosten nicht unterschreiten kann. Die Konstruktion des Graphen erfolgt sehr nah an der des t -Spanners aus Kapitel 4.5. Wir werden den Graphen gerade soweit abändern, dass aus dem t -Spanner ein (t, l) -Self-Spanner entsteht. Die Strategie wird zu den exakt gleichen oder höchstens längeren Wegen gezwungen, weshalb wir dieselbe untere Schranke wie für

t -Spanner angeben können. Wir vervielfachen zuerst die Kanten, deren Blockierung ohnehin nicht vorgesehen ist. Dazu müssen am Startknoten jeweils l Kanten, mit $|\{s, v_{i,1}\}| := 1$, für alle $i \in \{1, \dots, k+1\}$, eingefügt werden.

Nun betrachten wir den Graphen G aus Kapitel 4.5. Zwischen den Knoten $v_{i,j}$ und $v_{i,j+1}$, mit $i \in \{1, \dots, k+1\}$ und $j \in \{1, \dots, k\}$, existiert stets eine Kante mit Kosten $|\{v_{i,j}, v_{i,j+1}\}| := x_j$, sowie eine mit Kosten $|\{v_{i,j}, v_{i,j+1}\}| := t \cdot x_j$. Letztere ist für uns von großem Interesse. In einem (t, l) -Self-Spanner dürfen nämlich l beliebige Kanten ausfallen, so dass die Kosten eines kürzesten Pfades zwischen zwei Knoten $u, v \in V$, die in diesem Graphen verbunden sind, maximal das t -fache betragen, verglichen mit den Kosten eines kürzesten Pfades im Ursprungsgraphen. Dieses Problem wurde bereits dadurch gelöst, dass diese Kanten auf das $(l+1)$ -fache Vorkommen vervielfacht wurden. Damit ist sichergestellt, dass auch nach Wegfall von l beliebigen Kanten keine größeren Umwege entstehen als es durch die Bedingungen eines (t, l) -Self-Spanners gestattet ist. In Abbildung 4.9 ist dies, der Übersichtlichkeit wegen, durch rote Kanten dargestellt, die $(l+1)$ schwarze Kanten mit identischer Gewichtung repräsentieren.

Das durch obige Konstruktion entstandene Szenario bietet nun analoge Möglichkeiten, wie das für die untere Schranke aus Kapitel 4.5, weshalb wir darauf verzichten die exakte Berechnung hier erneut vorzuführen. Für genauere Details verweisen wir daher auf Kapitel 4.5 und präsentieren an dieser Stelle lediglich das Resultat:

Satz 4.7. *Keine eindeutige deterministische Strategie ist auf (t, l) -Self-Spannern mit k ausfallenden Kanten besser als c -kompetitiv mit*

$$c = t - (t-1) \left(\frac{t-1}{t+1} \right)^k.$$

4.7 t -Self-Spanner

In diesem Kapitel beschreiben wir die Konstruktion einer unteren Schranke für eine beliebige deterministische Strategie auf t -Self-Spannern. Ein hoher kompetitiver Faktor in Abhängigkeit von t ist nicht zu erwarten, da die hier betrachtete Graphklasse erhebliche Einschränkungen für die Konstruktion eines Graphen vorsieht. Außerdem ergeben sich bei der Betrachtung einer beliebigen Strategie keinerlei Anhaltspunkte, die ausgenutzt werden könnten, um eine konkrete Strategie zu hohen Kosten zu zwingen. Wir verwenden erneut das Szenario aus Abschnitt 3.4.3 und zeigen so eine untere Schranke für den kompetitiven Faktor, die folgenden Satz stützt:

Satz 4.8. *Keine eindeutige deterministische Strategie ist auf t -Self-Spannern mit $t \geq 3$ bei k ausfallenden Kanten besser als c -kompetitiv mit*

$$c = (2k+1) \frac{t-1}{t+1} + \frac{2}{t+1}.$$

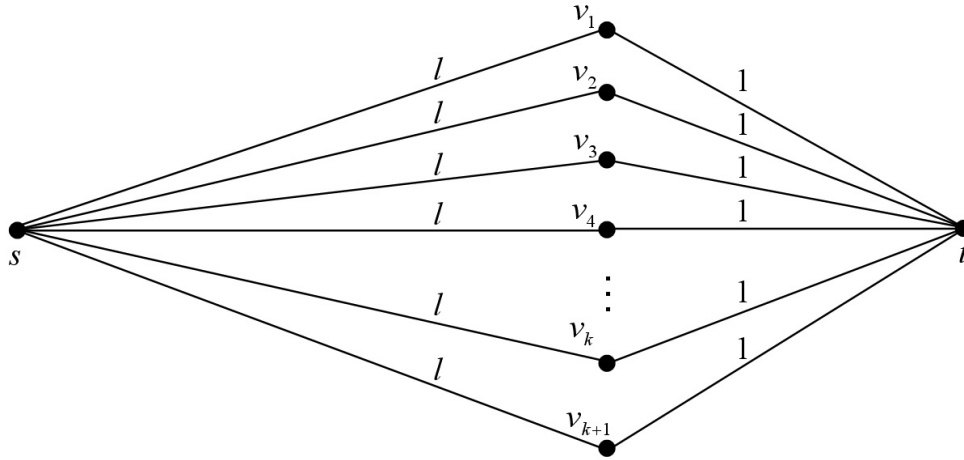


Abbildung 4.10: worst-case-Szenario einer beliebigen deterministischen Strategie auf t -Self-Spannern

Beweis. Die Konstruktion verläuft analog zu der für *Greedy* auf t -Self-Spannern mit $t \geq 3$ aus Abschnitt 3.4.3. Jede beliebige Strategie kann k mal zur Rückkehr zum Startknoten s gezwungen werden und verursacht dabei, die in Abschnitt 3.4.3 ausführlich diskutierten Kosten, die Satz 4.8 beweisen.

□

Für $t = 3$ erhalten wir den erwarteten kompetitiven Faktor von $k + 1$, den die *Greedy*-Strategie erreicht, welche somit auf 3-Self-Spannern unter allen Strategien optimal ist. Mit *Greedy* kennen wir bereits eine Strategie, die auf allen t -Self-Spannern mit $t < 3$ optimal ist. In Abbildung 4.11 ist der Zusammenhang zwischen Stretch-Faktor und kompetitivem Faktor noch einmal graphisch dargestellt. Wie der Graphik zu entnehmen ist, steigt der kompetitive Faktor bereits für marginale Änderungen des Stretch-Faktors signifikant an.

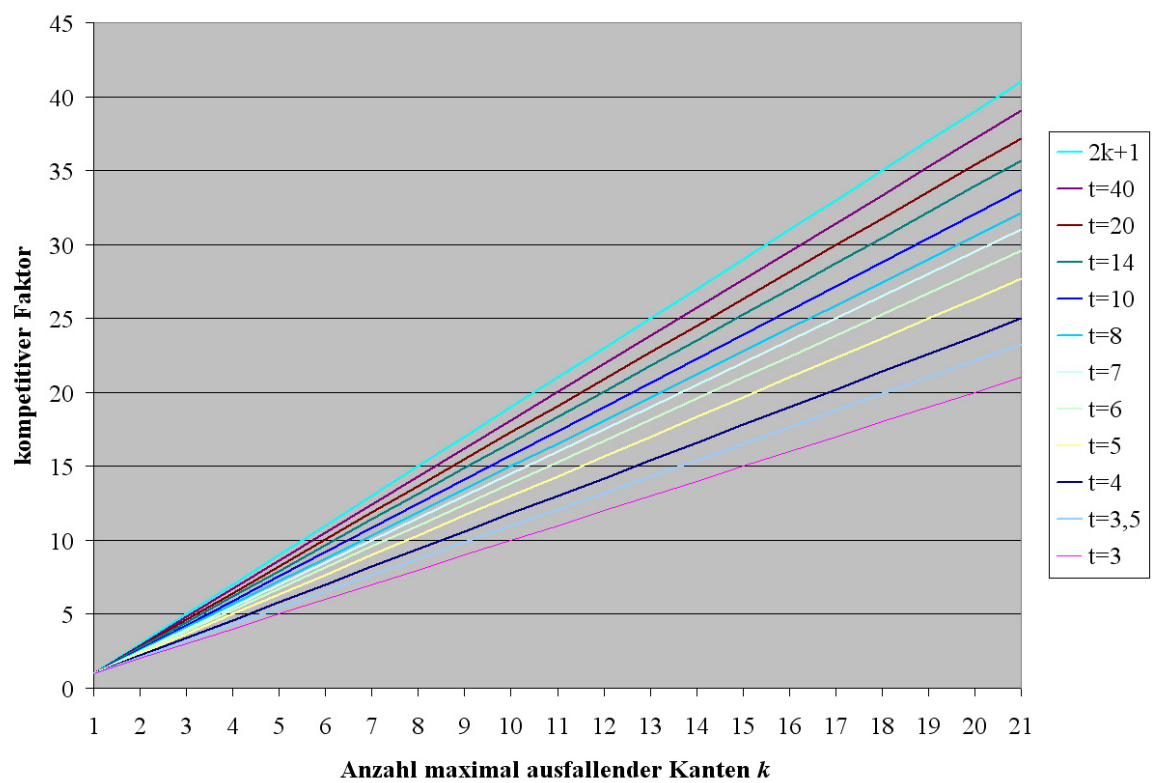


Abbildung 4.11: untere Schranke des kompetitiven Faktors auf t -Self-Spannern

Kapitel 5

Untere Schranke randomisierter Strategien

Ging es in den vorangegangenen Kapiteln darum, eindeutige oder mehrdeutige deterministische Strategien zu untersuchen, so werden wir in diesem Kapitel das k -Canadian Traveller Problem unter Verwendung randomisierter Strategien untersuchen. Da es in der Literatur verschiedenste Formen eines Gegenspielers gibt, wollen wir zuerst den in unserem Zusammenhang verwendeten Begriff etwas genauer formulieren.

Definition 5.1. *Wenn einem Gegenspieler die Strategie bekannt ist, er Auskunft über jeden von der Strategie bisher ausgeführten Elementarschritt hat und er die optimale Offline-Lösung nach vollständiger Antwort der Strategie geben darf, reden wir von einem **adaptiven Offline-Gegner** (adaptive offline adversary).*

Für diesen speziellen Gegner ist nach [BD94] folgender Satz bekannt:

Satz 5.2. (S. Ben-David et al.)

Sei ALG ein c -kompetitiver Online-Algorithmus gegen einen adaptiven Offline-Gegner. Dann existiert ein deterministischer c -kompetitiver Online-Algorithmus.

Mit der unteren Schranke für eine beliebige deterministische Strategie auf allgemeinen Graphen aus Satz 4.1 ergibt sich somit folgendes Korollar:

Korollar 5.3. *Eine beliebige randomisierte Online-Strategie gegen einen adaptiven Offline-Gegner auf allgemeinen Graphen ist nicht c -kompetitiv für $c < 2k + 1$.*

Dem adaptiven Offline-Gegner, der die einzelnen Schritte der Strategie kennt, steht der vergessliche Gegner gegenüber, der seine Entscheidung nur durch Kenntnis über eine Wahrscheinlichkeit für jeden einzelnen Schritt zu treffen hat. Es ist zu erwarten, dass dies eine Einschränkung an den Gegner stellt und die untere Schranke für eine beliebige randomisierte Strategie senken wird. Wir stellen nachfolgend die Ergebnisse von Stephan Westphal vor, der in [SW07] unter anderem eine untere Schranke für randomisierte Strategien auf allgemeinen Graphen gegen vergessliche Gegner untersucht hat.

Definition 5.4. Wenn einem Gegenspieler lediglich die Wahrscheinlichkeit bekannt ist, mit der die Strategie einzelne Entscheidung trifft sprechen wir von einem **vergesslichen Gegner** (oblivious adversary).

Satz 5.5. (Stephan Westphal)

Eine beliebige randomisierte Online-Strategie gegen einen vergesslichen Gegner auf allgemeinen Graphen ist nicht besser als $k + 1$ -kompetitiv.

Westphal benutzt das Szenario aus Abbildung 5.1, um zu zeigen, dass der Erwartungswert des kompetitiven Faktors einer randomisierten Strategie gegen einen vergesslichen Gegner mindestens $k + 1$ beträgt. Um dies zu beweisen benutzt er Yaos Prinzip (vgl.

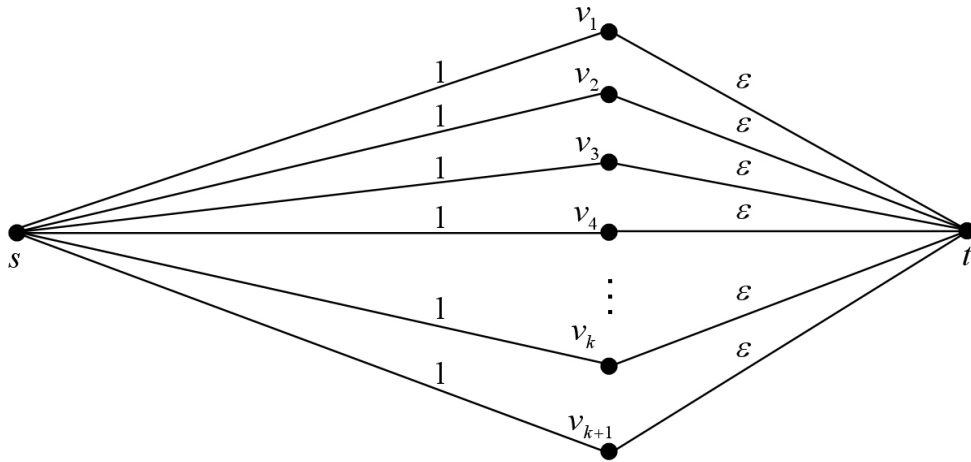


Abbildung 5.1: worst-case-Szenario für eine beliebige randomisierte Strategie

[AY77]. Er gibt hierzu eine Wahrscheinlichkeitsverteilung über Szenarien an, für die kein deterministischer Algorithmus einen besseren Erwartungswert als $k + 1$ erreichen kann. Nach Yaos Prinzip lässt sich so nachweisen, dass auch keine randomisierte Strategie einen besseren Erwartungswert für den kompetitiven Faktor als $k + 1$ erreichen kann.

Kapitel 6

Ergebnisse und Ausblick

Zum Abschluss unserer Diplomarbeit möchten wir unsere Ergebnisse noch einmal zusammenfassen und bewerten sowie Themenbereiche herausstellen, über die es sich aus unserer Sicht lohnt weiter nachzudenken.

6.1 Zusammenfassung der Ergebnisse

Zunächst möchten wir einen kurzen Überblick über die Kernergebnisse unserer Diplomarbeit geben. In Kapitel 2 zeigen wir, wie Papadimitriou und Yannakakis in ihrer Arbeit [PY89] beweisen, dass die Entwicklung einer bezüglich des kompetitiven Faktors optimalen Strategie für das Canadian Traveller Problem (CTP) *PSPACE*-vollständig ist. In Kapitel 3 untersuchen wir die beiden konkreten Strategien *Backtrack* und *Greedy* auf verschiedenen Graphklassen hinsichtlich ihrer Kompetitivität beim Lösen des k -CTP, in dem im Gegensatz zum CTP a priori bekannt ist, dass maximal k Kanten ausfallen können. Die gleichen Graphklassen betrachten wir in Kapitel 4 in einem allgemeineren Zusammenhang. Dort geht es nicht um die Analyse von *Backtrack* und *Greedy*, sondern darum, eine untere Schranke für den kompetitiven Faktor einer beliebigen deterministischen Strategie auf der betreffenden Graphklasse zu finden. Im Kontrast zum übrigen Teil unserer Diplomarbeit, in dem wir ausschließlich deterministische Strategien untersuchen, geht es in Kapitel 5 um eine untere Schranke für den kompetitiven Faktor einer beliebigen randomisierten Strategie auf allgemeinen Graphen. Im Gegensatz zum deterministischen Fall existieren zu randomisierten Strategien verschiedene Arten von Gegenspielern. S. Westphal hat in seiner Arbeit [SW07] eine untere Schranke randomisierter Strategien im Zusammenhang mit dem vergesslichen Gegner (oblivious adversary) untersucht. Sein Ergebnis stellen wir in Kapitel 5 vor.

Tabelle 6.1 fasst noch einmal die Ergebnisse der Schwerpunkte unserer Arbeit zusammen. Die Abkürzungen L-H-R und T-B-R stehen dabei für Linke-Hand-Regel bzw. allgemein für tie-breaking-rule. Sind keine Einschränkungen angegeben, gilt für die verwendeten Variablen $k \in \mathbb{N}$, $t \in \mathbb{R}_{\geq 1}$ und $l \in \{1, \dots, |E|\}$. Aus Gründen der Übersichtlichkeit haben wir auf

die explizite Angabe von Ergebnissen verzichtet, wenn diese bereits unmittelbar aus den übrigen Ergebnissen folgen. Ist ein Wert mit „ $\geq$ “ versehen, handelt es sich um eine untere Schranke des kompetitiven Faktors, bei „ $=$ “ ist der kompetitive Faktor hingegen genau der angegebene Wert, d.h. obere und untere Schranke sind in dem Fall gleich.

6.2 Bewertung der Ergebnisse

Bei erstmaliger Betrachtung der beiden Strategien *Backtrack* und *Greedy* könnte man zunächst annehmen, dass die *Greedy*-Strategie schneller vom Start ins Ziel findet, das k -CTP also mit geringeren Kosten löst als die *Backtrack*-Strategie. Oftmals ist das möglicherweise auch so, denn *Greedy* nutzt beim Antreffen einer blockierten Kante stets die Tatsache aus, sich dem Ziel bereits ein Stück genähert zu haben, indem unmittelbar vom aktuellen Standort ein neuer kürzester Weg im aktualisierten Graphen zum Ziel berechnet wird. *Backtrack* hingegen verhält sich in dieser Situation defensiver und kehrt zunächst zum Startknoten zurück. Das hat zur Folge, dass ein Gegenspieler die *Greedy*-Strategie deutlich leichter reinlegen kann, wodurch der kompetitive Faktor auf allgemeinen Graphen ungleich höher ist als der von *Backtrack*, nämlich $2^{k+1} - 1$ statt $2k + 1$ (vgl. Tabelle 6.1).

Bei Betrachtung der Kompetitivität beider Strategien auf speziellen Graphklassen, fällt zunächst einmal auf, dass die untere Schranke der *Backtrack*-Strategie auf allgemeinen Graphen in den meisten Fällen Gültigkeit behält. Dabei weicht der Zusammenhang zwischen t -Spannern und allgemeinen Graphen etwas von dem zwischen Self-Spannern und allgemeinen Graphen ab. Bei t -Spannern ist zu Beginn ein beliebiger allgemeiner Graph gegeben. Die Restriktion bezieht sich dort auf die im Rahmen des k -CTP ausfallenden Kanten. Diese müssen so geartet sein, dass der Restgraph, der durch Entfernen dieser k Kanten entsteht, ein t -Spanner des Ursprungsgraphen ist. Bei den Self-Spannern muss bereits der Ursprungsgraph die Eigenschaft besitzen, auch nach Entfernen von maximal l $((t, l)$ -Self-Spanner) bzw. $|E|$ (t -Self-Spanner) beliebigen Kanten den Stretch-Faktor von t für alle Knotenpaare einzuhalten, die auch nach Entfernen dieser Kanten noch durch einen Pfad im Graphen verbunden sind. An den Restgraphen, der im Rahmen des k -CTP durch Entfernen von k Kanten aus dem Ursprungsgraphen entsteht, werden hier keine besonderen Anforderungen gestellt.

Für *Greedy* ergibt sich für einige spezielle Graphklassen ein anderes Bild. Betrachten wir beispielsweise die Klasse der vollständigen (m, n) -Gittergraphen. Zwar stellt der Wert $k - \frac{1}{2}$ nur eine untere Schranke des kompetitiven Faktors dar, für die obere Schranke ist jedoch anzunehmen, dass sie zumindest deutlich unter $2^{k+1} - 1$ liegt, vermutlich sogar kleiner ist als $2k + 1$, also dem Ergebnis für die *Backtrack*-Strategie. Die Vollständigkeit der Gitterstruktur stellt sicher, dass sich der *Greedy*-Strategie bei Antreffen einer blockierten Kante stets ein gewisses Repertoire an Ausweichmöglichkeiten darbietet. Die Möglichkeiten, die ein Gegenspieler mit einer ausfallenden Kante hat, sind daher deutlich geringer als auf allgemeinen Graphen. Weil die Gitterstruktur auf allgemeinen (m, n) -Gittergraphen nicht

	<i>Backtrack</i>	<i>Greedy</i>	bel. det. Strategie	bel. rand. Str.
allgemeiner Graph	$= 2k + 1$	$= 2^{k+1} - 1$	$\geq 2k + 1$	$\geq k + 1$
vollständiger Graph	$= 2k + 1$	$= 2^{k+1} - 1$	$\geq 2k + 1$	—
vollst. (m, n) -Gittergr.	$= 2k + 1$	$\geq k - \frac{1}{2}$, für $k \geq 3$ $\geq 2k$, mit L-H-R	$\geq \lfloor \sqrt{k} \rfloor + \frac{1}{2} + \frac{1}{2\lfloor \sqrt{k} \rfloor}$	—
allg. (m, n) -Gittergr.	$= 2k + 1$	$= 2^{k+1} - 1$	$\geq 2k + 1$	—
t -Spanner	$= 2k + 1$, für $t > 1$ $= 2k + 1$, mit T-B-R	$= 2\left(\frac{2t}{t+1}\right)^k - 1$	$\geq t - (t - 1)\left(\frac{t-1}{t+1}\right)^k$	—
(t, l) -Self-Spanner	$\geq \frac{2k+1+\varepsilon^2 \frac{1}{2k} + \varepsilon}{1+\varepsilon^2 \frac{1}{2k} + \varepsilon}$, für $t = 1 + \varepsilon$ und $\varepsilon > 0$	$= 2\left(\frac{2t}{t+1}\right)^k - 1$, für $l \geq k$ $\geq \max\left\{2\left(\frac{2t}{t+1}\right)^k - 1, 2^{\lfloor \frac{k}{t+1} \rfloor + 1} - 1\right\}$, für $l < k$	$\geq t - (t - 1)\left(\frac{t-1}{t+1}\right)^k$	—
t -Self-Spanner	$= 2k + 1$, für $t > 1$	$\geq (2k + 1)\frac{t-1}{t+1} + \frac{2}{t+1}$, für $t \geq 3$ $= k + 1$, für $t = 3$ $= 1$, für $t < 3$	$\geq (2k + 1)\frac{t-1}{t+1} + \frac{2}{t+1}$, für $t \geq 3$ $= 1$, für $t < 3$	—
ohne Mehrfachkanten	$= 2k + 1$, für $t \geq 2$			

Tabelle 6.1: Übersicht der Ergebnisse

notwendigerweise vollständig ist, geht *Greedy* der oben genannte Vorteil wieder verloren, was sich deutlich auf den kompetitiven Faktor der Strategie auf dieser Graphklasse auswirkt. Der kompetitive Faktor von *Greedy* auf t -Spannern ist $2\left(\frac{2t}{t+1}\right)^k - 1$, hängt also nicht nur von k , sondern auch vom Stretch-Faktor t ab. In Abhängigkeit von t bewegt sich der kompetitive Faktor also zwischen 1 (für $t = 1$) und $2^{k+1} - 1$ (für beliebig große t). Auf 1-Spannern ist die *Greedy*-Strategie also optimal - ein Ergebnis, das wir in Satz 3.18 auf Seite 70 auch direkt zeigen. Ist der Stretch-Faktor t beliebig groß, hat das zur Folge, dass beliebige k Kanten aus dem Ursprungsgraphen entfernt werden können. Da an den Restgraphen in dem Fall also keine besonderen Bedingungen gestellt werden, entspricht die vorliegende Problemstellung der von allgemeinen Graphen. Daraus ergibt sich bereits, dass der kompetitive Faktor für beliebig große t nur gegen $2^{k+1} - 1$ streben kann. Es existieren jedoch durchaus realistische Kombinationen von k und t , für die *Greedy* auf t -Spannern einen niedrigeren kompetitiven Faktor hat als die *Backtrack*-Strategie. Vergleiche dazu Abbildung 3.23 auf Seite 93. Das worst-case-Szenario, mit dem ein Gegenspieler die *Greedy*-Strategie auf t -Spannern zu dem oben genannten kompetitiven Faktor zwingen kann, lässt sich leicht modifiziert auch auf (t, l) -Self-Spanner übertragen. Daher sind die unteren Schranken beider Graphklassen identisch. Da (t, l) -Self-Spanner für $l \geq k$ ein Spezialfall der t -Spanner darstellen, sind für diese l -Werte zudem die oberen Schranken beider Graphklassen identisch. Für $l < k$ gilt, neben der von t -Spannern bekannten untere Schranke, zudem die untere Schranke $2^{\lfloor \frac{k}{t+1} \rfloor + 1} - 1$, die für kleine l -Werte eine Vergrößerung erst genannter Schranke darstellt. Auch auf t -Self-Spannern verkleinert sich der kompetitive Faktor der *Greedy*-Strategie gegenüber dem auf allgemeinen Graphen. Hier sind allerdings verschiedene t -Intervalle zu betrachten. Für alle $t < 3$ ist die *Greedy*-Strategie optimal. Für $t = 3$ ergibt sich mit $k + 1$ ein kompetitiver Faktor, der sehr deutlich unter dem *Greedy*-Ergebnis für allgemeine Graphen liegt und sogar das *Backtrack*-Ergebnis verbessert. Für $t \geq 3$ können wir mit $(2k + 1)\frac{t-1}{t+1} + \frac{2}{t+1}$ lediglich eine untere Schranke des kompetitiven Faktors angeben. Da dieser Wert für beliebig große t gegen $2k + 1$ strebt, wir auf Grund des *Greedy*-Ergebnisses für allgemeine Graphen aber wissen, dass sich für diesen t -Wert ein Szenario angeben lässt, in dem *Greedy* nicht besser als $(2^{k+1} - 1)$ -kompetitiv ist, lässt sich für $t \geq 3$ schwer einschätzen, wo die Wahrheit genau liegt.

Da keine deterministische Strategie auf allgemeinen Graphen besser als $(2k + 1)$ -kompetitiv ist, kennen wir mit *Backtrack* bereits eine worst-case-optimale Online-Strategie zur Lösung des k -Canadian Traveller Problem. Gleiches gilt für das k -CTP auf vollständigen Graphen und allgemeinen (m, n) -Gittergraphen. Auch auf diesen Graphklassen ist keine deterministische Strategie besser als $(2k + 1)$ -kompetitiv, *Backtrack* zugleich jedoch *genau* $(2k + 1)$ -kompetitiv. Auf vollständigen (m, n) -Gittergraphen ist der kompetitive Faktor jeder beliebigen deterministischen Strategie mindestens $\lfloor \sqrt{k} \rfloor + \frac{1}{2} + \frac{1}{2\lfloor \sqrt{k} \rfloor}$. Wie Abbildung 4.5 auf Seite 132 zeigt, liegen die Ergebnisse für *Backtrack* und *Greedy* auf dieser Graphklasse mit steigendem k zunehmend deutlich über diesem Wert. Die generelle untere Kompetitivitätsschranke des k -CTP auf t -Spannern beträgt $t - (t - 1)\left(\frac{t-1}{t+1}\right)^k$. Für $t = 1$ bzw. beliebig große t ergeben sich erwartungsgemäß die unteren Schranken 1

und $2k + 1$. Mit *Greedy* bzw. *Backtrack* kennen wir für diese beiden extremalen Stretch-Faktoren sogar zwei konkrete Strategien mit exakt diesen kompetitiven Faktoren für diese Graphklasse. Abbildung 4.8 auf Seite 142 zeigt den Zusammenhang der Ergebnisse für *Backtrack*, *Greedy* sowie der unteren Schranke einer beliebigen deterministischen Strategie für einige weniger extremale Stretch-Faktoren. Wie bereits bei der *Greedy*-Analyse lässt sich auch die untere Schranke für den kompetitiven Faktor einer beliebigen deterministischen Strategie von t -Spannern auf (t, l) -Self-Spanner übertragen. Beide Ergebnisse sind daher identisch, vergleiche Tabelle 6.1. Als Konsequenz daraus, dass die *Greedy*-Strategie auf t -Self-Spannern, mit $t < 3$, das k -CTP optimal löst, ist die generelle untere Schranke für den kompetitiven Faktor auf dieser Graphklasse ebenfalls 1. Für $t \geq 3$ erhöht sich die Schranke auf $(2k + 1) \frac{t-1}{t+1} + \frac{2}{t+1}$, ein Ergebnis, das wir bereits von der *Greedy*-Strategie für diese Graphklasse kennen. Da es sich bei unserem *Greedy*-Ergebnis jedoch ebenfalls nur um eine untere Schranke handelt, lässt sich daraus nicht folgern, dass mit *Greedy* eine worst-case-optimale Online-Strategie zur Lösung des k -CTP bekannt ist. Wir wissen aus den Ergebnissen für *Greedy* auf allgemeinen Graphen sogar, dass sich für beliebig große t ein t -Self-Spanner angeben lässt, auf dem *Greedy* nicht besser als $(2^{k+1} - 1)$ -kompetitiv ist. Der Grenzwert $t \rightarrow \infty$ für oben genannte untere Schranke ist jedoch $2k + 1$, was sich auch bereits aus dem *Backtrack*-Ergebnis für t -Self-Spanner folgern lässt.

Eine untere Schranke für den Erwartungswert des kompetitiven Faktors einer beliebigen randomisierten Strategie auf allgemeinen Graphen lässt sich, wie in Kapitel 5 beschrieben, mit $k + 1$ angeben. Dieses Resultat kann allerdings nicht direkt mit den Ergebnissen für *Backtrack*, *Greedy* bzw. einer beliebigen deterministischen Strategie auf dieser Graphklasse verglichen werden. Ein wesentlicher Unterschied besteht in den jeweils betrachteten Gegenspieler. Im deterministischen Fall haben wir stets einen mit dem adaptiven Offline-Gegner für randomisierte Strategien vergleichbaren Gegenspieler betrachtet, vergleiche Definition 5.1. Diesen charakterisiert die Eigenschaft, auf jeden Elementarschritt einer Strategie reagieren zu können. Nach Korollar 5.3 kann für diesen Gegner die untere Schranke des kompetitiven Faktors einer beliebigen deterministischen Strategie $(2k + 1)$ auch durch keine randomisierte Strategie herabgesetzt werden. Anders sieht es bei dem von Westphal im Zusammenhang mit Randomisierung betrachteten vergesslichen Gegner (vgl. Definition 5.4) aus, für den die oben genannte untere Schranke von $k + 1$ gilt. Der vergessliche Gegner legt im Gegensatz zum adaptiven Offline-Gegner alle ausfallenden Kanten a priori fest, kann also nicht unmittelbar auf die Elementarschritte einer Strategie reagieren. Das stellt für den Gegner natürlich eine signifikante Einschränkung dar. Zu beachten ist auch, dass es sich bei der unteren Schranke von $k + 1$ im Gegensatz zu den deterministischen Ergebnissen lediglich um einen Erwartungswert für den kompetitiven Faktor handelt.

6.3 Ausblick

In diesem letzten Abschnitt möchten wir noch auf einige Themenbereiche hinweisen, über die es sich aus unserer Sicht weiter nachzudenken lohnt.

Bei Betrachtung der Tabelle 6.1 fällt auf, dass die angegebenen unteren Schranken für die kompetitiven Faktoren der *Greedy*-Strategie bei einigen Graphklassen von den bekannten oberen Schranken abweichen. Interessant wäre es hier sicherlich zu untersuchen, wo der kompetitive Faktor genau liegt.

Bei dem Versuch zu zeigen, dass *Greedy* auf vollständigen (m, n) -Gittergraphen unter Verwendung der Linke-Hand-Regel (nicht schlechter als) $2k$ -kompetitiv ist, haben wir verschiedene Ansätze verfolgt. Auch wenn wir vermuten, dass $2k$ oder zumindest aber ein anderer linearer Wert tatsächlich eine obere Schranke darstellt, führte kein Ansatz zum Erfolg. Nach Satz 3.10 auf Seite 54 lässt sich nicht ausschließen, dass *Greedy* eine Kante mehr als zweimal besucht, was uns den Nachweis der oberen Schranke erschwerte. Für die *Greedy*-Strategie auf vollständigen (m, n) -Gittergraphen wäre es gerade deshalb interessant, den genauen kompetitiven Faktor zu kennen, weil dieser gegenüber allgemeinen Graphen deutlich geringer zu sein scheint, vermutlich sogar unter dem der *Backtrack*-Strategie liegt. Das gilt insbesondere auch für *Greedy* ohne Verwendung der Linke-Hand-Regel, weil dort die von uns angegebene untere Schranke mit $k - \frac{1}{2}$ noch einmal niedriger ist.

Für *Greedy* auf (t, l) -Self-Spannern wäre es interessant zu wissen, ob sich auch für $l < k$ eine allgemeine obere Schranke in Abhängigkeit von t und l angeben lässt. Für diese l -Werte lassen sich (t, l) -Self-Spanner nicht als Spezialfall der t -Spanner betrachten, weswegen die obere Schranke von t -Spannern hier nicht gilt.

Ähnlich verhält es sich mit *Greedy* auf t -Self-Spannern. Für $t \geq 3$ ist uns keine kleinere obere Schranke als die für *Greedy* auf allgemeinen Graphen $(2^{k+1} - 1)$ bekannt. Darüber hinaus wissen wir aus den Ergebnissen für *Greedy* auf allgemeinen Graphen, dass sich für beliebig große t ein t -Self-Spanner angeben lässt, auf dem *Greedy* nicht besser als $(2^{k+1} - 1)$ -kompetitiv ist, denn ein beliebig großer Stretch-Faktor ist gleichbedeutend damit, dass keine besonderen Anforderungen an den Graphen gestellt werden. Der Grenzwert $t \rightarrow \infty$ für die von uns angegebene untere Schranke strebt jedoch gegen $2k + 1$. Ob es möglich ist, ein allgemeingültiges Szenario anzugeben, für das die untere Schranke der Kompetitivität von *Greedy* auf t -Self-Spannern höher ist als $(2k + 1)\frac{t-1}{t+1} + \frac{2}{t+1}$, ist uns nicht bekannt. Auch in diese Richtung erscheint es uns lohnenswert, weitere Überlegungen anzustellen.

Tabelle 6.1 zeigt in der Spalte zu beliebigen randomisierten Strategien noch einige freie Felder. Das liegt daran, dass wir in unserer Arbeit den Fokus auf deterministische Strategien gelegt haben und uns mit Randomisierung nur als Randthema beschäftigen konnten. Trotzdem wäre es interessant, auch für spezielle Graphklassen untere Schranken für beliebige k -CTP-Strategien zu untersuchen. Wie in Kapitel 5 bereits erwähnt, gibt es im Zusammenhang mit Randomisierung eine Reihe unterschiedlicher Gegenspieler. Möglicherweise lassen sich auch in diese Richtung weitere interessante Untersuchungen anstellen.

Aufteilung der Diplomarbeit

- 1 Einleitung: –
 - 1.1 Motivation und Problemstellung: *M. Wedemeier*
 - 1.2 Inhalt der Arbeit: *M. Wedemeier, S. Jacobi*
 - 1.3 Variationen des Canadian Traveller Problem (CTP): *S. Jacobi*
 - 1.4 Grundlagen: *M. Wedemeier, S. Jacobi*
- 2 Komplexitätsanalyse des Canadian Traveller Problem (CTP): *M. Wedemeier*
 - 2.1 Reduktion von DVG auf QSAT: *M. Wedemeier*
 - 2.2 Transformation des DVG zum CTP: *M. Wedemeier*
- 3 Die Strategien *Backtrack* und *Greedy*: –
 - 3.1 Darstellung der Strategien: *S. Jacobi*
 - 3.1.1 Die Strategie *Backtrack*: *M. Wedemeier*
 - 3.1.2 Die Strategie *Greedy*: *M. Wedemeier*
 - 3.2 *Backtrack* und *Greedy* auf allgemeinen Graphen: *M. Wedemeier, S. Jacobi*
 - 3.2.1 Die Strategie *Backtrack*: *M. Wedemeier, S. Jacobi*
 - 3.2.2 Die Strategie *Greedy*: *M. Wedemeier, S. Jacobi*
 - 3.3 *Backtrack* und *Greedy* auf speziellen Graphklassen: *S. Jacobi*
 - 3.3.1 Vollständiger Graph: *M. Wedemeier*
 - 3.3.2 Vollständiger (m, n) -Gittergraph: *S. Jacobi*
 - 3.3.3 Allgemeiner (m, n) -Gittergraph: *S. Jacobi*
 - 3.4 *Backtrack* und *Greedy* auf Spannern: –
 - 3.4.1 t -Spanner: *S. Jacobi*
 - 3.4.2 (t, l) -Self-Spanner: *M. Wedemeier*
 - 3.4.3 t -Self-Spanner: *M. Wedemeier*
- 4 Untere Schranken deterministischer Strategien: *M. Wedemeier, S. Jacobi*
 - 4.1 Allgemeiner Graph: *M. Wedemeier*
 - 4.2 Vollständiger Graph: *S. Jacobi*
 - 4.3 Vollständiger (m, n) -Gittergraph: *S. Jacobi*
 - 4.4 Allgemeiner (m, n) -Gittergraph: *S. Jacobi*
 - 4.5 t -Spanner: *S. Jacobi*
 - 4.6 (t, l) -Self-Spanner: *M. Wedemeier*
 - 4.7 t -Self-Spanner: *M. Wedemeier*
- 5 Untere Schranke randomisierter Strategien: *M. Wedemeier*
- 6 Ergebnisse und Ausblick: *S. Jacobi*
 - 6.1 Zusammenfassung der Ergebnisse: *S. Jacobi*
 - 6.2 Bewertung der Ergebnisse: *S. Jacobi*
 - 6.3 Ausblick: *S. Jacobi*

Literaturverzeichnis

- [AY77] A. C. C. Yao, „Probabilistic computations: Towards a unified measure of complexity“, Proceedings of the 18th IEEE Symposium on Foundations of Computer Science, Seiten 222-227, 1977.
- [BD94] S. Ben-David, A. Borodin, R. M. Karp, G. Tardos und A. Wigderson, „On the power of randomization in on-line algorithms“, *Algorithmica*, 11:2-14, 1994.
- [BS91] A. Bar-Noy und B. Schieber, „The Canadian Traveller Problem“, Proceedings of the second annual ACM-SIAM symposium on discrete algorithms., Kap. 29, Seiten 261-270. San-Francisco, 1991.
- [BY98] Allan Borodin und Ran El-Yaniv, „Online Computation and Competitive Analysis“, Cambridge University Press, Seiten 98-108, 1998.
- [CL90] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest und Clifford Stein, „Introduction to Algorithms“, MIT Press, Cambridge, MA, Kapitel 24.3, 2001.
- [CP83] C.H. Papadimitriou, „Games against nature“, 1983.
- [DH99] Dagmar Handke, „Graphs with Distance Guarantees“, Dissertation der Fakultät für Mathematik und Informatik, Universität Konstanz, Dezember 1999.
- [ED59] E. W. Dijkstra, „A note on two problems in connection with graphs“, *Numerische Mathematik*, 1:269-271, 1959.
- [HL00] Katie Hafner und Matthew Lyon, „ARPA Kadabra oder die Geschichte des Internet.“, Dpunkt Verlag, 2. Auflage, Januar 2000.
- [IN01] Internet Timeline, <http://www.indexdirect.com/mag/timelinehistory.htm>, Oktober 2000.
- [IN02] Geschichte des Internets, <http://www.tako.de/Internetgeschichte.29.0.html>.
- [PY89] C.H. Papadimitriou und M. Yannakakis, „Shortest paths without a map“, Proc. 16th ICALP, Lecture Notes in Computer Science, Nr. 372, Seiten 610-620. Springer-Verlag, Juli 1989.

- [RK97] Rolf Klein, „Algorithmische Geometrie“, Addison-Wesley-Verlag, 1. Auflage, 1997.
- [RS06] R. Schrader „Das Canadian Traveller Problem“, Vorlesungsskript, Universität Köln, 2006.
- [SM73] Stockmeyer und Meyer, „Word problems requiring exponential time“, STOC, 1973.
- [SW07] Stephan Westphal, „A Note on the k -Canadian Traveller Problem“, Universität Kaiserslautern, März 2007.
- [TS78] T. Schaefer „Complexity of some two-player perfect information games“, Journal of computer and system sciences, 16:187-225, 1978.
- [WS74] Walter J. Savitch, „Relationships between Nondeterministic and Deterministic Tape Complexities“, The Journal of Symbolic Logic, Band 39, Nr. 2, Seiten 346-347, 1974.

Index

A

additive Konstante 37
 adversary
 adaptive offline adversary 147
 oblivious adversary 148
 allgemeiner (m,n)-Gittergraph 61
 allgemeiner Graph 40

B

Backtrack-Strategie 38

C

Canadian Traveller Problem (CTP) . 7 f., 19
 CTP-Entscheidungsproblem 8

D

deterministisches Recoverable k-Canadian
 Traveller Problem (dr-k-CTP) ... 13
 Double Valued Graph 21
 DVG-Entscheidungsproblem 21

E

Erfüllbarkeitsproblem für quantifizierte boole-
 sche Formeln 19
 Ergebnisübersicht 151

G

Gegner
 adaptiver Offline-Gegner 147
 vergesslicher Gegner 148
 Gewichtsfunktion 17
 Größe eines Kreises 111
 Graph

allgemeiner (m,n)-Gittergraph 61
 allgemeiner Graph 40
 ungerichteter Graph 40
 vollständiger (m,n)-Gittergraph 47
 vollständiger Graph 46
 zyklenfreie Graphen 46
 Graph-Strategie 8
 Greedy-Strategie 39
 Gruppe von G 118

K

k-Canadian Traveller Problem (k-CTP) .. 9,
 15
 k-Vital Edges Problem (k-VEP) 16
 kürzester Pfad 17
 kompetitiver Faktor 11, 37

L

Linke-Hand-Regel 51

M

minimale Pfadlänge 17

O

Online-Algorithmus 11
 Online-Problem 11
 Ordnungszahl 95

P

Problem
 Canadian Traveller Problem (CTP) 7 f.,
 19
 CTP-Entscheidungsproblem 8

- deterministisches Recoverable k-Canadian Traveller Problem (dr-k-CTP) 13
- DVG 21
- DVG-Entscheidungsproblem 21
- k-Canadian Traveller Problem (k-CTP) 9, 15
- k-Vital Edges Problem (k-VEP) 16
- Online-Problem 11
- PSPACE 19
- QSAT 19
- QSAT-Spiel 20
- Recoverable-CTP 11
- stochastisches Recoverable Canadian Traveller Problem (sr-CTP) 12
- PSPACE 19
- Q**
- QSAT 19
- QSAT-Spiel 20
- Quantified Boolean Formula (QBF) 19
- quasi-zyklenfrei 106
- R**
- Recoverable-CTP 11
- S**
- Spanner
- (t,l)-Self-Spanner 94
- t-Spanner 65
- stochastisches Recoverable Canadian Traveller Problem (sr-CTP) 12
- Strategie 9
- Backtrack-Strategie 38
- eindeutig 17
- Greedy-Strategie 39
- nicht eindeutig 10
- Stretch-Faktor 65
- T**
- (t,l)-Self-Spanner 94
- t-Spanner 65
- tie-breaking-rule 97
- U**
- ungerichteter Graph 40
- V**
- vollständiger (m,n)-Gittergraph 47
- vollständiger Graph 46
- Z**
- zyklenfreie Graphen 46