

RWTH Aachen University  
Lehrstuhl für Informatik 1  
Dr. W. Unger

Seminar Kommunikationsprobleme im SS 2008

# **Broadcasting in Kantengraphen**

Markus Arndt (265555)

15. September 2008

## Zusammenfassung

Diese Ausarbeitung beschäftigt sich mit zwei Broadcast-Algorithmen auf gerichteten, regulären Kantengraphen, welche konstruktiv aus Broadcast-Algorithmen derer Ursprungsgraphen hervorgehen.

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Motivation . . . . .	3
1.2	Übersicht . . . . .	3
<b>2</b>	<b>Definitionen</b>	<b>4</b>
2.1	Gerichteter Kantengraph . . . . .	4
2.2	D-Regularität . . . . .	4
2.3	Starker Graphenzusammenhang . . . . .	5
2.4	D-Labeling . . . . .	5
2.5	Labeled Broadcasting Tree . . . . .	6
<b>3</b>	<b>Erster Algorithmus auf Kantengraphen</b>	<b>7</b>
3.1	Idee der Konstruktion . . . . .	7
3.1.1	Wiederverwendung für partiellen Broadcast . . . . .	7
3.1.2	Erweiterung auf vollständigen Broadcast . . . . .	8
3.2	Theorem zur Laufzeit . . . . .	8
3.2.1	Beweis . . . . .	8
3.3	Fazit . . . . .	9
<b>4</b>	<b>Verbesserter Algorithmus auf Kantengraphen</b>	<b>10</b>
4.1	Probleme des ersten Algorithmus . . . . .	10
4.2	Modifikation bzgl. untätiger Knoten . . . . .	10
4.3	Modifikation bzgl. Redundanz . . . . .	11
4.4	Theorem zur Laufzeit . . . . .	11
4.4.1	Beweis . . . . .	12
<b>5</b>	<b>Zusammenfassung</b>	<b>15</b>
<b>6</b>	<b>Literaturverzeichnis</b>	<b>16</b>

# 1 Einleitung

## 1.1 Motivation

Da Kantengraphen von ihrer Struktur her eng mit ihren Ursprungsgraphen zusammenhängen liegt die Annahme nahe es sei möglich bereits bekannte Verfahren bzw. Algorithmen für Graphen auf deren Kantengraphen anzuwenden. Weiterhin werden wir uns in dieser Ausarbeitung konkret mit Broadcast-Algorithmen auseinandersetzen und beliebige Broadcasts auf Kantengraphen erweitern.

## 1.2 Übersicht

Zunächst definieren wir einige grundlegende Konstruktionen die im weiteren Verlauf benötigt werden. Danach wird die Idee für den ersten Broadcast-Algorithmus auf Kantengraphen erläutert, in der ein partielles Broadcast im Kantengraphen in Anlehnung an einen bereits existierenden Broadcast-Algorithmus bzgl. des Ursprungsgraphen durchführt und anschließend in logarithmisch vielen Runden komplettiert wird. Es folgt ein Beweis der Laufzeit.

Alsdann identifizieren wir zwei Schwachstellen in der ersten Konstruktion und führen zwei Modifikationen ein um diese aufzuheben. So gelingt eine zweckmäßige Verbesserung der Laufzeit, welche im Anschluß bewiesen wird.

## 2 Definitionen

Zunächst werden einige für die Ausarbeitung benötigte Konstrukte und Konzepte definiert und erläutert.

### 2.1 Gerichteter Kantengraph

Sei  $G = (V, E)$  ein gerichteter Graph, dann ist  $L(G) = (E, E')$  der dazugehörige **gerichtete Kantengraph**, mit

$$E' := \{(v, w) \in E \times E \mid \exists a, b, c \in V : v = (a, b) \text{ und } w = (b, c)\}$$

Eine anschauliche Interpretation ist, dass die Kanten von  $G$  (Abbildung 1) zu Knoten in  $L(G)$  (Abbildung 2) werden und eine  $L(G)$ -Kante  $(v, w)$  genau dann existiert wenn  $v$  eine Eingangskante eines Knotens in  $G$  war zu dem  $w$  die Ausgangskante war.

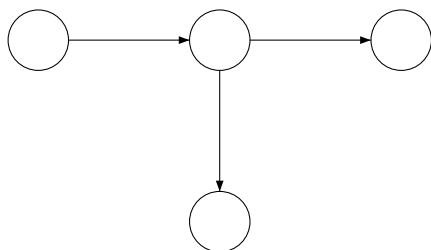


Abbildung 1:  $G = (V, E)$

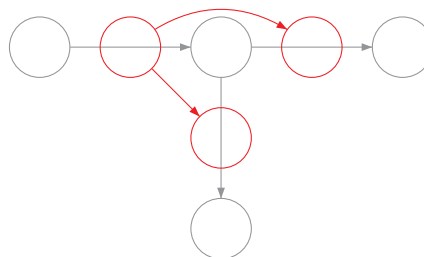


Abbildung 2:  $L(G) = (E, E')$

### 2.2 D-Regularität

Ein gerichteter Graph (Digraph)  $G = (V, E)$  nennt sich  $d$ -regulär

$$\Leftrightarrow \forall v \in V : d = |\{(x, v) \in E\}| = |\{(v, y) \in E\}|$$

Der Eingangsknotengrad, sowie der Ausgangsknotengrad aller Knoten  $x \in E$  muss  $d$  entsprechen.

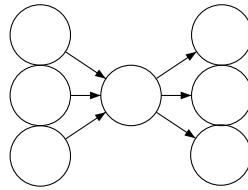


Abbildung: Knoten eines 3-regulären Graphen.

### 2.3 Starker Graphenzusammenhang

Ein Graph ist **stark zusammenhängend**, wenn von jedem Knoten aus jeder andere Knoten aus erreichbar ist.

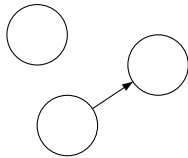


Abbildung 3: Unzusammenhängender Graph.

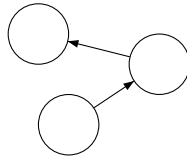


Abbildung 4: Zusammenhängender Graph.

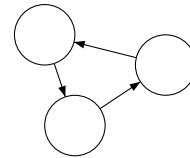


Abbildung 5: Stark zusammenhängender Graph.

### 2.4 D-Labeling

Ein **d-Labeling** ist eine Abbildung  $\pi : E \rightarrow \mathbb{Z}/d\mathbb{Z}$  über einem d-regulären Kantengraphen, so dass gilt:

$$\forall v \in V, \forall (x, v) \in E : \pi((x, v)) \neq \pi(y, v) \text{ falls } y \neq x$$

$$\forall v \in V, \forall (v, x) \in E : \pi((v, x)) \neq \pi(v, y) \text{ falls } y \neq x$$

Man kann sagen es wird jeder Ein- und Ausgangskante eine aus  $\mathbb{Z}/d\mathbb{Z}$  zugeordnet (Repräsentanten sind z.B. 0, 1, ..., d-1). Dabei dürfen an einem Knoten nie zwei unterschiedliche Eingangskanten auf das gleiche Element abgebildet werden, das gleiche gilt für die Ausgangskanten.

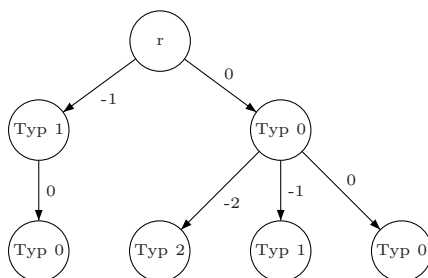
## 2.5 Labeled Broadcasting Tree

Ein **Labeled Broadcasting Tree** (später  $T(r)$  genannt) ist ein (gerichteter) Teilbaum eines gerichteten Graphen  $G$  bzgl. eines Broadcasts. Die Baumwurzel, genannt  $r$ , ist der Knoten der den Broadcast in  $G$  startet, die Kanten in  $T(r)$  sind die im Broadcast benutzten Kanten.

Die Ausgangskanten an jedem Knoten werden nun mit den größten nicht-positiven Repräsentanten (also auch 0) der Äquivalenzklassen aus  $\mathbb{Z}/d\mathbb{Z}$  gelabeld, und zwar derart, dass die vom Knoten zuletzt benachrichtigte Ausgangskante mit 0 gelabeld wird, die davor benachrichtigte mit  $-1$ , die davor benachrichtigte mit  $-2$ , und so weiter.

Man beachte dass eine  $-k$ -Kante genau  $k$  Runden vor der zuletzt benachrichtigten Kante informiert wird.

Man nennt einen Knoten vom Typ  $k$ , wenn er über eine  $-k$ -Kante informiert wurde.



Ein Graph nennt sich mit einem Labeled Broadcasting Tree **konsistent**, wenn

- der Baum ein Teilgraph ist und
- die Kanten von  $G$  so gelabeld werden können, dass das Labeling des  $T(r)$  erhalten bleibt.

### 3 Erster Algorithmus auf Kantengraphen

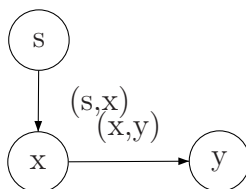
Wir betrachten ab nun nur noch stark zusammenhängende Kantengraphen, die  $d$ -regulär sind. Man beachte dass ein  $D$ -Labeling so immer möglich ist. Anzumerken ist noch dass Multikanten für die folgenden Betrachtungen ebenfalls legitim sind.

#### 3.1 Idee der Konstruktion

Es wird nun gezeigt wie die Konstruktion des Broadcast-Algorithmus realisiert wird. Dazu bedienen wir uns zweier Ideen, der Wiederverwendung eines Algorithmus welcher auf den Ursprungsgraphen zugeschnitten ist und einer Erweiterung die der Terminierung des Ursprungsalgorithmus folgt.

##### 3.1.1 Wiederverwendung für partiellen Broadcast

Informiert ein Knoten  $x$  einen anderen Knoten  $y$ , so sagen wir dass auch die Kante  $(x, y)$  informiert wurde. Es ist leicht einzusehen dass bei jedem Informieren eines Knotens  $y$  die entsprechende Eingangskante  $(\cdot, y)$  informiert wird.



Betrachten wir also einen fertiges Broadcast auf  $G$ , so können wir sagen dass zu jedem Knoten außer der Broadcast-Quelle eine informierte Eingangskante existiert. Da die Kanten in  $G$  gleichzeitig Knoten in  $LG$  entsprechen induziert ein  $G$ -Broadcast mit Quelle  $x$  einen partiellen (unvollständigen)  $LG$ -Broadcast mit Quelle  $(s, x)$ , für alle  $x$ -Eingangskanten  $s$ .

Ein  $G$ -Broadcast kann also einfach als partieller  $LG$ -Broadcast benutzt werden an dessen Ende genau die  $LG$ -Knoten informiert sind, die den benutzten bzw. informierten  $G$ -Kanten entsprechen.

Im Folgenden werden wir  $LG$ -Broadcasts einfach als Kantenbroadcasts in

$G$  betrachten, d.h. es sollen alle  $G$ -Kanten informiert werden. Wir sprechen dann davon dass  $G$ -Kanten miteinander kommunizieren wenn wir meinen dass  $LG$ -Knoten miteinander kommunizieren.

Wir halten fest dass wir nach  $t$  Runden (unter der Annahme der  $G$ -Broadcast lief in  $t$  Runden) zu jedem Knoten eine informierte Eingangskante haben.

### 3.1.2 Erweiterung auf vollständigen Broadcast

Liegt nach  $t$  Runden ein partieller Broadcast derart wie oben beschrieben vor, so wende man ein  $d$ -Labeling (Labels  $1, \dots, d$ ) auf den  $d$ -regulären Graphen an.

An jedem Knoten informiert in Runde  $t + 1$  die bereits informierte Eingangskante die Ausgangskanten mit Label 1, so dass wir nun mit Sicherheit sagen können dass alle Label-1-Kanten informiert sind (alle Ausgangskanten sind natürlich auch bei anderem Bezugsknoten Eingangskanten).

Ab jetzt ( $t + 2$ ) wird ein Schema verwendet um die Information in den Kanten exponentiell anwachsen zu lassen. Alle Label-1-Kanten informieren die Label-2-Kanten in Runde  $t + 2$ . In Runde  $t + 3$  informieren Label 1 und Label 2 die Label 3 und 4. In  $t + 4$  informieren Label 1,2,3 und 4 die Label 5,6,7 und 8. Dieses Schema setzt sich dann so fort bis alle Labels informiert wurden.

Zusammengefasst kann man sagen dass in Runde  $t + 1$  alle Label 1 Kanten informiert werden und dass in den Runden  $t + 1 + k$  die Label  $j$  an die Ausgangskanten mit Label  $2^{k-1} + j$  senden.

## 3.2 Theorem zur Laufzeit

Falls ein Broadcast-Algorithmus für den  $d$ -regulären Graphen  $G$  existiert, welcher in  $t$  Runden beendet ist, so existiert nach dem genannten Vorgehen ein Broadcast-Algorithmus, welcher in  $t + \lceil \log(d) \rceil + 1$  beendet ist.

### 3.2.1 Beweis

Unterteilen wir diese Behauptung nun in zwei Aussagen.

- Teil 1: Zur Runde  $t + 1$  sind alle Label 1 Kanten informiert.

- Teil 2: Nach  $k$  weiteren Runden sind alle Label  $i, i \leq 2^k$  informiert.

### **Beweis von Teil 1**

Zur Zeit  $t$  sind alle Knoten  $x$  in  $G$  informiert. Für jeden beliebigen Knoten  $x$  sei  $(i, x)$  die Kante durch welche  $x$  informiert wurde. In Runde  $t$  hat  $(i, x)$  die Information. In Runde  $t + 1$  sendet  $(i, x)$  die Information in  $LG$  an die Nachfolgekante von  $x$  mit Label 1.

Da  $x$  beliebig war sind alle Kanten mit Label 1 zur Zeit  $t + 1$  informiert.

### **Beweis von Teil 2**

Der Beweis wird durch Induktion über die verstrichenen Runden  $k$  geführt.

#### **Induktionsanfang $k = 1$**

In der ersten Runde senden alle Kanten mit Label 1 die Information an Kanten mit Label 2. Somit sind alle Kanten mit Label  $i, i \leq 2^k = 2$ , informiert.

#### **Induktionsannahme**

Die Behauptung gelte für ein festes  $k \in \mathbb{N}$ .

#### **Induktionsschritt $k \Rightarrow k + 1$**

Nach Induktionsvoraussetzung sind alle Kanten mit Label  $i, i \leq 2^k$ , informiert. Nun senden alle Kanten mit Label  $j$  an die adjazente Kanten mit Label  $j + 2^k$ . Da alle Labelklassen gleich groß sind informiert also in jedem Schritt eine Labelklasse eine komplette andere Klasse.

Daraus folgt dass nun die Klassen  $2^k + 1$  bis  $2^k + 2^k = 2^{k+1}$  zusätzlich benachrichtigt, und insgesamt alle Kanten mit Label  $i, i \leq 2^{k+1}$  informiert wurden.

## **3.3 Fazit**

Durch die Beweisführung wurde gezeigt dass ein LG-Broadcastalgorithmus in  $t + \lceil \log(d) \rceil + 1$  konstruierbar ist, auf Basis eines G-Broadcastalgorithmus welcher in  $t$  Runden läuft.

## 4 Verbesserter Algorithmus auf Kantengraphen

Im weiteren Verlauf werden nur noch Graphen betrachtet für die es zu jedem Knoten ein Broadcast gibt, so dass der Graph zum dazugehörigen Labeled-Broadcasting-Tree konsistent ist. Dies ist wichtig da wir das Labeling nun nicht mehr wahllos durchführen sondern aus dem LBT übernehmen.

Anzumerken ist dass nun nicht-positive Repräsentanten als Labels verwendet werden (d.h.  $-d + 1, -d + 2, \dots, -1, 0$ ), da wir die Labels nach der LBT-Konstruktion wählen.

### 4.1 Probleme des ersten Algorithmus

Um den gesehenen Algorithmus zu verbessern identifizieren wir zwei Probleme.

**Untätige Knoten:** Die Information kann Kanten bereits vor  $t$  Runden erreichen, die dann bis zur Runde  $t$  nichts tun. Effizienter ist es diese Zeit auszunutzen und bis zur Runde  $t$  weitere Kanten zu informieren.

**Redundanz:** Vor  $t$  Runden werden bereits gelabelte G-Kanten benutzt. Danach werden nochmal **alle** Label informiert, also ebenfalls bereits informierte.

Nun werden zwei Modifikationen eingeführt um diese beiden Schwachstellen auszubessern.

### 4.2 Modifikation bzgl. untätiger Knoten

Betrachten wir einen beliebigen Knoten vom Typ  $k$  in der Runde in der er und die entsprechende Eingangskante informiert werden. Dies geschieht noch zur Zeit des G-Broadcasts, also vor der Runde  $t$ .

Nach Konstruktion des LBT hat dieser Knoten nun mindestens noch  $k$  Runden Zeit bis zum Beenden der Runde  $t$ , es könnten also  $k$  Nachfolger informiert werden. Angenommen es werden  $k'$  ( $k' \leq k$ ) Knoten informiert, so sind also die Label  $-k' + 1, -k' + 2, \dots, 0$  informiert (der LBT gibt vor dass die

Label derart chronologisch gewählt wurden). Zusätzlich werden nun in dieser Modifikation in den  $k - k'$  Runden danach alle weiteren Label absteigend informiert, wobei bei  $-k'$  begonnen wird.

So sind an einem Typ  $k$ -Knoten alle Ausgangskanten höheren Labels zur Runde  $t$  bereits informiert, dort waren ja mindestens  $k$  Runden Zeit bis zum Ende von  $t$ .

### 4.3 Modifikation bzgl. Redundanz

In Runde  $t + 1$  werden an allen Knoten die Ausgangskanten mit Label  $-d + 1$  informiert (entspricht erstem Algorithmus, da  $-d + 1 \equiv_d 1$ ).

Betrachten wir nun einen beliebigen Knoten  $x$  vom Typ  $k$ .

**Fall 1,**  $-k = -d + 1$

Der Knoten wurde durch eine Eingangskante vom niedrigstem Typ, d.h.  $-d + 1$  informiert. In der vorherigen Modifikation haben wir gesehen dass bereits alle Ausgangskanten höheren Labels zur Runde  $t$  informiert wurden. Da in  $t + 1$  nun auch  $-d + 1$  informiert wurde sind alle Ausgangskanten informiert und der Knoten wird nicht weiter betrachtet.

**Falls 2,**  $-k \neq -d + 1$

Da die informierte Eingangskante nicht vom Label  $-d + 1$  ist und in  $t + 1$  genau diese Label informiert wurden existieren an diesem Knoten nun bereits zwei benachrichtigte Eingangskanten, anstelle von einer wie im ersten Algorithmus. In  $t + 2$  kann die Information also bereits durch zwei Kanten weiterpropagiert werden.

Von nun an wird die Information wieder exponentiell weiterpropagiert wie im ersten Algorithmus.

### 4.4 Theorem zur Laufzeit

Falls ein Broadcast-Algorithmus für den  $d$ -regulären Graphen  $G$  existiert, welcher in  $t$  Runden beendet ist, so existiert nach dem verbesserten Vorgehen ein Broadcast-Algorithmus, welcher in  $t + \lfloor \log(d) \rfloor + 1$  beendet ist.

#### 4.4.1 Beweis

Dass an allen Knoten des Typ  $k$  alle Ausgangskanten mit größerem Label als  $-k$  informiert sind ist ein Resultat das wir in der ersten Modifikation gesehen haben. Auch haben wir gesehen dass jeder relevante Knoten nach Beenden der Runde  $t + 1$  zwei informierte Eingangskanten hat. Betrachten wir den verbesserten Algorithmus auf dem Graphen zum Zeitpunkt  $t$ .

Wir wollen nun zunächst das Folgende Lemma durch Induktion über die der Runde  $t$  folgenden Runden  $h$  beweisen.

**Induktionsbehauptung** Betrachten wir einen beliebigen Knoten  $x$  vom Typ  $k$  zum Zeitpunkt  $t + h$ .

- Alle Ausgangskanten sind informiert, falls  $-k \in \{1, \dots, 2^h - 1\}$  (**Aussage A**),
- oder alle Eingangskanten mit Label  $j \in \{1, \dots, 2^h - 1\}$  sind inf, falls  $-k \notin \{1, \dots, 2^h - 1\}$  (**Aussage B**).

#### Induktionsanfang

Für  $h = 1$  gilt  $2^h - 1 = 1 \equiv_d -d + 1$  und die Fallunterscheidung zerfällt zu ' $-k = -d + 1$ ' oder ' $-k = -d + 1$ '.

- **Annahme**  $-k = -d + 1$

Alle Nachfolger von höherem Label (Labels  $-d + 2, \dots, 0$ ) waren bereits informiert (Voraussetzung). Label  $-d + 1$  werden in  $t + 1$  informiert. Die Ausgangskanten der Knoten des Typ  $-d + 1$  sind nun also alle informiert (**Aussage A** gilt).

- **Annahme**  $-k \neq -d + 1$

Labels  $-d + 1 \equiv_d 1$  werden in  $t + 1$  informiert. Nun sind alle  $j$ -Eingangskanten ( $1 \leq j \leq 2^h - 1 = 1$ ) informiert (**Aussage B** gilt).

### Induktionsvoraussetzung

Die Induktionsbehauptung gelte für festes  $h \in \mathbb{N}$ .

### Induktionsschritt $h \Rightarrow h + 1$

Wir zerlegen den Induktionsschritt in zwei Fälle.

- **Fall 1:**  $-k \in \{1, \dots, 2^h - 1\}$

$$-k \in \{1, \dots, 2^h - 1\} \Rightarrow -k \in \{1, \dots, 2^{h+1} - 1\}$$

Da in in Runde  $h$  bereits alle Ausgangskanten informiert waren und bleiben diese in Runde  $h + 1$  informiert. **Fall A** trifft zu.

- **Fall 2:**  $-k \notin \{1, \dots, 2^h - 1\}$

Nach Induktionsvoraussetzung gilt dass alle Eingangskanten mit Labels  $\{1, \dots, 2^h - 1\}$  informiert sind. Zusätzlich ist noch eine weitere Eingangskante informiert, und zwar jene mit Label  $-k$ , da sich diese nach Fallunterscheidung nicht in  $\{1, \dots, 2^h - 1\}$  befindet ( $-k \notin \{1, \dots, 2^h - 1\}$ ).

Insgesamt stehen  $2^h$  informierte Eingangskanten zur Verfügung, mit denen nun die Label  $2^h$  bis  $2^{h+1} - 1$  in Runde  $t + h + 1$  informiert werden.

- Falls danach  $-k \in \{1, \dots, 2^{h+1} - 1\}$  gilt:

$1, \dots, 2^h - 1, 2^h, \dots, 2^{h+1} - 1$  wurden informiert. Außerdem sind bereits nach Voraussetzung alle Nachfolger von größerem Label als  $-k$  informiert. Da  $-k \leq 2^{h+1} - 1$  wurden somit alle Nachfolgekanten informiert. Also gilt **Aussage A**.

- Falls danach  $-k \notin \{1, \dots, 2^{h+1} - 1\}$  gilt:

Aussage A ist hier nicht mehr haltbar. Jedoch wurden in Fall 2 zur Runde  $t + h + 1$  die Labels  $2^h$  bis  $2^{h+1} - 1$  informiert, wobei jene die noch niedriger waren bereits nach Induktionsvoraussetzung informiert waren. Also gilt **Aussage B**.

Somit haben wir das Lemma bewiesen. Betrachten wir die **Aussage A** des Lemmas:

*Zur Runde  $t + h$  sind alle  $-k$  gelabelten Ausgangskanten informiert, falls  $-k \in \{1, \dots, 2^h - 1\}$  (**Aussage A**).*

Da des zuletzt erreichte Label 0 bzw.  $d$  ist gilt für die letzte Runde  $h$

$$\begin{aligned}d &= 2^h - 1 \\ \Rightarrow h &= \lceil \log_2(d + 1) \rceil\end{aligned}$$

Da  $\lceil \log_2(d + 1) \rceil = \lfloor \log_2(d) \rfloor + 1$  in  $\mathbb{N}$  ist die Laufzeit also  $t + \lfloor \log_2(d) \rfloor + 1$ .

Die Behauptung ist somit bewiesen.

## 5 Zusammenfassung

Es wurde gezeigt dass mit einer einfachen Erweiterung ein bestehender Broadcast-Algorithmus (Laufzeit  $t$ ) für Graphen ein Broadcast-Algorithmus für dessen Kantengraphen konstruierbar ist, welcher die Laufzeit  $t + \lceil \log_2(d) \rceil + 1$  besitzt.

Weiterhin wurde ein verbesserter Ansatz vorgestellt welcher die Laufzeit auf  $t + \lfloor \log_2(d) \rfloor + 1$  verbessert.

Eine rekursive Anwendung des Schemas erzeugt Broadcast-Algorithmen für Kantengraphen  $L^k(G)$  höherer Dimension. Ebenfalls sind Verbesserungen der Laufzeit für spezielle Grade der Ursprungsgraphen bekannt.

## 6 Literaturverzeichnis

**A Broadcasting protocol in Line Digraphs,**

Jean-Claude Bermond, Xavier Muñoz, Alberto Marchetti-Spaccamela

19th February 1999