

# Das Kreuzungsproblem in Zwei-Lagen Graphen

basierend auf

## Edge Crossings in Drawings of Bipartite Graphs

von

Peter Eades und Nicholas C. Wormald,  
Algorithmica 11, 1994, pp. 379-403

Seminar Zeichnen von Graphen

Sandra Mangano

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>2</b>
<b>2</b>	<b>Grundlagen</b>	<b>3</b>
<b>3</b>	<b>Das Kreuzungsproblem und die <math>\mathcal{NP}</math>-Vollständigkeit von <math>DCP</math></b>	<b>4</b>
3.1	Das Kreuzungsproblem (Crossing Problem, $CP$ ) . . . . .	4
3.2	Das Entscheidungskreuzungsproblem (Decision Crossing Problem, $DCP$ ) . . . . .	4
3.3	Das FAS-Problem ( $FASP$ ) . . . . .	4
<b>4</b>	<b>Heuristiken</b>	<b>9</b>
4.1	Die Schwerpunkt-Heuristik . . . . .	9
4.2	Die Median-Heuristik . . . . .	10
<b>5</b>	<b>Bemerkungen</b>	<b>15</b>

# 1 Einführung

Die Darstellungen von großen Systemen, komplizierten Verbindungen oder anderen Sachverhalten findet sich heutzutage in den verschiedensten Bereichen wieder, wie beispielsweise in der Chemie, der Biologie, der Informatik, aber auch häufig in der Wirtschaft sowie in Zeitungsartikeln. Um diese Sachverhalte leicht verständlich zu machen, werden sie oft anhand von Graphen dargestellt, die einen einfachen und schnell erfassbaren Zugang ermöglichen sollen. Dafür muß jedoch auch das Layout des Graphen so einfach wie möglich aussehen. Für die Vereinfachung eines Graphenlayouts benutzt man unter anderem folgende Kriterien:

- (i) Die Knoten im Graphen sollen gleichmäßig verteilt sein;
- (ii) Es sollen so wenig Kantenkreuzungen wie möglich entstehen.

Punkt (i) wird besonders bei gerichteten Graphen durch das Einführen von sogenannten Lagen realisiert, wobei die Knoten in mehreren Schichten übereinander platziert werden. Um zu vermeiden, dass Kanten über zwei oder mehr Lagen gezeichnet werden (dass also zwischen Anfangs- und Endknoten noch weitere Lagen existieren), werden sogenannte Dummy-Knoten geschaffen, die in den Zwischenlagen platziert werden. Die Kante wird nun als Pfad zwischen Anfangs- und Endknoten auf diesen Dummy-Knoten realisiert.

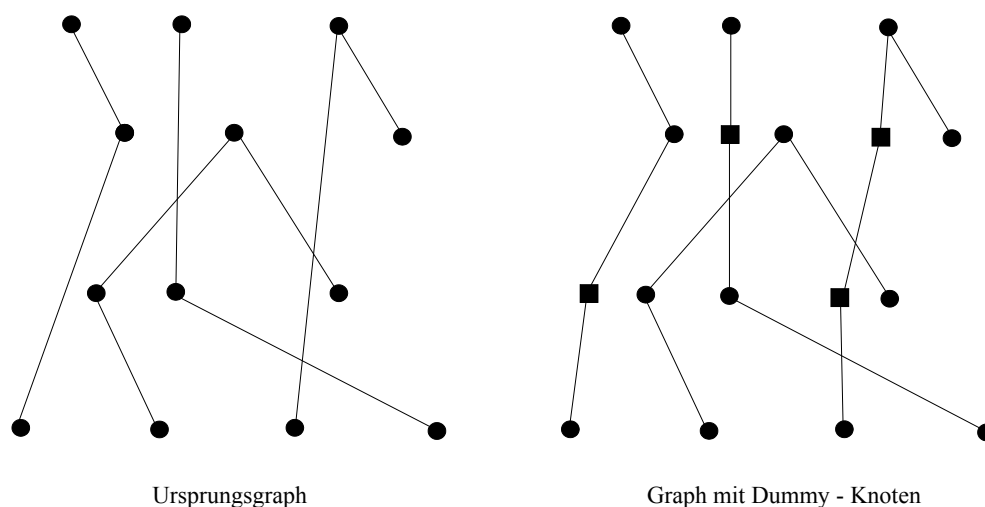


Abbildung 1: Vermeiden von langen Kanten durch Dummy-Knoten

Im folgenden werden wir uns mit der Realisierung von Punkt (ii) in einem solchen Lagengraphen beschäftigen, basierend auf [4]. Dabei können wir uns auf einen Graphen mit lediglich zwei Lagen beschränken, da nur zwischen benachbarten Lagen Kreuzungen entstehen können.

Um die Anzahl der Kantenkreuzungen zu minimieren, werden wir die Knoten in der unteren Lage fest lassen und eine Permutation der Knoten in der oberen Lage suchen, bei der am wenigsten Kreuzungen entstehen. Dieses sogenannte Kreuzungsproblem wird in Kapitel 3 formuliert werden; außerdem werden wir zeigen, dass es  $\mathcal{NP}$ -schwer ist. In Kapitel 4 werden wir zwei Heuristiken kennenlernen, die eine Permutation der oberen Lage festlegen. Zudem wird gezeigt werden, dass die Erste eine asymptotische und die Zweite sogar eine konstante Schranke für ihre Güte besitzen.

## 2 Grundlagen

Im folgenden betrachten wir stets einen ungerichteten Graphen  $G = (L_0, L_1, A)$ , dessen Knoten auf den beiden Mengen  $L_0$  und  $L_1$ , den sogenannten Lagen, aufgeteilt sind. Diese Lagen sollen jeweils horizontal angeordnet sein; die Lage  $L_0$  soll dabei unter der Lage  $L_1$  liegen. Die Menge  $A \subseteq L_0 \times L_1$  besteht aus den Kanten des Graphen, die nur zwischen Knoten aus verschiedenen Lagen verlaufen.

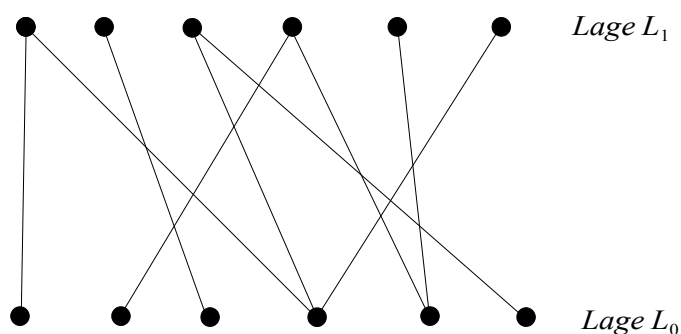


Abbildung 2: Beispiel für einen Zwei Lagen-Graphen

Mit  $x_0$  bzw.  $x_1$  bezeichnen wir die x-Koordinaten der Knoten in der Lage  $L_0$  bzw.  $L_1$ . Da wir uns allerdings nur für die Reihenfolge der Knoten und nicht ihre genaue Position interessieren, können wir  $x_0$  und  $x_1$  auch einfach als eine Anordnung der Knoten in den beiden Lagen  $L_0$  und  $L_1$  betrachten.

Weiterhin sei die Nachbarschaft eines Knoten  $u$  die Menge aller zu  $u$  inzidenten Knoten, also  $N_u := \{v : \{u, v\} \in A\}$ . Zudem sei  $d_u := |N_u|$  der Grad des Knoten  $u$ .

Die Anzahl der Kantenkreuzungen im Graphen  $G$  mit den Lagenanordnungen  $x_0$  und  $x_1$  nennen wir  $cross(G, x_0, x_1)$ . Desweiteren wird mit  $opt(G, x_0) := \min_{x_1} cross(G, x_0, x_1)$  die minimale Anzahl von Kreuzungen im Graphen  $G$  bei fester Anordnung  $x_0$  bezeichnet.

Mit Hilfe dieser Begriffe lässt sich nun das Kreuzungsproblem definieren, von dem wir noch zeigen werden, dass es  $\mathcal{NP}$ -schwer ist.

### 3 Das Kreuzungsproblem und die $\mathcal{NP}$ -Vollständigkeit von $DCP$

Unser Problem der Kreuzungsminimierung lässt sich wie folgt formulieren:

#### 3.1 Das Kreuzungsproblem (Crossing Problem, $CP$ )

Gegeben: Ein Graph  $G = (L_0, L_1, A)$  und eine Anordnung  $x_0$  von  $L_0$ .

Gesucht: Eine Anordnung  $x_1$  von  $L_1$ , so dass  $cross(G, x_0, x_1) = opt(G, x_0)$ .

Um zu zeigen, dass das Kreuzungsproblem  $\mathcal{NP}$ -schwer ist, werden wir es mit Hilfe einer oberen Schranke für die Anzahl der Kreuzungen als Entscheidungsproblem umformulieren. Ist dieses Entscheidungsproblem  $\mathcal{NP}$ -vollständig, so ist das zugehörige Optimierungsproblem  $CP$  auch  $\mathcal{NP}$ -schwer.

#### 3.2 Das Entscheidungskreuzungsproblem (Decision Crossing Problem, $DCP$ )

Gegeben: Ein Graph  $G = (L_0, L_1, A)$  und eine Anordnung  $x_0$  von  $L_0$  und eine Schranke  $M \in \mathbb{N}$ .

Frage: Gibt es eine Anordnung  $x_1$  von  $L_1$ , so dass  $cross(G, x_0, x_1) \leq M$  ?

Zum Beweis der  $\mathcal{NP}$ -Vollständigkeit wird jedoch noch ein weiteres Entscheidungsproblem benötigt, von dem bereits bekannt ist, dass es  $\mathcal{NP}$ -vollständig ist. Können wir dieses Problem auf  $DCP$  transformieren, so ist es auch  $\mathcal{NP}$ -vollständig.

Wir werden dafür ein Problem benutzen, das sich mit sogenannten Kantenrückflussmengen in einem gerichteten Graphen beschäftigt. Dieses Problem ist  $\mathcal{NP}$ -vollständig.

#### 3.3 Das FAS-Problem ( $FASP$ )

Zuerst jedoch zur Definition einer Kantenrückflussmenge:

**Definition 3.3.1** Sei  $D = (U, B)$  ein gerichteter Graph. Eine Teilmenge  $B' \subset B$  ist eine **Kantenrückflussmenge (Feedback Arc Set, FAS)**, falls  $B'$  mindestens eine Kante aus jedem gerichteten Zykel in  $D$  enthält.

Wir definieren weiterhin  $\beta := |B|$  und  $\nu := |U|$ .

Damit kann nun das FAS-Problem formuliert werden:

Gegeben: Ein gerichteter Graph  $D = (U, B)$ , eine Schranke  $K \in \mathbb{N}$ .

Frage: Besitzt  $D$  eine Kantenrückflussmenge  $B'$  mit  $|B'| \leq K$  ?

**Satz 3.3.1** *Das Entscheidungskreuzungsproblem DCP ist  $\mathcal{NP}$  – vollständig.*

**Beweis:**

(a)  $DCP \in \mathcal{NP}$  :

Dazu müssen wir nur zeigen: Falls es eine Anordnung  $x_1$  von  $L_1$  gibt, die die Anforderungen erfüllt (dass also  $cross(G, x_0, x_1) \leq M$ ), so kann man einen Algorithmus angeben, der dies in polynomialer Laufzeit überprüfen kann.

Dieser Algorithmus wird in [1] ausführlich behandelt.

(b) Reduktion von  $FASP$ :

Gegeben sei nun ein gerichteter Graph, wie er für  $FASP$  benötigt wird. Aus diesem Graphen wollen wir einen Zwei-Lagen Graphen mit fester Anordnung  $x_1$  und einer Schranke  $M$  für die Anzahl der Kreuzungen konstruieren.

Es sei also  $D = (U, B)$  und  $K \in \mathbb{N}$ .

– Konstruktion der beiden Lagen  $L_0$  und  $L_1$ :

Definiere für alle Kanten  $a \in B$  eine Menge von sechs Knoten wie folgt:

$$C(a) := \{c_1(a), \dots, c_6(a)\}.$$

Damit definiere nun

$$L_0 := \bigcup_{a \in B} C(a) \quad \text{und} \quad L_1 := U.$$

– Konstruktion der Kantenmenge  $A$ :

Für alle Knoten  $u \in U$  und Kanten  $a \in B$  fügen wir folgendermaßen die Kanten zwischen den Lagen ein:

- (i) Falls  $a = (u, v)$  für einen Knoten  $v \in U$  ist, so verbinde  $u$  mit  $c_1(a)$  und  $c_5(a)$ ;
- (ii) Falls  $a = (v, u)$  für einen Knoten  $v \in U$  ist, so verbinde  $u$  mit  $c_2(a)$  und  $c_6(a)$ ;
- (iii) Falls  $a$  nicht inzident zu  $u$  ist, so verbinde  $u$  mit  $c_3(a)$  und  $c_4(a)$ .

Die Fälle (i)-(iii) sind nochmals in Abbildung 3 dargestellt.

– Konstruktion der Anordnung  $x_0$ :

Wir ordnen die Lage  $L_0$  so, dass sich die Mengen  $C(a)$  nicht gegenseitig schneiden. Innerhalb einer Menge sollen die Knoten gemäß ihrer Indizierung geordnet sein; es soll also gelten

$$x_0(c_i(a)) < x_0(c_j(a)) \quad \text{für} \quad 1 \leq i < j \leq 6 \quad \text{und} \quad \forall a \in B.$$

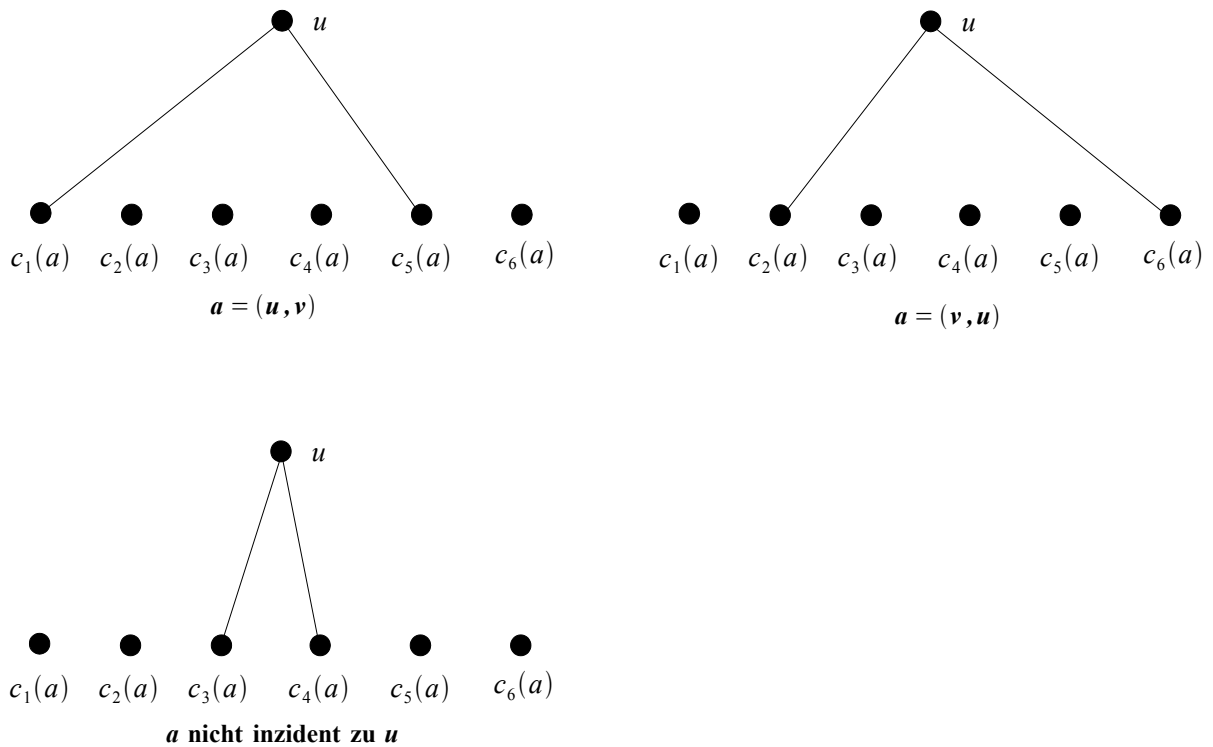


Abbildung 3: Fälle (i)-(iii)

- Konstruktion der Schranke  $M$ :  
Es soll gelten:

$$M := 4 \binom{\beta}{2} \binom{\nu}{2} + \beta \binom{\nu-2}{2} + 4\beta(\nu-2) + \beta + 2K.$$

Wir haben nun also aus einem gerichteten Graphen  $D$  und aus der oberen Schranke für die Kantenrückflussmenge  $K$  einen neuen Graphen konstruiert, wie er für  $DCP$  gebraucht wird. Als Letztes muß noch gezeigt werden, dass der gegebene Graph  $D$  des  $FASP$  genau dann eine Ja-Instanz ist, wenn der konstruierte Zwei-Lagen Graph  $G$  ebenfalls eine Ja-Instanz von  $DCP$  ist.

**Satz 3.3.2** Sei  $D = (U, B)$  ein gerichteter Graph,  $K \in \mathbb{N}$ , und  $G = (L_0, L_1, A)$  der konstruierte Graph mit fester Anordnung  $x_0$  und Schranke  $M$  wie gezeigt. Dann gilt:

Es gibt eine Kantenrückflussmenge  $B' \subset B$  mit  $|B'| \leq K \iff$   
 Es gibt eine Anordnung  $x_1$  von  $L_1$  mit  $\text{cross}(G, x_0, x_1) \leq M$ .

Dazu benötigen wir folgendes Lemma:

**Lemma 3.3.3** Sei  $x_1$  eine Anordnung von  $L_1$  und  $B' := \{(u, v) \in B : x_1(u) > x_1(v)\}$ . Dann gilt:

$$\text{cross}(G, x_0, x_1) = 4 \binom{\beta}{2} \binom{\nu}{2} + \beta \binom{\nu-2}{2} + 4\beta(\nu-2) + \beta + 2|B'|.$$

Damit kann nun der vorherige Satz bewiesen werden:

**Beweis von 3.3.2:**

” $\implies$ ” Sei  $B'$  eine Kantenrückflussmenge mit  $|B'| \leq K$ .  
 $\Rightarrow D' := (U, B \setminus B')$  ist azyklisch.

Also können wir auf  $D'$  eine topologische Sortierung durchführen und legen diese dann als die gesuchte Anordnung  $x_1$  fest; es gilt somit

$$x_1(u) < x_1(v) \quad \forall (u, v) \in B \setminus B'.$$

Da nun aber  $B'$  eine Kantenrückflussmenge ist, muss für alle  $a \in B'$  gelten, dass  $x_1(u) > x_1(v)$  ist. Somit genügt  $B'$  den Voraussetzungen des Lemmas; wir können also folgern

$$\text{cross}(G, x_0, x_1) \leq M,$$

da ja  $|B'| \leq K$  ist.

” $\impliedby$ ” Sei nun  $x_1$  eine Anordnung von  $L_1$  mit  $\text{cross}(G, x_0, x_1)$ . Wir definieren eine Menge  $B' := \{(u, v) \in B : x_1(u) > x_1(v)\}$  und betrachten wieder den Graphen  $D' := (U, B \setminus B')$ . Gäbe es in diesen Graphen einen gerichteten Zykel, so müsste für mindestens einer Kante  $(u, v)$  auf diesem Zykel gelten, dass  $x_1(u) > x_1(v)$  ist. Diese Kante liegt aber in  $B'$  und ist somit nicht in  $D'$ .

Mit Hilfe des Lemmas kann weiterhin gezeigt werden, dass  $|B'| \leq K$  ist.  $\square$

**Beweis von 3.3.3:**

Seien zunächst  $a, b \in A, a \neq b$ .

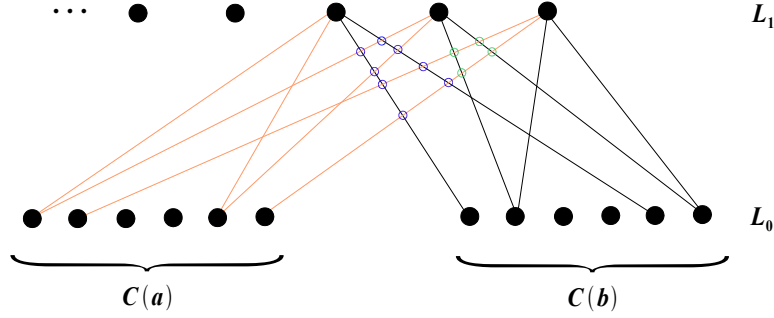


Abbildung 4: Kreuzungen zwischen  $C(a)$  und  $C(b)$

- (a) Als Erstes werden die Kreuzungen der Kanten betrachtet, die inzident zu  $C(a)$  und die inzident zu  $C(b)$  sind. (Siehe Abbildung 4.)

Deren Anzahl ergibt sich aus

$$4 + 8 + \dots + 4 \cdot (\nu - 1) = 4 \sum_{i=1}^{\nu-1} i = 4 \binom{\nu}{2}.$$

Damit ergibt sich für alle Mengen  $C$  genau  $4 \binom{\beta}{2} \binom{\nu}{2}$  Kreuzungen.

- (b) Als nächstes werden die Kreuzungen innerhalb einer Menge  $C(a)$  betrachtet: Zuerst bestimmen wir die Anzahl der Kreuzungen der Kanten inzident zu  $c_3(a)$  und inzident zu  $c_4(a)$ .

Aus der Konstruktion von  $G$  wissen wir:  $\nu - 2$  Knoten aus  $L_1$  sind inzident zu diesen beiden Knoten. Damit ist die Anzahl der Kreuzungen gerade

$$\sum_{i=0}^{\nu-3} i = \binom{\nu-2}{2}$$

und für alle Mengen  $C$  gibt es somit  $\beta \binom{\nu-2}{2}$  Kreuzungen zwischen diesen Kanten.

- (c) Sei nun  $a = (u, v)$ .

In  $G$  existieren laut Konstruktion die Kanten  $a_1 = (u, c_1(a))$ ,  $a_2 = (u, c_5(a))$ ,  $a_3 = (v, c_2(a))$  und  $a_4 = (v, c_6(a))$ . Nun bestimmen wir die Anzahl der Kreuzungen dieser vier Kanten und den Kanten, die inzident zu  $c_3(a)$  und  $c_4(a)$  sind. Da jede ausgehende Kante von  $c_3(a)$  und  $c_4(a)$  entweder  $a_1$  oder  $a_2$  schneidet sowie entweder  $a_3$  oder  $a_4$ , ergeben sich  $2 \cdot 2(\nu - 2)$  Kreuzungen; für den ganzen Graphen sind es also  $4\beta(\nu - 2)$  Kreuzungen zwischen diesen Kanten.

- (d) Als Letztes fehlen nun noch die Kreuzungen zwischen  $a_1, a_2, a_3$  und  $a_4$ . Hier gilt: Falls  $x_1(u) < x_1(v)$  ist, gibt es eine Kreuzung, und falls  $x_1(u) > x_1(v)$  gibt es

3 Kreuzungen. Da der letzte Fall genau  $|B'|$ -mal vorkommt, existieren für den gesamten Graphen  $\beta + 2|B'|$  Kreuzungen dieser Kanten.

Damit ergibt sich für die Gesamtanzahl der Kantenkreuzungen gerade

$$4\binom{\beta}{2}\binom{\nu}{2} + \beta\binom{\nu-2}{2} + 4\beta(\nu-2) + \beta + 2|B'|. \quad \square$$

Wir haben nun also gesehen, dass das Entscheidungskreuzungsproblem  $\mathcal{NP}$ -vollständig und damit das Kreuzungsproblem selbst  $\mathcal{NP}$ -schwer ist; es gibt also wahrscheinlich keinen Algorithmus, der in polynomialer Zeit das gesuchte Minimum für die Anzahl der Kreuzungen in einem gegebenen Zwei-Lagen Graphen berechnen kann. Jedoch gibt es einige Heuristiken, die eine Reihenfolge der Knoten in der oberen Lage festlegen und dabei nah beim Optimum bleiben.

Wir werden im folgenden zwei solche Heuristiken und deren Gütegarantie kennenlernen. Dabei wird sich zeigen, dass die Letztere höchstens dreimal mehr Kreuzungen als das gesuchte Minimum produziert.

## 4 Heuristiken

Zunächst aber kommen wir zu einer häufig benutzten Methode, der sogenannten Schwerpunkt-Heuristik. Dabei berechnen wir die  $x_1$ -Koordinate eines Knotens in  $L_1$  aus den Mittelwert der  $x_0$ -Koordinaten seiner Nachbarn:

### 4.1 Die Schwerpunkt-Heuristik

**Definition 4.1.1** Wir definieren für  $u \in L_1$

$$x_1(u) := avg(u) := \frac{1}{d_u} \sum_{v \in N_u} x_0(v).$$

Weiterhin bezeichne  $avg(G, x_0)$  die Anzahl der Kreuzungen im Graphen  $G$ , der durch die Schwerpunkt-Heuristik erzeugt wurde.

Ein Graphenlayout, das durch diese Schwerpunkt-Heuristik entstanden ist, kann allerdings sehr viele Kreuzungen beinhalten, wenn die Knoten aus der unteren Lage  $L_0$  ungleich verteilt sind. Abbildung 5 zeigt einen solchen Graphen, bei dem  $avg(G, x_0) = n - 4$ , aber  $opt(G, x_0) = 1$  ist.

In den meisten Programmen entstehen allerdings keine solchen grossen Abstände in der unteren Lage, da die  $x_0$ -Koordinaten der Knoten in  $L_0$  als bijektive Abbildung auf die Menge  $\{1, 2, \dots, \max\{|L_0|, |L_1|\}\}$  realisiert wird. Somit haben die Knoten immer den gleichen Abstand zu ihren Nachbarn.

Fordert man diese Bedingung an die Lage  $L_0$ , so zeigt der Graph in Abbildung 6 den worst

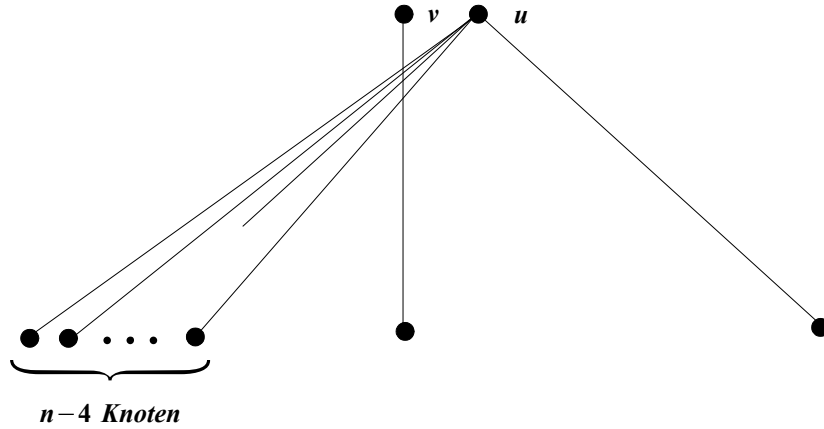


Abbildung 5: Graph mit  $\frac{avg(G, x_0)}{opt(G, x_0)} = n - 4$

case für die Anzahl der Kreuzungen: Hier ist  $avg(G, x_0) = k(k - 1)$  und  $opt(G, x_0) = k$ ; somit gilt  $\frac{avg(G, x_0)}{opt(G, x_0)} = k - 1 = \sqrt{n} - 2$ .

Damit zeigt sich auch, dass die Gütegarantie der Schwerpunkt-Heuristik nicht konstant sein kann, sondern von der Menge der Knoten abhängt. Im folgenden Satz wird die asymptotisch obere Schranke der Gütegarantie näher beschrieben.

**Satz 4.1.2** *Für alle Graphen  $G$  und Anordnungen  $x_0$  gilt:*

$$\frac{avg(G, x_0)}{opt(G, x_0)} \in \mathcal{O}(\sqrt{n}).$$

Der Beweis dieses Satzes ist äusserst aufwendig und schwierig. Es soll an dieser Stelle auf ihn verzichtet werden zugunsten der nächsten Heuristik, die tatsächlich eine konstante Schranke für die Anzahl der Kreuzungen besitzt und dessen Beweis demselben Prinzip folgt wie der Beweis von 4.1.2.

## 4.2 Die Median-Heuristik

Diese Heuristik setzt jeden Knoten aus der oberen Lage  $L_0$  über denjenigen Nachbarn, der in ihrer Reihenfolge in der Mitte vorkommt:

**Definition 4.2.1** *Es sei zunächst  $u \in L_1$  und  $N_u = \{v_1, v_2, \dots, v_j\}$  mit  $x_0(v_1) \leq x_0(v_2) \leq \dots \leq x_0(v_j)$ .*

*Nun definiere*

$$x_1(u) := med(u) := x_0(v_{\lfloor j/2 \rfloor}).$$

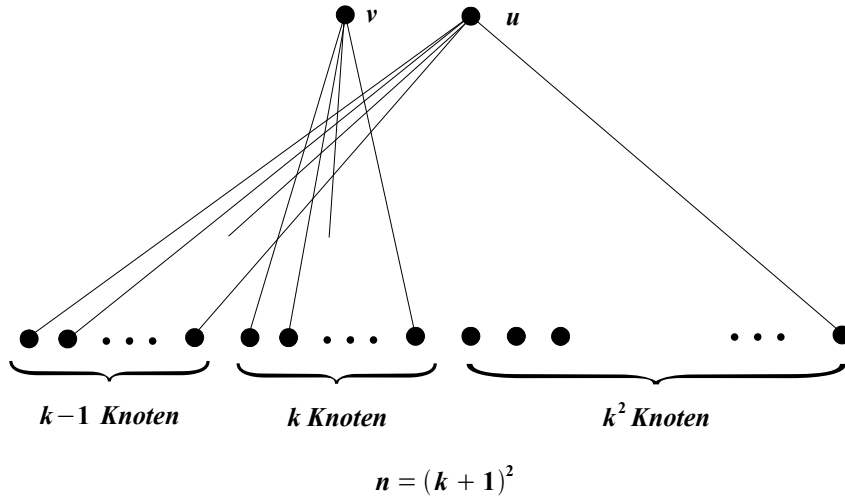


Abbildung 6: Graph mit  $\frac{\text{avg}(G, x_0)}{\text{opt}(G, x_0)} = \sqrt{n} - 2$

Für  $N_u = \emptyset$  setze man  $\text{med}(u) = 0$ . Falls  $\text{med}(u) = \text{med}(v)$  für  $u \neq v$  gilt, so werden die Knoten in  $x$ -Richtung um einen kleinen Wert verschoben. Dabei soll gelten: Besitzt einer der Knoten einen ungeraden und der andere einen geraden Grad, so erhält der Knoten ungeraden Grades die kleinere  $x$ -Koordinate (liegt also links von dem anderen Knoten).

**Satz 4.2.2** Für alle Graphen  $G$  und Anordnungen  $x_0$  gilt:

$$\text{med}(G, x_0) \leq 3 \cdot \text{opt}(G, x_0).$$

Für den Beweis dieser Gütegarantie wird noch die sogenannte **Kreuzungsmatrix** benötigt:

**Definition 4.2.3** Sei  $C = (c_{uv})_{u,v \in L_1}$ ; Für  $u \neq v$  und  $x_1(u) < x_1(v)$  definiere

$$c_{uv} := |\{(u, t), (v, w)\} \subset A : x_0(t) > x_0(w)\}|.$$

An der  $uv$ -ten Stelle der Matrix stehen also die Anzahl der Kreuzungen zwischen den Kanten, die inzident zu  $u$  und den Kanten, die inzident zu  $v$  sind.

**Beweis von 4.2.2:**

Generell gilt: Um zu zeigen, dass eine Gütegarantie der Form

$$\text{cross}(G, x_0, x_1) \leq B \cdot \text{opt}(G, x_0)$$

mit einer Schranke  $B$  existiert, genügt es bereits zu beweisen, dass es eine Schranke  $B$  gibt, so dass für alle  $x_1(u) < x_1(v)$  gilt:

$$c_{uv} \leq B \cdot c_{vu};$$

denn zusammen mit den Gleichungen

$$\begin{aligned} \text{cross}(G, x_0, x_1) &= \sum_{x_1(u) < x_1(v)} c_{uv} \quad \text{und} \\ \text{opt}(G, x_0) &\geq \sum_{u,v} \min(c_{uv}, c_{vu}) \end{aligned}$$

erhält man durch Summieren über alle Paare  $(u, v)$  mit  $x_1(u) < x_1(v)$  die gewünschte Gütegarantie. (Zum Beweis dieser Ungleichungen siehe [3].)

Die gesuchte Ungleichung liefert folgendes Lemma:

**Lemma 4.2.4** Sei  $\iota_{uv} := |N_u \cap N_v|$  die Anzahl der gemeinsamen Nachbarn,  $\chi(t) := \frac{t(t-1)}{2}$  und  $\tau_{uv} := \begin{cases} 1 & \text{falls } \text{med}(u) = \text{med}(v) \\ 0 & \text{sonst.} \end{cases}$ .

Weiterhin seien  $u, v \in L_1$ ,  $\text{med}(u) \leq \text{med}(v)$ .

(a) Falls  $d_u$  und  $d_v$  gerade sind, gilt

$$\begin{aligned} 4c_{uv} &\leq 3d_u d_v - 2d_u - 8\chi\left(\frac{\iota_{uv}}{2}\right) \\ 4c_{vu} &\geq d_u d_v + 2d_u + 8\chi\left(\frac{\iota_{uv}}{2}\right) - (1 + 2\iota_{uv})\tau_{uv}. \end{aligned}$$

(b) Falls  $d_u$  gerade und  $d_v$  ungerade ist, gilt

$$\begin{aligned} 4c_{uv} &\leq 3d_u d_v - d_u - 8\chi\left(\frac{\iota_{uv}}{2}\right) \\ 4c_{vu} &\geq d_u d_v + d_u + 8\chi\left(\frac{\iota_{uv}}{2}\right) - (1 + 2\iota_{uv})\tau_{uv}. \end{aligned}$$

(c) Falls  $d_u$  ungerade und  $d_v$  gerade ist, gilt

$$\begin{aligned} 4c_{uv} &\leq 3d_u d_v - 2d_u - d_v - 2 - 8\chi\left(\frac{\iota_{uv}}{2}\right) \\ 4c_{vu} &\geq d_u d_v + 2d_u + d_v + 2 + 8\chi\left(\frac{\iota_{uv}}{2}\right) - (1 + 2\iota_{uv})\tau_{uv}. \end{aligned}$$

(d) Falls  $d_u$  und  $d_v$  ungerade sind, gilt

$$\begin{aligned} 4c_{uv} &\leq 3d_u d_v - d_u - d_v - 1 - 8\chi\left(\frac{\iota_{uv}}{2}\right) \\ 4c_{vu} &\geq d_u d_v + d_u + d_v + 1 + 8\chi\left(\frac{\iota_{uv}}{2}\right) - (1 + 2\iota_{uv})\tau_{uv}. \end{aligned}$$

Daraus kann nun die Abschätzung  $c_{uv} \leq 3c_{vu}$  abgeleitet werden:

Man beachte, das zusätzlich folgende Ungleichungen gelten:

$$\iota_{uv} \leq \min\{d_u, d_v\} \quad \text{und} \quad 8\chi\left(\frac{\iota_{uv}}{2}\right) \geq -1$$

Mit Hilfe dieser Ungleichungen folgt für die Fälle (c) und (d) die Behauptung.

Im Fall (b) gilt, dass  $u$  geraden und  $v$  ungeraden Grad hat. Also können wir  $\mathbb{E}$  annehmen, dass  $\tau_{uv} = 0$  ist (da hier  $u$  und  $v$  auseinander geschoben werden und somit  $x_1(u) < x_1(v)$  gilt.)

Im Fall (a) erhalten wir

$$4c_{uv} \leq 3d_u d_v - 3 = 3(d_u d_v - 1),$$

da  $d_u \geq 2$  ist, und

$$4c_{vu} \geq d_u d_v - 1$$

da  $2d_u + 8\chi\left(\frac{\iota_{uv}}{2}\right) - 2\iota_{uv} \geq 0$ . Somit gilt auch hier  $c_{uv} \leq 3c_{vu}$ . □

#### **Beweisidee von 4.2.4:**

Zunächst werden die Kanten, die zu  $u$  und  $v$  inzident sind, in vier Gruppen aufgeteilt:

$$\begin{aligned} \alpha &= \{\{u, w\} : x_0(w) \leq \text{med}(u)\} \\ \beta &= \{\{v, w\} : x_0(w) \geq \text{med}(v)\} \\ \gamma &= \{\{v, w\} : x_0(w) < \text{med}(v)\} \\ \delta &= \{\{u, w\} : x_0(w) > \text{med}(u)\} \end{aligned}$$

Die vier Gruppen sind nochmal in Abbildung 7 dargestellt.

Wir bezeichnen  $a = |\alpha|$ ,  $b = |\beta|$ ,  $c = |\gamma|$  und  $d = |\delta|$ . Weiterhin sei

$$\begin{aligned} \mathbf{e} &= \{\{u, w\} \in \alpha : \{v, w\} \in \gamma\} \\ \mathbf{f} &= \{\{v, w\} \in \beta : \{u, w\} \in \delta\} \\ \mathbf{g} &= \{\{v, w\} \in \gamma : \{u, w\} \in \delta\} \end{aligned}$$

und  $e = |\mathbf{e}|$ ,  $f = |\mathbf{f}|$  und  $g = |\mathbf{g}|$ .

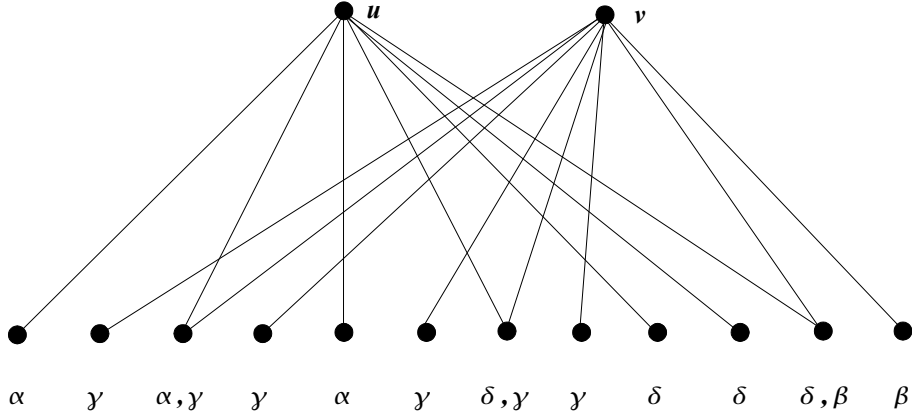


Abbildung 7: Beispielgraph mit den Gruppen  $\alpha$ ,  $\beta$ ,  $\gamma$  und  $\delta$ .

- Nun sei zuerst  $x_1(u) \leq x_1(v)$ . Es werden die Anzahl der Kreuzungen zwischen den Gruppen  $\alpha, \beta, \gamma$  und  $\delta$  nach oben abgeschätzt. Zwischen  $\alpha$  und  $\beta$  können keine Kreuzungen entstehen; genauso auch zwischen  $\alpha$  und  $\delta$ . Zwischen  $\alpha$  und  $\gamma$  gibt es die meisten Kreuzungen, wenn die Knoten aus  $L_0$  inzident zur Menge  $\gamma \setminus \mathbf{g}$  links von den Knoten inzident zu  $\alpha$  liegen. Beachtet man noch die Kanten aus  $\mathbf{e}$ , so führt dies zu  $a(c - g) - \chi(e + 1)$  Kreuzungen. (Siehe Abbildung 8.) Analoges gilt für die Kantenkreuzungen zwischen  $\beta$  und  $\delta$ . Für die maximale Anzahl von Kreuzungen zwischen  $\gamma$  und  $\delta$  gilt  $cd - \chi(g + 1)$ .

Damit erhält man die Ungleichung

$$c_{uv} \leq a(c - g) - \chi(e + 1) + b(d - g) - \chi(f + 1) + cd - \chi(g + 1).$$

Beachtet man, dass  $\iota_{uv} = e + f + g - \tau_{uv}$ ,  $\chi(t) \geq t^2/2$  und  $a \geq e$  sowie  $b \geq f$  ist, erhalten wir nach einigen Umformungen

$$c_{uv} \leq ac + bc + cd - 2\chi\left(\frac{\iota_{uv}}{2}\right).$$

- Es sei nun  $x_1(u) > x_1(v)$ . Wir schätzen jetzt die minimale Anzahl der Kreuzungen ab. Zwischen  $\alpha$  und  $\beta$  müssen nun Kreuzungen entstehen; falls  $med(u) = med(v)$  ist, muss eine Kreuzung abgezogen werden. Damit gilt

$$c_{vu} \geq ab + \chi(e) + \chi(f) + \chi(g) + ef + fg - \tau_{uv}.$$

Wir betrachten nun Fall (a) des Lemmas; es sollen also sowohl  $d_u$  als auch  $d_v$  gerade sein. Das bedeutet, dass  $a = d_u/2$  und  $b = (d_v - 2)/2$  gilt. Damit erhalten wir die Ungleichungen wie in (a) behauptet.  $\square$

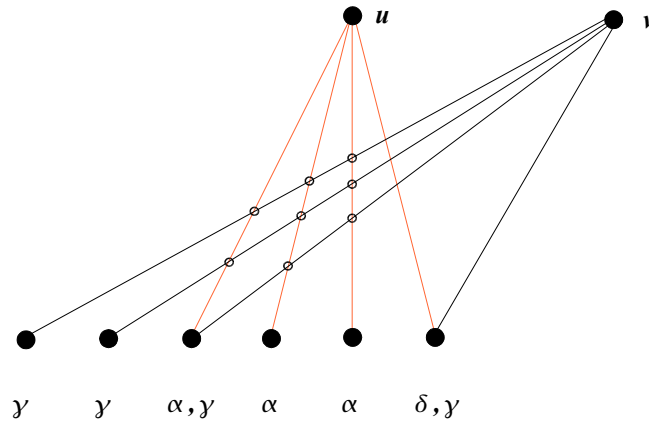


Abbildung 8: Kreuzungen zwischen  $\alpha$  und  $\gamma$ .

## 5 Bemerkungen

Es wurde in Kapitel 3 gezeigt, dass das Kreuzungsproblem in einem Zwei-Lagen Graphen, bei dem eine Lage fest bleibt und die andere permutiert wird,  $\mathcal{NP}$ -schwer ist. Die Schwerpunkt-Heuristik aus 4.1, die oft für dieses Problem verwendet wird, hatte allerdings nur eine asymptotische obere Schranke für die Gütegarantie und kann daher ein schlechtes Layout für einen Graphen liefern. Dagegen hatte die Median-Heuristik aus 4.2 eine konstante obere Schranke für die Anzahl der Kreuzungen und kann auch in Linearzeit implementiert werden.

Es hat sich allerdings gezeigt, dass die beiden Heuristiken im Allgemeinen ähnliche Layouts und daher auch etwa dieselbe Anzahl von Kantenkreuzungen erzeugen. Oft wird dabei auch eine Kombination der beiden Methoden verwendet.

Trotz der  $\mathcal{NP}$ -Schwere des Problems ist es möglich, ein Layout mit der minimalen Anzahl der Kantenkreuzungen zu erzeugen. Dies wird mit Hilfe eines FPT-Algorithmus realisiert, der in [2] genauer beschrieben ist.

## Literatur

- [1] Barth, Jünger, and Mutzel. Simple and efficient bilayer cross counting. *Proc. GD'02, Springer LNCS 2528*, pages 130–141, 2002.
- [2] Vida Dujmovic and Sue Whitesides. An efficient fixed parameter tractable algorithm for 1-sided crossing minimization. *Algorithmica 40*, 2004.
- [3] P. Eades and D. Kelly. Heuristics for reducing crossings in 2-layered networks. *Ars Combinatoria 21A*, pages 89–98, 1986.
- [4] Peter Eades and Nicholas C. Wormald. Edge crossings in drawings of bipartite graphs. *Algorithmica 11*, pages 379–403, 1994.