

Ein neuer Algorithmus zur Beschriftung von Punkten mit je zwei Kreisen

Michael Thon*

Forschungsarbeit in Zusammenarbeit mit Alexander Wolff[†] und Yinfeng Xu[‡]

Fachgruppe 0.1.2 Algorithmische Geometrie

Zusammenfassung

Bei dem von uns betrachteten Problem geht es darum, bei gegebener Menge P von n Punkten in der Ebene $2n$ gleichgroße (offene) Kreise so zu plazieren, dass jeder Punkt genau zwei Kreise berührt, sich keine der Kreise schneiden und die Größe der Kreise maximal ist.

Es ist bekannt, dass eine Approximation der Beschriftungsgröße besser als $\approx 73\%$ der optimalen Größe NP-schwer ist. Unser Algorithmus verbessert den bisher besten erzielten Approximationsfaktor von $\approx 51\%$ auf $(1 + \sqrt{3})/4 \approx 68\%$. Dabei behalten wir die $O(n \log n)$ Laufzeit- und $O(n)$ Speichergrenze des vorigen Algorithmus bei.

1 Einleitung

Bei der Visualisierung von Daten, etwa bei der Erstellung von Landkarten, Diagrammen und technischen Zeichnungen, ist das Beschriften der Daten ein wesentlicher und zeitaufwendiger Bestandteil. Es ist also von besonderem Interesse, den Prozess der Beschriftung durch Algorithmen zu automatisieren. Bevor man jedoch solche Algorithmen entwickeln kann, benötigt man klare Kriterien dafür, wie eine gute Beschriftung auszusehen hat. Solche Kriterien wurden 1962 von dem schweizer Kartographen Imhof für den speziellen Fall der Beschriftung von Landkarten aufgestellt [4], welche jedoch im Wesentlichen auf das allgemeine Beschriftungsproblem übertragbar sind.

Imhof schreibt, dass eine Landkarte aus vielen, in der Regel einigen hundert bis einigen tausend Objekten besteht, welche möglichst eindeutig zu beschriften sind. Diese Objekte lassen sich ferner in Typen, wie Flüsse, Landesgrenzen, Städte, etc. einteilen, deren Unterscheidung maßgeblich durch gute Beschriftung erleichtert wird. Die Qualität einer Karte hängt also von einer guten Beschriftung ab. Dabei stellt Imhof fest, dass nicht nur die Eigenschaften einer Beschriftung wie Schrifttyp, Farbe, Größe, Form, etc., welche für einen Typ von Objekten einheitlich sein sollten, zur Lesbarkeit einer Karte beitragen, sondern auch die Plazierung. Die Plazierung habe so zu erfolgen, dass sich die einzelnen Beschriftungen gegenseitig nicht stören, und die Zuordnung zwischen Objekt und Beschriftung klar ersichtlich ist. Daraus ergibt sich, dass die Plazierungen der Beschriftungen bei dicht beschrifteten Karten voneinander abhängen, welches die Berechnung einer guten Plazierung zum wesentlichen Problem bei der Landkartenbeschriftung macht.

Imhof unterteilt das Beschriftungsproblem weiter, da es drei allgemeine Typen von Objekten gibt, nämlich Flächen (z.B. Länder), Linien (z.B. Flüsse) und Punkte (z.B. Städte). Während es bei der Beschriftung von Gebieten vor allem darum geht, die Beschriftung möglichst innerhalb des Objekts und über dieses erstreckt zu plazieren, ist bei der Punktbeschriftung die Nähe der Beschriftung zum Objekt das wesentliche Kriterium.

Nun kann man das allgemeine Beschriftungsproblem so formulieren, dass beliebig viele Beschriftungskandidaten pro Objekt zugelassen werden, oder nur einige diskrete Beschriftungskandidaten pro Objekt (bei der Punktbeschriftung können dies z.B. vier sein, nämlich links-oben, rechts-oben, links-unten und rechts-unten vom jeweiligen Punkt). Außerdem ergeben sich je nach Form der Beschriftung (rechteckig, rund, ...), Ausrichtung (z.B. horizontal) und Anzahl der Beschriftungen pro Objekt wiederum viele Teilprobleme, bei denen wiederum entweder die Beschriftungsgröße oder die Anzahl der überhaupt beschrifteten Objekte maximiert wird.

*Institut für Mathematik und Informatik, Ernst-Moritz-Arndt-Universität, Greifswald, Deutschland. mithon42@gmx.de

[†]Institut für Mathematik und Informatik, Ernst-Moritz-Arndt-Universität, Greifswald, Deutschland. awolff@mail.uni-greifswald.de

[‡]School of Management, Xi'an Jiaotong University, Xi'an, China. yfxu@xjtu.edu.cn

Selbst diese speziellen Beschriftungsprobleme sind oft NP-schwer, so dass es darum geht, effiziente Approximationsalgorithmen oder Heuristiken zu entwickeln, wobei es nun sowohl interessant ist, für die Praxis gut funktionierende Beschriftungsalgorithmen zu entwickeln, als auch prinzipielle Aussagen über die Approximierbarkeit der Probleme zu treffen.

Für die Platzierung von Beschriftungen wurden bisher zahlreiche Lösungsansätze vorgeschlagen, z.B. Expertensysteme, Null-Eins-Integerprogrammierung, sowie Simulated Annealing und andere inkrementelle Algorithmen. Wir stellen hier einen Approximationsalgorithmus für ein Teilproblem vor.

2 Unser spezielles Beschriftungsproblem

Definition 1 (Punktbeschriftung mit je zwei Kreisen) *Man plaziere bei gegebener Menge P von n Punkten in der Ebene $2n$ gleichgroße (offene) Kreise so, dass jeder Punkt genau zwei Kreise berührt, sich keine der Kreise schneiden und die Größe der Kreise maximal ist.*

Wir betrachten also, um noch einmal auf die obige Einteilung zurückzukommen, die Punktbeschriftung mit je zwei kreisförmigen Beschriftungen, wobei beliebig viele Beschriftungskandidaten zugelassen sind und die Ausrichtung der Beschriftungen beliebig ist. Dabei maximieren wir die einheitliche Größe der Beschriftungen, die im Prinzip wie starre Achten aussehen.

Die Beschriftung von Punkten mit je zwei Beschriftungen wird z.B. durch Wetterkarten motiviert, wo jede Stadt mit zwei achsenparallelen Beschriftungen – dem Namen der Stadt und der Temperatur bzw. Regenwahrscheinlichkeit – beschriftet werden muss. Kakoulis und Tollis haben sich als erste mit Algorithmen beschäftigt, die Objekte mit mehr als einer Beschriftung versehen. Sie haben zwei Heuristiken vorgestellt, die Knoten und Kanten in einer Graphzeichnung jeweils mit mehreren Rechtecken beschriften [5]. Mittlerweile wurde die Punktbeschriftung mit je zwei Quadraten recht ausführlich von Zhu und Poon [8] sowie von Qin et al. [6] behandelt. Die Punktbeschriftung mit kreisförmigen Beschriftungen ist zwar für die Praxis nicht von so großer Bedeutung, dafür aber ein mathematisch interessantes Problem. Das Problem, jeden Punkt mit *einer* kreisförmigen Beschriftung zu versehen, wurde bereits ausführlich untersucht [7, 3].

Für das von uns betrachtete Problem der Größenmaximierung kreisförmiger Beschriftungen, zwei pro Punkt, gaben Zhu und Poon den ersten Approximationsalgorithmus an [8]. Sie erzielten einen Approximationsfaktor von $\frac{1}{2}$. Dabei vergleichen sie ihre Lösung mit einer oberen Schranke für die Größe der optimalen Lösung. Wie alle nachfolgenden Algorithmen beruht auch ihr Algorithmus auf der Tatsache, dass es um jeden Eingabepunkt p eine Region gibt, in der p beschriftet werden kann, ohne die Region eines anderen Punktes zu schneiden.

Vor kurzem gelang es Qin et al. dieses Resultat zu verbessern [6]. Sie gaben einen Approximationsalgorithmus mit einem Approximationsfaktor von $\frac{1}{1+\cos 18^\circ} \approx 0,5125$ an, den sie anhand derselben oberen Schranke für die maximale Beschriftungsgröße berechnen, die auch Zhu und Poon benutzten [8]. Ferner zeigen sie, dass es NP-schwer ist, die optimale Beschriftungsgröße besser als mit einem Faktor von $\approx 73,21\%$ zu approximieren. Die Regionen, in welche sie die Beschriftungen plazieren, sind die Zellen des Voronoi-Diagramms der Punktmenge P . Dabei berechnen sie das Voronoi-Diagramm nicht explizit, sondern nutzen bestimmte Eigenschaften der dazu dualen Delauney-Triangulierung aus.

Das Voronoi-Diagramm ist eine wohlbekanntes und vielseitige geometrische Datenstruktur [2]. Die Voronoi-Zelle eines Punktes p der Eingabepunktmenge P ist die Menge der Punkte des \mathbb{R}^2 , dessen euklidischer Abstand zu p kleiner als ihr Abstand zu jedem anderen Punkt aus P ist. Das Voronoi-Diagramm setzt sich aus den Voronoi-Zellen und den Kanten zwischen den Voronoi-Zellen zusammen. Eine Voronoi-Zelle wird immer durch ein konvexes Polygon (eventuell unbeschränkt) begrenzt. Die Voronoi-Zellen sind paarweise disjunkt, und die Anzahl der Kanten im Voronoi-Diagramm ist linear in der Anzahl der Punkte der Eingabepunktmenge.

Wir geben nun einen Algorithmus an, welcher auch jeden Punkt in seiner Voronoi-Zelle beschriftet, jedoch tun wir dies im Gegensatz zum vorigen Algorithmus optimal. Ferner vergleichen wir die von unserem Algorithmus erzielte Beschriftungsgröße direkt mit der optimalen Größe und nicht mit einer oberen Schranke. So erhalten wir einen Approximationsfaktor von $(1 + \sqrt{3})/4 \approx 68,3\%$, welches beinahe die Lücke zu obigem Resultat über die Nicht-Approximierbarkeit unseres Problems schließt. Die Laufzeit von $O(n \log n)$ und der Speicherbedarf von $O(n)$ des vorigen Algorithmus wird beibehalten.

3 Unser Algorithmus

Wir berechnen erst das Voronoi-Diagramm von P . Dann beschriften wir jeden Eingabepunkt p optimal in seiner Voronoi-Zelle $\text{Vor}(p)$. Zuletzt suchen wir uns von allen Beschriftungen den kleinsten Beschriftungsdurchmesser und verkleinern alle Beschriftungen auf diese Größe.

Zunächst zeigen wir, dass es eine Region $Z_{\text{free}}(p)$ um jeden Eingabepunkt p gibt, welche keinen weiteren Eingabepunkt enthält. Die Größe von $Z_{\text{free}}(p)$ ist nur vom Durchmesser d_{opt} der Kreise in der optimalen Beschriftung abhängig, welchen wir auf 1 setzen (dies können wir immer durch skalieren der Eingabepunkte erreichen).

Definition 2 Sei $C_{m,r}$ ein (offener) Kreis mit dem Radius r und dem Mittelpunkt m . Für jeden Punkt $p \in P$ sei $Z_{\text{free}}(p) = C_{Z_1, \frac{1}{2}\sqrt{3}} \cup C_{Z_2, \frac{1}{2}\sqrt{3}} \cup C_{p,1}$, wobei Z_1 und Z_2 die Mittelpunkte der zwei Beschriftungen L_1 und L_2 von p in einer fixierten optimalen Lösung C_{opt} sind. $Z_{\text{label}}(p)$ sei die Region, die man bei Skalierung von $Z_{\text{free}}(p)$ um einen Faktor von $\frac{1}{2}$ relativ zu p erhält. (siehe Abb. 1)

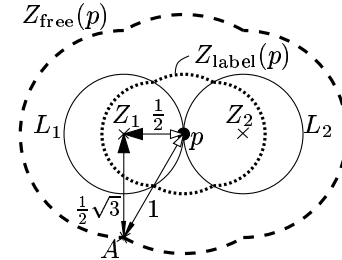


Abb. 1: $Z_{\text{free}}(p)$ und $Z_{\text{label}}(p)$

Die Begrenzung von $Z_{\text{free}}(p)$ ist gerade der Ort aller Eingabepunkte, die für $d_{\text{opt}} = 1$ so nahe an p liegen wie möglich. Daraus folgt:

Lemma 1 $Z_{\text{free}}(p)$ enthält keine Eingabepunkte außer p .

Hieraus, und aufgrund der Definition der Voronoi-Zelle $\text{Vor}(p)$ von p und der Region $Z_{\text{label}}(p)$ ergibt sich, dass $Z_{\text{label}}(p)$ vollständig in $\text{Vor}(p)$ liegt.

Es ist leicht, den größtmöglichen Beschriftungsdurchmesser d_{lower} zu berechnen, so dass die zwei Beschriftungen von p vollständig in $Z_{\text{label}}(p)$ enthalten sind. Also:

Lemma 2 Für jeden Eingabepunkt p gibt es zwei disjunkte Kreise vom Durchmesser $d_{\text{lower}} = \frac{1+\sqrt{3}}{4}$, welche p berühren und vollständig in $Z_{\text{label}}(p)$ liegen.

Aus Platzgründen beweisen wir nur das folgende, aus algorithmischer Sicht zentrale Lemma 3.

Lemma 3 Ein Punkt p im Innern des konvexen Polygons G mit m Ecken v_1, \dots, v_m kann mit zwei gleichgroßen disjunkten Kreisen maximaler Größe in einer Laufzeit von $O(m \log m)$ und mit einem Speicherbedarf von $O(m)$ vollständig innerhalb des Polygons G beschriftet werden.

Beweis. Betrachten wir die stetige Funktion f , welche einem Winkel $0 \leq \alpha < 2\pi$ den größtmöglichen Radius r_α zuordnet, so dass C_{z_α, r_α} den Punkt p berührt und vollständig im Polygon G liegt, wobei der Mittelpunkt z_α von C_{z_α, r_α} , der Punkt p und die rechtsgerichtete Horizontale \vec{h}_p durch p einen Winkel der Größe α bilden.

Da wir p mit zwei disjunkten, gleichgroßen Kreisen in G beschriften wollen, müssen wir den Winkel α bestimmen, wo $\min(r_\alpha, r_{\alpha+\pi})$ maximal ist. Dies erzielen wir durch die Berechnung des Maximums der unteren Einhüllenden von f und f' , wobei $f'(\alpha) = f(\alpha + \pi \bmod 2\pi)$. Da f (und somit auch f') aus höchstens $2m$ monotonen Abschnitten besteht, können die abschnittswise unteren Einhüllenden von f und f' und somit deren Maximum in linearer Laufzeit bestimmt werden. Schließlich sind die Positionen der Mittelpunkte der größtmöglichen Beschriftungen von p innerhalb des Polygons G durch den errechneten Winkel α eindeutig bestimmt. Der Radius der Beschriftungen ist natürlich r_α .

Es bleibt zu zeigen, wie man f berechnet. Da für unsere Zwecke eine Laufzeit von $O(m \log m)$ ausreichend ist, können wir eine sehr allgemeine Herangehensweise wählen, nämlich wiederum die Berechnung von unteren Einhüllenden. Sei ℓ_i die Gerade durch die Ecken v_i und $v_{i+1 \bmod m}$ des Polygons G , und sei β_i der Winkel zwischen ℓ_i (nach oben gerichtet) und \vec{h}_p , siehe Abb. 2. Sei f_i die Funktion, welche $0 \leq \alpha < 2\pi$ den Radius des eindeutig bestimmten Kreises zuordnet, der ℓ_i und \vec{h}_p berührt, wobei die Gerade durch p und den Mittelpunkt $c_i(\alpha)$ des Kreises mit der Horizontalen \vec{h}_p den Winkel α bildet. Die Funktion f_i hat genau ein Minimum bei $\beta_i + \pi/2$ (zu dem sie symmetrisch modulo 2π ist) und eine Definitionslücke bei $\beta_i + 3\pi/2$.

Zwei Funktionen f_i und f_j mit $i \neq j$ schneiden sich genau zwei Mal, nämlich bei den Winkeln, welche den Kreisen entsprechen, die ℓ_i, ℓ_j und p berühren, siehe Abb. 3. (Falls $\ell_i \parallel \ell_j$, so liegt p

zwischen ℓ_i und ℓ_j .) Nun gibt es eine Repräsentation von f_i (welche in konstanter Laufzeit berechnet werden kann), so dass die Schnittpunkte von f_i mit f_j für alle i, j in konstanter Zeit berechnet werden können. Daher können wir den einfachen Divide-and-conquer-Algorithmus [1] benutzen, um die untere Einhüllende von f_1, \dots, f_m in $O(m \log m)$ Laufzeit mit $O(m)$ Speicherbedarf zu berechnen. Die untere Einhüllende dieser Funktionen entspricht f , weil das Minimum von f_1, \dots, f_m für jedes α gerade den Radius des größtmöglichen Kreises liefert, der keine der Kanten ℓ_1, \dots, ℓ_m schneidet und dessen Mittelpunkt in der gewünschten Richtung in Abhängigkeit von p liegt. \circledast

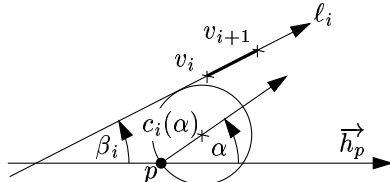


Abb. 2: Notation.

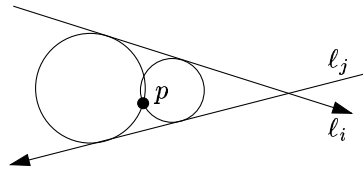


Abb. 3: Es gibt genau zwei Kreise, die ℓ_i , ℓ_j und p berühren.

Satz 1 Eine Menge P von n Punkten in der Ebene kann mit $2n$ disjunkten kreisförmigen Beschriftungen, zwei pro Punkt, mit einem Durchmesser $d_{\text{lower}} = (1 + \sqrt{3})/4 d_{\text{opt}} \approx 68,3\%$ der optimalen Beschriftungsgröße in $O(n \log n)$ Zeit und mit linearem Speicherbedarf beschriftet werden.

Beweis. Zunächst berechnen wir das Voronoi-Diagramm von P in $O(n \log n)$ Zeit und mit linearem Speicherbedarf [2]. Dann benutzen wir den im Beweis des zentralen Lemma 3 beschriebenen Algorithmus, um für jeden Eingabepunkt p das Kreispaar maximaler Größe zu berechnen, welches p beschriftet und dabei vollständig in $\text{Vor}(p)$ liegt. Sei d_p der Durchmesser dieser beiden Kreise. Sei m_p die Anzahl der Kanten in $\text{Vor}(p)$. Da die Komplexität des Voronoi-Diagramms linear ist [2], erhalten wir $\sum_{p \in P} m_p = O(n)$. Folglich können wir laut Lemma 3 alle Beschriftungen in $O(\sum_{p \in P} m_p \log m_p) = O(n \log n)$ Zeit und mit $O(n)$ Speicherbedarf plazieren. Wir setzen $d_{\text{algo}} = \min_{p \in P} d_p$ und gehen noch einmal alle Eingabepunkte durch. Wir skalieren die Beschriftungen jedes Punktes p mit dem Faktor $\frac{d_{\text{algo}}}{d_p}$. Wir geben diese skalierten Beschriftungen aus, welche nun alle den Durchmesser d_{algo} haben. Wegen $\frac{d_{\text{algo}}}{d_p} \leq 1$ liegen die skalierten Beschriftungen alle in den entsprechenden Voronoi-Zellen und können sich nicht schneiden. Es bleibt zu zeigen, dass $d_{\text{algo}} \geq d_{\text{lower}}$. Lemma 1 garantiert, dass $Z_{\text{label}}(p)$ vollständig in $\text{Vor}(p)$ liegt. Ferner besagt Lemma 2, dass es zwei disjunkte Beschriftungen für p , beide mit dem Durchmesser d_{lower} , gibt, die vollständig in $Z_{\text{label}}(p)$ – und somit in $\text{Vor}(p)$ – liegen. Dies gilt auch für den Eingabepunkt q , der die kleinste Beschriftung vor der Skalierung erhält, also $d_q = d_{\text{algo}}$. Andererseits wissen wir, dass $d_q \geq d_{\text{lower}}$, da d_q der größtmögliche Beschriftungsdurchmesser eines vollständig in $\text{Vor}(p)$ liegenden Paares von Beschriftungen für q ist. \circledast

Literatur

- [1] P. Agarwal und M. Sharir. Davenport-Schinzel sequences and their geometric applications. In J.-R. Sack und J. Urrutia, Herausgeber, *Handbook of Computational Geometry*. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 1999.
- [2] F. Aurenhammer. Voronoi diagrams: A survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, Sept. 1991.
- [3] S. Doddi, M. V. Marathe, and B. M. Moret. Point labeling with specified positions. In *Proc. 16th Annu. ACM Sympos. Comput. Geom. (SoCG'00)*, Seiten 182–190, Hongkong, 12.–14. Juni 2000.
- [4] E. Imhof. Die Anordnung der Namen in der Karte. In *International Yearbook of Cartography*, Seiten 93–129, Bonn Bad Godesberg, 1962. Kirschbaum.
- [5] K. G. Kakoulis and I. G. Tollis. On the multiple label placement problem. In *Proc. 10th Canadian Conf. Computational Geometry (CCCG'98)*, Seiten 66–67, 1998.
- [6] Z. Qin, A. Wolff, Y. Xu, and B. Zhu. New algorithms for two-label point labeling. In *Proc. 8th Annu. Europ. Symp. on Algorithms (ESA'00)*, LNCS, Saarbrücken, 5.–8. Sept. 2000. Springer-Verlag. erscheint in Bände.
- [7] T. Strijk und A. Wolff. Labeling points with circles. *International Journal of Computational Geometry and Applications*, 2000. erscheint in Bände.
- [8] B. Zhu und C. K. Poon. Efficient approximation algorithms for multi-label map labeling. In A. Aggarwal und C. P. Rangan, Herausgeber, *Proc. 10th Annu. Int. Symp. on Algorithms and Computation (ISAAC'99)*, Band 1741 aus LNCS, Seiten 143–152, Chennai, India, 16.–18. Dez. 1999. Springer-Verlag.