RWTH Aachen **SS 2010**
Lehrgebiet Theoretische Informatik Übungsblatt 8
Kneis—Langer—Rossmanith 23.06.2010

Übung zur Vorlesung Effiziente Algorithmen

Tutoraufgabe 19

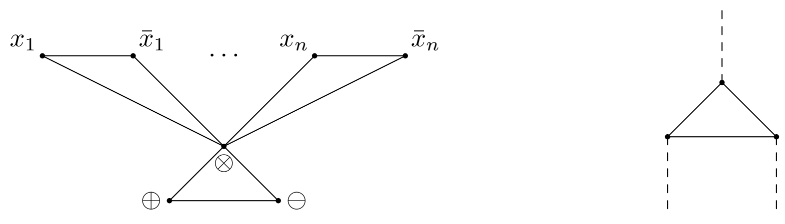
Sei $k \in \mathbf{N}$. Ein Graph heißt k-färbbar, wenn man die Knoten des Graphens so mit k Farben färben kann, daß niemals zwei adjazente Knoten die gleiche Farbe haben. Das zugehörige k-FÄRBBARKEIT-Problem besteht darin, für einen gegebenen G Graphen zu entscheiden, ob G k-färbbar ist.
Finden Sie für jedes $k \in \mathbf{N}$ polynomielle Algorithmen für das k-FÄRBBARKEIT-Problem oder beweisen Sie NP-Härte, indem Sie von 3-SAT reduzieren (das Erfüllbarkeitsproblem für aussagenlogische Formeln in KNF, bei denen jede Klausel höchstens drei Literale enthält).

Ansatz:
Für $k = 2$ entwerfen wir einen Greedy-Algorithmus, der eine Menge U von bereits gefärbten Knoten berechnet, bis alle Knoten des Graphens $G = (V, E)$ in U enthalten sind.

1. Solange $U \neq V$, finde einen Knoten $v \in U$, der einen Nachbarn $u \in V \setminus U$ hat.
2. Wenn kein solches v existiert, wähle einen beliebigen Knoten $v \in V \setminus U$, färbe ihn mit der Farbe 0 und setze $U := U \cup \{v\}$.
3. Sonst prüfe, ob u noch gefärbt werden kann. Wenn nicht, antworte „nein“. Sonst färbe ihn entsprechend (wegen v eindeutig) und setze $U := U \cup \{u\}$.

Da zwei adjazente Knoten unterschiedliche Farben haben müssen, aber nur zwei Farben zur Verfügung stehen, läßt sich Korrektheit dieses Ansatzes leicht nachweisen.

Für die NP-Härte von 3-FÄRBBARKEIT verwenden wir die zwei folgenden *gadgets*:



Gegeben ist eine aussagenlogische Formel in 3-KNF mit Variablen $x_1, \dots, x_n$. ObdA können wir annehmen, daß jede Klausel genau drei Literale enthält (sonst padden). Wir konstruieren nun einen Graphen wie folgt: Zunächst enthält der Graph das linke *gadget*, das über die Farbe $\otimes$ bestimmte Farben der Knoten $\oplus$ und $\ominus$ und entsprechende Farben der Variablen forciert (positiv und negativ).

Das rechte Gadget verwenden wir als logisches $\vee$ (oder), von denen wir für eine Klausel $\{l_1, l_2, l_3\}$, also $(l_1 \vee l_2) \vee l_3$, zwei benötigen und geeignet mit den Knoten der entsprechenden Literale verbinden. Schließlich verbinden wir den Ausgang des äußersten $\vee$-*gadget*s einer jeden Klausel, den wir unten als Ausgangsknoten der Klausel bezeichnen werden, mit den Knoten $\otimes$ und $\ominus$ des linken *gadget*s.

Durch eine leichte Fallunterscheidung stellen wir nun zunächst fest, daß das rechte *gadget* tatsächlich geeignet ein $\vee$ kodiert, genauer: Fixiere eine Farbe $\ominus$, die für *false* stehen soll. Wenn die Knoten, zu denen die beiden Eingänge führen, beide mit dieser Farbe $\ominus$ gefärbt sind, dann muß auch der Ausgangsknoten des *gadget*s in dieser Farbe gefärbt sein. Für alle anderen Fälle gibt es immer auch eine Färbung, in der der Ausgangsknoten mit einer weiteren Farbe $\oplus$ gefärbt ist.

Angenommen nun, die Formel ist erfüllbar. Färbe die beiden Knoten $\oplus$ und $\ominus$ mit zwei analog bezeichneten Farben $\oplus$ und $\ominus$, dann verbleibt für den Knoten $\otimes$ nur noch die dritte Farbe $\otimes$. Für jede Variable x_i Färben wir nun den Knoten x_i mit $\oplus$ und $\bar{x}_i$ mit $\ominus$, was auch noch gültig ist. Falls an mindestens einem Eingang eines beliebigen $\vee$-*gadgets* nun die Farbe $\oplus$ anlegt, wählen wir für den Ausgangsknoten die Farbe $\oplus$, ansonsten $\ominus$ (siehe Ausführungen oben). Das bedeutet insbesondere, daß nun auch der Ausgangsknoten jeder Klausel mit $\oplus$ gefärbt ist, so daß die Kanten zu den Knoten $\oplus$ und $\otimes$ korrekt sind.

Ist umgekehrt der so konstruierte Graph dreifärbar, dann sind die Knoten $\oplus$, $\ominus$ und $\otimes$ jeweils mit unterschiedlichen Farben gefärbt. Auch wissen wir, dass die Ausgangsknoten der Klauseln nicht mit $\otimes$ oder $\ominus$ gefärbt sind. Analog zu oben überlegen wir, daß damit auch für jede Klausel mindestens einer der Knoten, die zu den entsprechenden Literalen der Klausel gehören, ungleich $\ominus$, also $\oplus$ gefärbt sein muss. Verwende diese Farbkodierungen für eine Belegung der Variablen, dann ist klar, daß diese die Formel erfüllen.

Für die Härte von $k+1$-Färbbarkeit für $k > 2$ verwenden wir Induktion über k und eine Reduktion von k auf $k+1$, indem wir für einen gegebenen Graphen einen weiteren Knoten hinzufügen, den wir mit allen anderen Knoten verbinden.

Tutoraufgabe 20

Finden Sie möglichst gute Approximationsalgorithmen für MaxSAT und Triangle-Packing!

Lösung:

Für beide Probleme bietet sich ein Greedy-Algorithmus an. Für MaxSAT wählt man dazu eine beliebige Variable x aus, welche in den meisten Klauseln vorkommt. Tritt x häufiger auf als $\overline{x}$, so setzen wir x auf *true*, andernfalls auf *false*. Dies wird solange wiederholt, bis keine Klauseln übrig sind.

Da in jedem einzelnen Schritt mehr Klauseln erfüllt werden als komplett wegfallen erhalten wir mit diesem Ansatz eine 2-Approximation.

Im Fall von TrianglePacking wählen wir in jedem Schritt ein beliebiges Dreieck und entfernen die zugehörigen Kanten. Da jede Kante, also insbesondere auch jede entfernte Kante, nur in einem Dreieck der optimalen Lösung vorkommen kann, ergibt dies eine 3-Approximation.

Hausaufgabe 16 (10 Punkte)

Verwenden Sie eine LP-Relaxierung eines geeigneten ILPs für VERTEX COVER, um einen Approximationsalgorithmus für VERTEX COVER zu erhalten. Beweisen Sie seine Güte.

Lösung

Ziel ist es, eine 2-Approximation zu berechnen.Sei $G = (V, E)$ ein Graph. Wie schon bekannt, können wir die Minimierungsvariante von VERTEX COVER wie folgt als ILP formulieren:

Für jeden Knoten $v_i \in V$ benötigen wir eine Variable x_i. Die Zielfunktion ist dann $\min \sum_{v_i \in V} x_i$. Als Nebenbedingungen erhalten wir $x_i \in \{0, 1\}$ für alle $v_i \in V$ und $x_i + x_j \geq 1$ für jede Kante $\{v_i, v_j\} \in E$. In der Relaxierung als LP wird aus $x_i \in \{0, 1\}$ die Bedingung $x_1 \geq 0$.

Sei nun $\overline{x_1}, \ldots, \overline{x_n}$ eine optimale Lösung des relaxierten LPs.Um eine Approximationslösung für VERTEX COVER zu erhalten, fügen wir Knoten v_i genau dann dem Vertex Cover hinzu, wenn $\overline{x_i} \geq 0.5$.

Wir müssen nun zeigen, daß $V' = \{v_i \mid \overline{x_i} \geq 0.5\}$ ein Vertex Cover ist und daß V' höchstens doppelt soviele Knoten wie eine optimale Lösung enthält.

1. Sei $\{v_i, v_j\} \in E$ eine beliebige Kante. Da $\overline{x_1}, \ldots, \overline{x_n}$ eine zulässige Lösung des relaxierten LPs ist, gilt insbesondere $\overline{x_i} + \overline{x_j} \geq 1$. Dann ist aber entweder $\overline{x_i} \geq 0.5$ oder $\overline{x_j} \geq 0.5$, möglicherweise sogar beide. Daraus folgt, daß mindestens einer der Knoten v_i, v_j in V' enthalten ist.

   Somit sind alle Kanten zu V' inzident und V' ist ein Vertex Cover.

2. Ein minimales Vertex Cover C entspricht offensichtlich einer zulässigen Lösung des relaxierten LPs, indem wir $x_i = 1$ setzen, falls $v_i \in C$ und sonst $x_i = 0$. Somit gilt aber für die optimale Lösung L des relaxierten LPs $L \leq |C|$.

   Weiterhing gilt aber auch $|V'| \leq 2L$, da für jeden Knoten $v_i \in V'$ nach Definition $\overline{x_i} \geq 0.5$ gilt.

   Somit ist $|V'| \leq 2L \leq 2|C|$.

Dieser Algorithmus berechnet also eine 2-Approximation für die Optimierungsvariante des VERTEX COVER Problems.

Hausaufgabe 17 (10 Punkte)

In einem exotischen Land wird der Bildungsetat gekürzt. Die Regierung erachtet es von nun an als ausreichend, wenn von jedem Punkt im Lande eine Universität nicht weiter als d km entfernt ist. Von den bestehenden Universitäten $u_1, \ldots, u_n$ sollen möglichst viele geschlossen werden, ohne daß diese Bedingung verletzt wird.

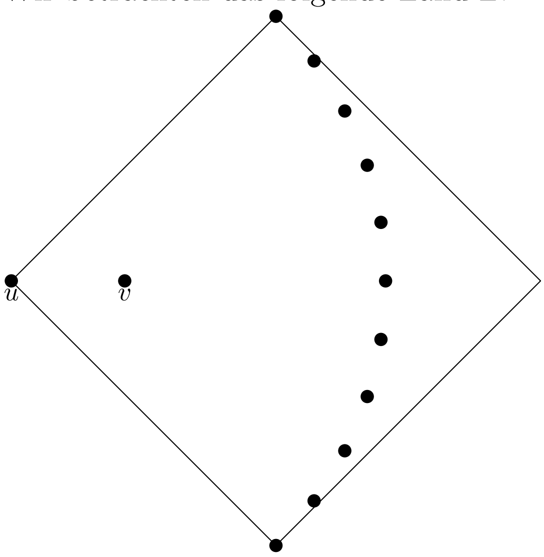
Ein Land ist hierbei der einfachheitshalber eine konvexe Teilmenge des $\mathbf{R}^2$.

Aus Zeitgründen wird nun einfach in einer beliebigen Reihenfolge geprüft, ob eine Universität überflüssig ist, um sie dann zu entfernen. Wie gut ist dieses Verfahren im Vergleich zu einer optimalen Lösung?

Ansatz: Offensichtlich liefert der Algorithmus eine $O(|U|)$ Approximation. Wir zeigen nun, daß er tatsächlich nicht besser ist.

Der folgende Beweis gibt eine sehr genaue untere Schranke für die Güte des Algorithmus an. Wir akzeptieren aber auch einfachere Lösungen, die zum Ergebnis kommen, daß der Algorithmus keine konstante Güte hat.

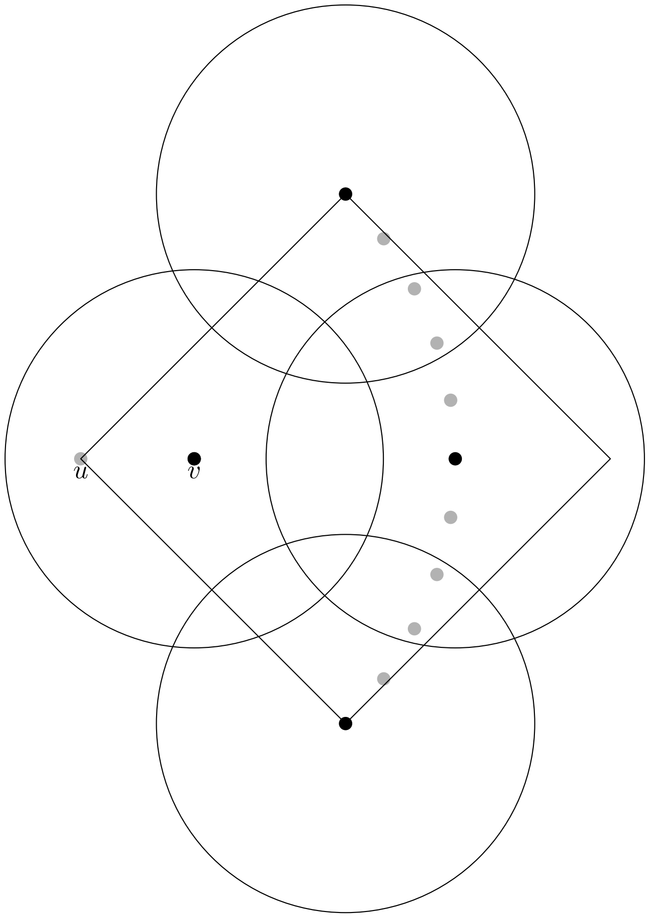
Wir betrachten das folgende Land L:



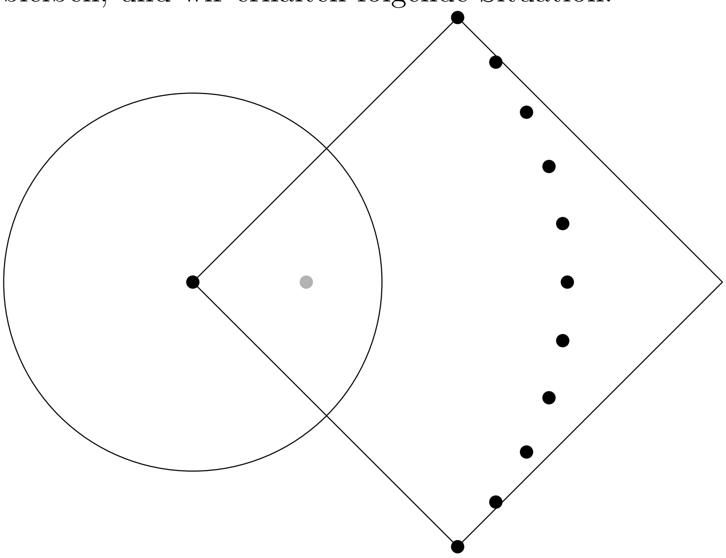
Der Abstand der Knoten ist dabei so gewählt, daß u und v genau $0.3d$ auseinanderliegen und der Abstand von u zu jedem Knoten auf der rechten Seite genau $(1+\alpha)d$ ist. Sei t die Anzahl der Knoten rechts und bezeichne R alle Knoten außer u, v (also $|R| = t$). Der Winkel zwischen zwei Knoten rechts ist genau $(t-1)/(0.5\pi)$. Die Menge aller Universitäten ist also $U := R \cup \{u, v\}$.

Sei $w \in R$ und $E(v) = \{u \in \mathbf{R^2} \mid dist(u, v) \leq d\}$ der Einzugsbereich eines Knotens (und analog für eine Menge von Knoten).

Eine gute Lösung wäre nun z.B. v und drei weiter Knoten zu behalten. Je nach genauer Positionierung der Knoten rechts reichen sogar zwei von ihnen:



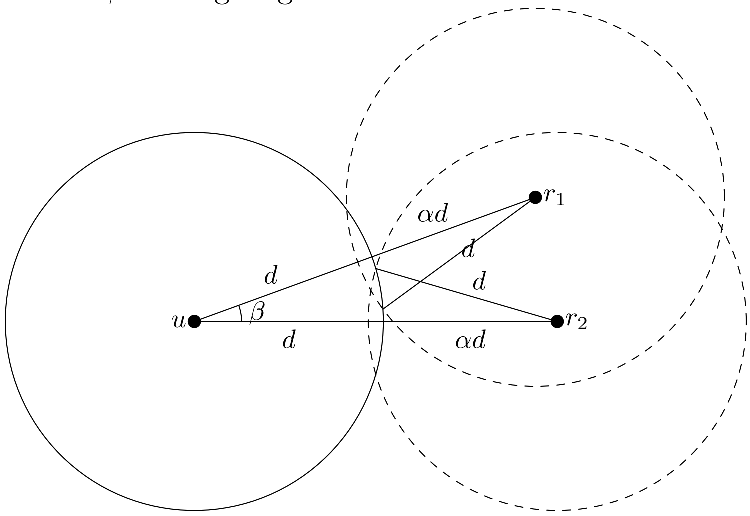
Der Greedyalgorithmus löscht aber möglicherweise v. Dann muss zwingend u bestehen bleiben, und wir erhalten folgende Situation:



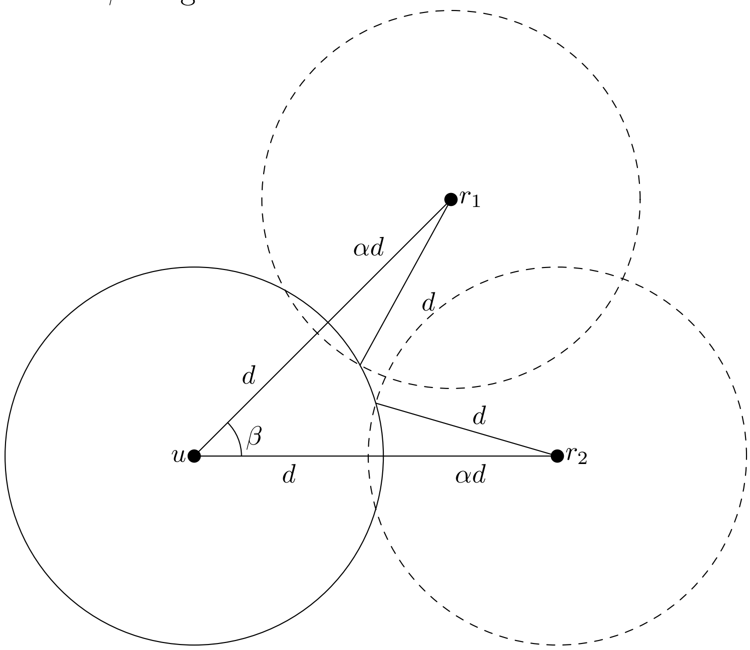
Wir zeigen nun, daß für jedes feste t ein α existiert, sodaß es notwendig ist, alle Knoten aus R zu behalten, falls v gelöscht wird. Daraus folgt dann, daß der Algorithmus keine feste Approximationsgüte liefert, sondern nur eine $O(U)$-Approximation.

Es ist leicht zu sehen, daß eine Teilmenge X von $R \cup \{u\}$ genau dann eine Lösung ist, wenn jeder Punkt des Landes mit Abstand d zu u auch von einem anderen Knoten $r \in X$ abgedeckt wird. Andernfalls entstehen Lücken. Formal also wenn für jeden Punkt $x \in L$ mit $dist(u, x) = d$ ein $r \in X$ existiert mit $dist(r, x) \leq d$. Dies ist genau dann der Fall, falls der Winkel β zwischen zwei benachbarten Knoten $r_1, r_2 \in X$ nicht zu groß ist (abhängig von α).

Winkel β klein genug:



Winkel β zu groß:



Mit etwas Geometrie kommt man leicht zu der Erkenntnis, daß sich für

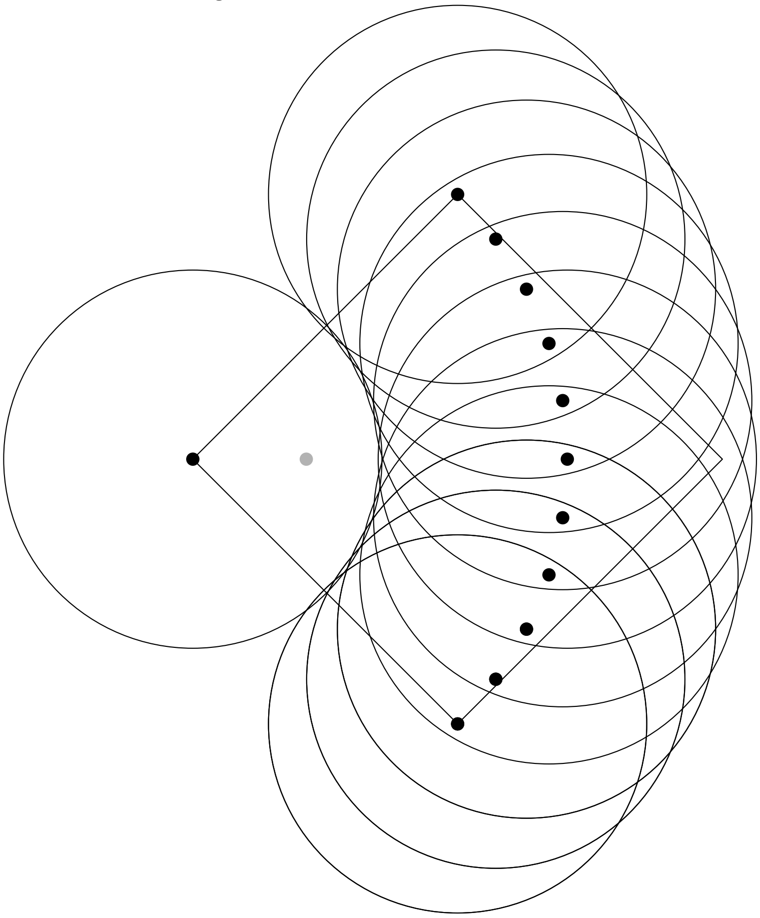
$$\beta \leq 2\cos^{-1}\left(\frac{1+\alpha}{2}\right)$$

beide Kreise um r_1 und r_2 auf dem Rand des Kreises um u überlappen. Dazu betrachte man die Mittelsenkrecht der Strecke u, r_1, deren Länge genau $\frac{1}{2}(1+\alpha)d$ ist. Diese geht durch den Schnittpunkt s der Kreise um u und r_1. Im entstehenden rechtwinkligen Dreieck kann man nun leicht alle Winkel berechnen. (Für $\alpha = 1$ ergibt sich übrigens $\beta \leq 0$).

Wählen wir also α derartig, daß

$$|R| = \frac{\pi}{2\cos^{-1}\left(\frac{1+\alpha}{2}\right)}$$

so ist nach dem Löschen von v kein weitere Knoten mehr entfernbar. Eine Lösung sieht dann also zwangsweise soaus:



Da

$$\frac{\pi}{2\cos^{-1}\left(\frac{1+\alpha}{2}\right)}$$

monoton und stetig mit α steigt ist eine solche Wahl möglich.
Wir erhalten somit die untere Schranke $\Omega(|U|)$ für die Approximationsgüte.