

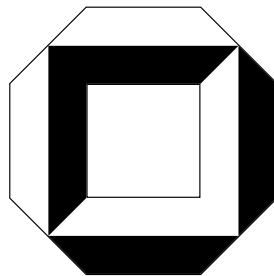
Landmark Selection and Greedy Landmark Descent Routing in Sensor Networks

Felix Brandt

Ausarbeitung zum Seminar
„Algorithmen für Ad-hoc- und Sensornetze“ im SoSe 2007

Institut für theoretische Informatik
Prof. Dr. Dorothea Wagner
Betreuer: Bastian Katz

25. September 2007



Universität Karlsruhe (TH)

Inhaltsverzeichnis

1	Einleitung	3
1.1	Gliederung	4
2	Landmarken	5
2.1	Landmarkenauswahl	6
2.2	r -Sampling Algorithmus	7
2.2.1	Kontinuierlicher Definitionsbereich	8
2.2.2	Diskreter Definitionsbereich	9
2.2.3	Das Protokoll	11
3	Routing mittels Landmarken	13
3.1	Die Greedy-Regel	14
3.1.1	Übertragungsqualität im stetigen Definitionsbereich	15
3.1.2	Obere Schranke der Pfadlänge im stetigen Definitionsbereich	16
3.2	Der Algorithmus	17
3.2.1	Adressaufbau	17
3.2.2	Greedy-Landmark-Descent-Routing	18
3.3	Randeffekte	18
3.4	Simulation	20
4	Zusammenfassung	21
	Literatur	22

1 Einleitung

Sensornetze stellen einen weiteren Schritt zur Allgegenwart der Informatik in unserer Umwelt dar. Möglich geworden durch immer kleinere, billigere und leistungsfähigere Hardware, eröffnen sie ganz neue Anwendungsbereiche, die sich allgemein als „Gebietsüberwachung“ zusammenfassen lassen. So können Sensornetze zur Erkennung, Aufzeichnung und Auswertung von Naturphänomenen wie Erdbeben, Vulkanaktivitäten oder Waldbränden eingesetzt werden. Vorstellbar wären auch Beobachtungen des Sozialverhaltens von Tieren, Netzwerke zwischen Fahrzeugen im Straßenverkehr sowie Anwendungen aus dem militärischen Umfeld.

Früher wurden Daten manuell gesammelt und zur Verarbeitung und Auswertung an leistungsstarke Hardware weitergereicht. Ein Sensornetz hingegen besteht aus vielen kleinen im Raum verteilten Sensorknoten und kann diese Aufgaben durch Bündelung der Ressourcen bereits im Netzwerk lösen und nach außen zum Beispiel in Form einer Datenbankschnittstelle anbieten. Die einzelnen Sensorknoten sind heutzutage etwa faust- bis münzgroße Computer, die nur aus Prozessor, Speicher, Kommunikationseinheit, Sensorik und einem Energiespeicher bestehen.

Die Verteilung der Knoten geschieht meist zufällig in der Untersuchungsregion. Nach dem Ausbringen müssen die Knoten selbständig ihre Kommunikationsinfrastruktur aufbauen und sich als Netzwerk organisieren. Die Koordinierung vieler kleiner, zum Teil mobiler Einheiten ist jedoch deutlich komplexer als eine feste Installation. Durch die schiere Anzahl der Knoten kann nicht jeder einzeln konfiguriert werden. Direkte Funkverbindungen sind nur zwischen „nahen“ Knoten möglich, um Kosten und Energie zu sparen. Zusätzlich können diese wenigen Verbindungen noch durch Hindernisse gestört werden oder durch Knotenbewegungen abbrechen. Ebenso können im Betrieb neue Verbindungen entstehen. Das Sensornetzwerk muss weiterhin fehlertolerant auf ausgefallene Knoten reagieren, da der Ausfall von Knoten spätestens durch die begrenzten Energieressourcen vorprogrammiert ist. Die Initialisierungsphase sollte kurz und der Betrieb energieeffizient gestaltet werden um die Knotenlaufzeit zu erhöhen.

Da die Knoten nur eine begrenzte Funkreichweite haben, sind keine direkten Verbindungen zu weit entfernten Knoten möglich. Die dazu nötige Weiterleitung von Paketen über andere Knoten erfordert Routingprotokolle, welche auf die speziellen Eigenheiten von Sensornetzen (begrenzte Ressourcen, Mobilität, Funkschatten im Gelände, Fehlertoleranz) ausgelegt sind. Die auf diesem Gebiet vorhandenen Routingalgorithmen lassen sich in zwei Gruppen einteilen: geographisches und nichtgeographisches Routing.

Geographische Routingalgorithmen orientieren sich weitestgehend am Koordinatensystem der Erde (oder einem eigenen Äquivalent) und versenden Pakete an einen bestimmten Ort, an dem der Zielknoten vermutet wird. Die Knoten verfügen dazu meist über einen *GPS*-Empfänger oder können ihre Position über Triangulation an lokalen Funkbaken errechnen. Nichtgeographische Routingverfahren setzen hingegen auf das Erstellen virtueller Koordinatensysteme, welche die Entfernungen der Knoten untereinander zur Positionsbestimmung benutzt.

1.1 Gliederung

Die vorliegende Ausarbeitung besteht aus zwei Teilen und beschreibt ein Verfahren zum nichtgeographischen Routing. Die vorgestellten Algorithmen wurden von An Ngyen, Nikola Milosavljevic, Qing Fang, Jie Gao und Leonidas J. Guibas an der Stanford und Stony Brook University erarbeitet und 2007 in [Ngu07] veröffentlicht.

Im ersten Teil werden Landmarken eingeführt, welche die Ankerpunkte für ein virtuelles Koordinatensystem darstellen. Es wird ein Protokoll vorgestellt, welches die selbständige Erzeugung und Verteilung der Landmarken im Sensornetz mit bestimmten garantierten Eigenschaften ermöglicht. Anschließend wird darauf ein virtuelles Koordinatensystem erstellt und eine Zuordnung von Adressen an die Knoten vorgestellt.

Im zweiten Teil dieser Arbeit wird aufbauend auf dem ersten Teil, ein auf Landmarken basierender Routingalgorithmus (*Greedy-Landmark-Descent-Routing*) vorgestellt. Dessen Qualität zeichnet sich, im Vergleich zu Vorgängeralgorithmen wie *BVR* oder *GLIDER*, weniger durch bessere Performance als durch einen wesentlich schonenderen Ressourcenverbrauch aus. Zum Abschluss werden noch einige Gütegarantien sowie Simulationsergebnisse dieses Routingansatzes vorgestellt.

2 Landmarken

Einleitung

In Ad-hoc- und Sensornetzen müssen beziehungsweise sollen Daten zielgerichtet transportiert werden, ohne dass direkt nach dem Ausbringen Informationen über die Struktur oder Ausdehnung des Netzes vorhanden sind. Kann jeder Knoten seine Position bestimmen, zum Beispiel über GPS oder lokale Funkbaken, lassen sich geographische Routingalgorithmen (siehe [?]) anwenden. Die Paketvermittlung erfolgt dann anhand der Koordinaten der Knoten, nicht anhand von Adressen, wie sie beispielsweise in hierarchischen Netzwerken oder im Internet Anwendung finden. Das Bestimmen der Koordinaten macht jeden Knoten jedoch unnötig teuer, vor allem wenn die Lokalisierung nur einmal durchgeführt wird und der Knoten danach am selben Ort verbleibt. Ohne die Möglichkeit der Ortsbestimmung einzelner Knoten gestaltet sich das Auffinden eines Zielknotens im Netz schon schwieriger.

Daher wurde in letzter Zeit auch an nichtgeographischen Routingalgorithmen (z.B. [Rao03]) geforscht. Diese errichten virtuelle Koordinatensysteme und beschreiben Regeln, wie die Daten durch diese Koordinatensysteme in Richtung Ziel gelangen können. Als Entfernungsmaß zwischen virtuellen Koordinaten dient dabei der euklidische Abstand oder die minimale Anzahl der Sprünge zwischen Knoten.

Eine Möglichkeit, ein virtuelles Koordinatensystem zu erzeugen, ist die Verwendung von Landmarken. Landmarken sind normale Knoten im Sensornetz, die jedoch beim

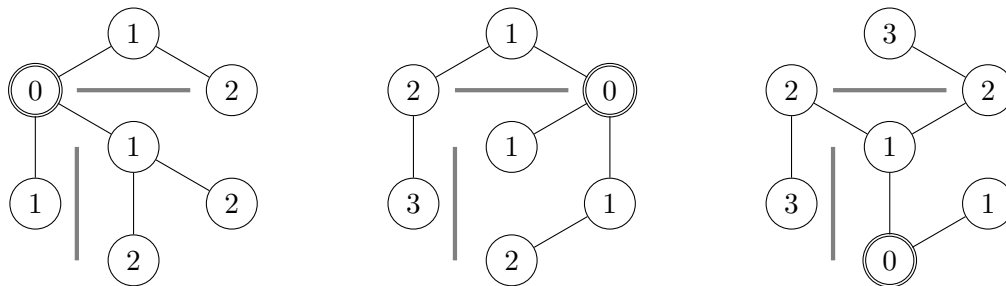


Abbildung 2.1: Entstehung der Knotenkoordinaten aus Landmarkenentfernungen. Die Landmarken sind in ihrem *Shortest-Path-Tree* jeweils doppelt umrandet. Dicke entsprechen Hindernissen, durch die kein Funkkontakt möglich ist.

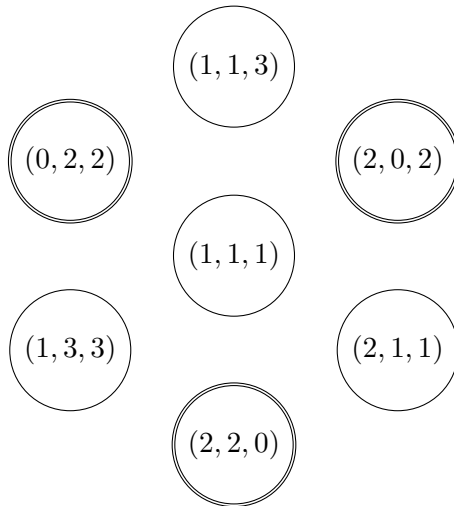


Abbildung 2.2: Knotenkoordinaten in Netzwerk

Aufbau des Koordinatensystems als Referenzknoten fungieren. Im Betrieb ermittelt jeder Knoten seine minimale Entfernung (Anzahl der Sprünge) zu jeder (oder einer bestimmten Menge) von Landmarken. Dadurch entsteht, wie in Abbildung 2.1 zu sehen, für jede Landmarke ein *Shortest-Path-Tree* im Sensornetzwerk. Der Vektor der Entfernungen zu den Landmarken bildet die Adresse des Knotens (siehe Abbildung 2.2). Das Routing besteht nun darin, den Adressvektor des Senderknotens an den des Empfängerknotens gierig anzunähern. Ein Knoten schickt das Paket also an denjenigen seiner direkten Kommunikationsnachbarn weiter, der im Sinne des verwendeten Routingalgorithmus am nächsten am Zielknoten ist. Nach dem Ausbringen eines Sensornetzes muss jedoch erst einmal die Routinginfrastruktur aufgebaut werden, bevor mit dem Routing begonnen werden kann.

2.1 Landmarkenauswahl

Ein bisher wissenschaftlich wenig bearbeitetes Problem ist die Auswahl der zum Routing verwendeten Landmarken. Dabei hat die Wahl einen nicht unbedeutenden Einfluss auf die Routing-Performance im gesamten Netzwerk. Sind die vorhandenen Landmarken schlecht verteilt, so lassen sich ganze Netzwerksegmente gar nicht erreichen oder es existieren viele Knoten mit der gleichen Adresse.

Das Ziel muss also sein, die Landmarken gleichmäßig im Netz zu verteilen und sicherzustellen, dass jeder Knoten von mehreren Landmarken umgeben ist. In dem im zweiten Teil dieser Arbeit vorgestellten Routingalgorithmus sollen die Pakete immer entlang der *Shortest-Path-Trees* in Richtung der jeweiligen Wurzel-Landmarke geschickt werden. Es ist also von Vorteil in der Nähe jedes Knotens mindestens eine Landmarke zu haben, um durch dieses „abfließen“ möglichst nah an das Ziel zu kommen. Auf der anderen Seite

sollten nicht zu viele Landmarken erzeugt werden, da sonst unnötig viele Ressourcen für das Routing verschwendet werden.

Es ergeben sich damit folgende Ziele für die Landmarkenauswahl:

- möglichst wenige Landmarken
- jeder Knoten soll „nahe“ Landmarken haben
- gleichmäßige Verteilung der Landmarken in Netzwerk
- schnelle Landmarkenauswahl

In [Ngu07] wurde ein fortgeschrittener Algorithmus zur Landmarkenauswahl präsentiert, auf den ich im Folgenden genauer eingehen werde. Zunächst sollen jedoch erst einmal die bisher bekannten Vorgehen zur Landmarkenauswahl kurz erläutert werden.

Bekannte Methoden zur Landmarkenauswahl

Manuelle Auswahl: Die manuelle Auswahl der Landmarken (wie z.B. in [Rao03]) bietet den Vorteil, dass Wissen über die Topologie und zu erwartende Netzwerkkonnektivität mit berücksichtigt werden kann. So können speziell Randknoten zu Landmarken gemacht werden (ein Fakt, der später noch wichtig werden wird). Oder es ist möglich, Landmarken gleichmäßig und sparsam einzusetzen. Für große Ad-hoc- und Sensornetze disqualifiziert sich dieses Verfahren durch den immens hohen Aufwand jedoch von selbst.

Drei-Anker-Knoten: Ein anderer interessanter Ansatz wurde in [Car05] vorgestellt. Hier werden unabhängig von der Ausdehnung des Netzwerkes drei möglichst weit voneinander entfernte Landmarken bestimmt. Die durch die geringe Anzahl an Landmarken entstehenden Adressen sind dabei nicht eindeutig und es lassen sich nur „Zonen“ von Knoten mit gleicher Adresse festlegen. Für Netzwerke komplexerer Topologie, bei denen die Zonen gleicher Adressen beispielsweise nicht zusammenhängend sind, reicht diese Art der Adressierung jedoch nicht mehr aus.

Zufällige Auswahl: Das einfachste Verfahren zum Bestimmen der Landmarken ist wohl deren zufällige Auswahl. Dazu bestimmt jeder Knoten einen zufälligen Wert $x_i \in [0, 1)$. Liegt der Wert des i . Knoten oberhalb eines Netzwerkparameters K wird der i . Knoten zu einer Landmarke. Die Anzahl der erzeugten Landmarken lässt sich über K a-priori festlegen. Weiterhin ist die Auswahl sehr schnell erledigt, liefert aber keinerlei Garantie über die Verteilung der Landmarken.

2.2 r -Sampling Algorithmus

Die bisher vorgestellten Möglichkeiten der Landmarkenauswahl bieten keinerlei Garantien für die Qualität der erzeugten Menge an Landmarken. In diesem Abschnitt soll

gezeigt werden, dass ein sogenanntes r -Sampling des Netzes eine „gute“ Menge Landmarken mit garantierter Güte liefert. Aus den Eigenschaften des r -Samplings geht außerdem hervor, dass der im zweiten Teil dieser Arbeit vorgestellte Routingalgorithmus *Greedy-Landmark-Descent-Routing* funktioniert. Darüber hinaus lassen sich sogar theoretische Aussagen zur maximalen Pfadverlängerung treffen, die auf Grund der Ungenauigkeiten des virtuellen Koordinatensystems zu erwarten sind.

Zuerst soll der allgemeinere Fall im kontinuierlichen Definitionsbereich betrachtet werden. Anschließend können die Erkenntnisse fast analog auf den diskreten Definitionsbereich der Sensorknoten übertragen werden.

2.2.1 Kontinuierlicher Definitionsbereich

Im Folgenden bezeichne $B(p, r)$ eine Kugel vom Radius r um den Punkt p .

Definition 1 (r -Sampling) *Das r -Sampling eines Gebietes S entspricht einer Punktmenge $R \subset S$, sodass sich jeder Punkt des Gebietes S innerhalb eines Abstandes r von einem beliebigen Punkt aus R befindet. Es gilt also insbesondere:*

$$S \subseteq \bigcup_{p \in R} B(p, r)$$

Ist die Menge R ein gültiges r -Sampling von S und wählen wir R als die Menge unserer Landmarken, ist mindestens die Bedingung der „nahen“ Landmarke an jedem Punkt erfüllt. Das r -Sampling garantiert außerdem, dass alle Punkte, welche nicht in der Nähe des Randes liegen von einer gewissen Menge Landmarken umgeben sind. Das führt zur nächsten Definition.

Definition 2 (Innerer Punkt) *Jeder Punkt P , dessen $4r$ -Kugel komplett in S liegt ist ein innerer Punkt.*

$$B(P_{\text{innen}}, 4r) \subseteq S$$

Unser Ziel ist es nun, ein r -Sampling mit möglichst wenig Landmarken zu erhalten. Auf einer unendlichen Ebene ist das bestmögliche Ergebnis offensichtlich ein sechseckiges Sampling-Muster (siehe Abbildung 2.3). Dieses kann durch ein globales sequentielles Vorgehen recht einfach erhalten werden.

Da die Vorteile der zufälligen Landmarkenauswahl bestechend sind, soll nun noch untersucht werden, inwieweit sich damit ein r -Sampling erzeugen lässt. Wird über das Gebiet ein Gitter mit quadratischen Zellen der Größe $\Theta(r^2)$ gelegt und befindet sich in jeder dieser Zellen mindestens eine Landmarke, so handelt es sich offensichtlich um ein r -Sampling. Dieses Problem entspricht dem bekannten „Coupon Collector’s Problem“. Bei einer Aufteilung des Gebietes auf m Gitterzellen liegt der Erwartungswert also bei $m(\lg m + O(1))$ Landmarken. Bemerkenswert ist hierbei, dass die Menge der zu viel erzeugten Landmarken lediglich um einen logarithmischen Faktor mit der Anzahl der Gitterzellen wächst.

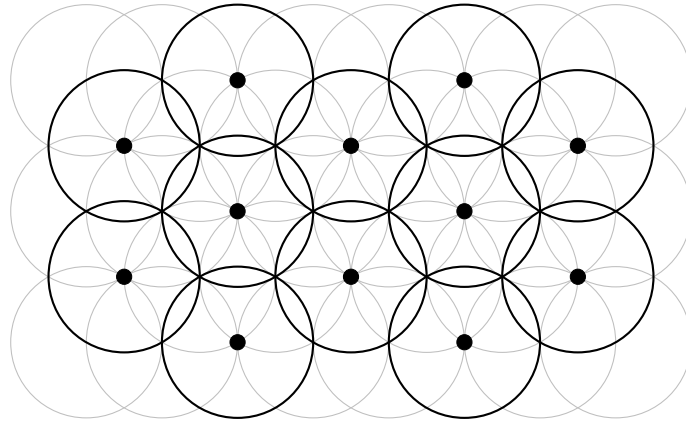


Abbildung 2.3: Ideales Samplingmuster auf unendlicher Ebene

2.2.2 Diskreter Definitionsbereich

Die Definitionen des r -Sampling und der inneren Punkte können fast analog in den diskreten Bereich übernommen werden. Das Entfernungsmaß zwischen den Punkten wird lediglich durch den kürzesten Pfad (die minimale Anzahl der Sprünge zwischen den Knoten) ersetzt.

Definition 3 (r -Umgebung) Die r -Umgebung $B(c, r)$ eines Knotens c entspricht der Menge aller Knoten, die über maximal r Sprünge von c aus erreichbar sind.

Im Folgenden soll angenommen werden, dass die Knoten relativ gleichmäßig verteilt sind und das Netz eine gewisse Dichte nicht unterschreitet. Insbesondere befinden sich damit in jeder r -Umgebung eines Knotens mindestens αr^2 andere Knoten, wobei $\alpha > 0$ eine Netzwerkkonstante ist.

Um nun ein gültiges r -Sampling zu erhalten, muss in der r -Umgebung jedes Knotens mindestens eine Landmarke existieren. Der folgende Greedy-Algorithmus leistet genau dies.

Algorithmus 1 (Sequentieller Algorithmus) Zu Beginn ist kein Knoten markiert. Solange es unmarkierte Knoten gibt, wähle einen Knoten aus, mache ihn zur Landmarke und markiere alle Knoten innerhalb seiner r -Umgebung.

Durch diesen Algorithmus erhält man eine Menge von Landmarken die jeweils mindestens r Sprünge voneinander entfernt sind. Dadurch sind alle $\frac{r}{2}$ -Umgebungen paarweise disjunkt. Der Algorithmus erzeugt also maximal $4n/(\alpha r^2)$ Landmarken.

$$\frac{n}{\alpha(\frac{r}{2})^2} = \frac{4n}{\alpha r^2}$$

Dieses Vorgehen verbietet sich aber in Ad-hoc- und Sensornetzen von selbst, da die Laufzeit mit der Größe des Netzes wächst. Da im Netzwerk nur sehr wenig Aktivität

herrscht, bietet es sich an die Landmarkenauswahl zu parallelisieren. Ein paralleler Algorithmus bietet hier deutliche Vorteile, allen voran die bessere Skalierbarkeit.

Algorithmus 2 (Randomisierter paralleler Algorithmus) *Mache aus jedem Knoten mit Wahrscheinlichkeit p eine Landmarke.*

Dabei ist p ein Netzwerkparameter über den die Anzahl der Landmarken eingestellt und damit das Übersampling beeinflusst werden kann. Von p hängt außerdem ab, mit welcher Sicherheit der Algorithmus überhaupt ein r -Sampling produziert.

Satz 1 *Algorithmus 2 erzeugt mit Wahrscheinlichkeit s ein gültiges r -Sampling im Netzwerk, genau dann wenn*

$$p \geq 1 - \left(1 - s \frac{\alpha r^2}{16n}\right)^{\frac{4}{\alpha r^2}}$$

Beweis: Um wie bei Algorithmus 1 die Eigenschaft disjunkter Umgebungen nutzen zu können, verwenden wir statt der r -Umgebung nur die $\frac{r}{2}$ -Umgebung und bestimmen zuerst mittels Algorithmus 1 eine Menge Knoten die wir im Folgenden Zentren nennen. Die obige Betrachtung der maximal möglichen Landmarkenanzahl sagt uns, dass maximal $m = \frac{16n}{\alpha r^2}$ solche Zentren zu erwarten sind.

Befindet sich in einem Zentrum eine Landmarke, so ist kein Punkt der $\frac{r}{2}$ -Umgebung des Zentrums weiter als r von dieser Landmarke entfernt. Unter der Bedingung, dass sich in der $\frac{r}{2}$ -Umgebung jedes Zentrums mindestens eine Landmarke befindet, liefert Algorithmus 2 also ein gültiges r -Sampling. Sei ε_i nun der Fall, dass in der r -Umgebung von Zentrum i keine Landmarke liegt. Da in dieser $\frac{r}{2}$ -Umgebung mindestens $\frac{\alpha r^2}{4}$ Knoten liegen, ergibt sich als Wahrscheinlichkeit:

$$Pr[\varepsilon_i] \leq (1 - p)^{\frac{\alpha r^2}{4}}$$

Liegt nun in der $\frac{r}{2}$ -Umgebung jedes Zentrums eine mindestens Landmarke, so erzeugt Algorithmus 2 ein gültiges r -Sampling. Das führt zu der Wahrscheinlichkeit:

$$Pr\left[\bigcap_{i=1}^m (\neg\varepsilon_i)\right] \geq \left(1 - (1 - p)^{\frac{\alpha r^2}{4}}\right)^{\frac{16n}{\alpha r^2}}$$

$$\left(1 - (1 - p)^{\frac{\alpha r^2}{4}}\right)^{\frac{16n}{\alpha r^2}} \geq s$$

$$(1 - p)^{\frac{\alpha r^2}{4}} \leq 1 - s \frac{\alpha r^2}{16n}$$

$$p \geq 1 - \left(1 - s \frac{\alpha r^2}{16n}\right)^{\frac{4}{\alpha r^2}}$$

Da Algorithmus 2 nicht ausschließt, dass Landmarken näher als r Schritte voneinander entfernt existieren, sind auch hier mehr Landmarken als bei Algorithmus 1 zu erwarten.

Während der Algorithmus 1 maximal $4n/\alpha r^2$ Landmarken erzeugt, produziert Algorithmus 2 aller Voraussicht nach $n \cdot p$ Landmarken. Das Verhältnis der zu viel erzeugten Landmarken setzt sich damit wie folgt zusammen:

$$\begin{aligned} \frac{LM_{par}}{LM_{seq}} &\geq \frac{n \cdot \left[1 - \left(1 - s^{\frac{\alpha r^2}{16n}} \right)^{\frac{4}{\alpha r^2}} \right]}{\frac{4n}{\alpha r^2}} \approx \frac{\alpha r^2}{4} \cdot \left[1 - \left(\frac{1-s}{\frac{16n}{\alpha r^2}} \right)^{\frac{4}{\alpha r^2}} \right] \\ &= k \cdot \left[1 - \left(\frac{1-s}{4m} \right)^{\frac{1}{k}} \right] \end{aligned}$$

Dabei entspricht k der minimalen Anzahl an Knoten in jeder $\frac{r}{2}$ -Umgebung, $k = \frac{\alpha r^2}{4}$.

Es wurden nun zwei verschiedene Algorithmen zur Landmarkenauswahl präsentiert, die jeder für sich betrachtet deutliche Nachteile aufweisen. Die durch den sequentiellen Algorithmus erzeugte Menge Landmarken hat eine garantierte Güte (Verteilung, Anzahl), der Algorithmus ist jedoch zu langsam. Der schnellere, randomisierte Algorithmus hingegen erzeugt eine Landmarkenmenge ohne jede Gütegarantie. Durch Kombination beider Algorithmen soll im Folgenden ein Verfahren präsentiert werden, welches die Landmarkenauswahl in annehmbarer Zeit und mit gewissen Gütegarantien realisiert.

2.2.3 Das Protokoll

Aufbauend auf den eben vorgestellten Ergebnissen wird in [Ngu07] ein randomisiertes, verteiltes Protokoll präsentiert. Dieses zeichnet sich insbesondere durch folgende Eigenschaften aus:

- konstante Laufzeit $O(1)$
- garantierte Güte (gleichmäßige Verteilung der Landmarken)
- obere Schranke für die Landmarkenanzahl

Der Algorithmus ist eine Kombination der Algorithmen 1 und 2 und er vereint insbesondere deren Vorteile. Er verfügt über zwei Parameter, mit denen die Landmarkenauswahl beeinflusst werden kann. Der Parameter t entspricht der Zeit, welche ein Paket maximal benötigt, um von einem Knoten zu einem direkten Nachbarn verschickt zu werden (\approx Sendereichweite). Mit dem zweiten Parameter K , kann der Algorithmus hinsichtlich schnellerer Laufzeit oder weniger unnötigen Landmarken beeinflusst werden.

Jeder Knoten beginnt die Netzwerkinitialisierungsphase mit der zufälligen Wahl eines Wertes $v \in [0, 1)$. Danach wartet er Kvt Sekunden lang. Wurde der Knoten in dieser Zeit von keiner anderen Landmarke „unterdrückt“, so wird er nun selbst zur Landmarke. Anschließend schickt er per Broadcast an alle Knoten seiner r -Umgebung eine „Unterdrückungsnachricht“, welche den Algorithmus bei all diesen Knoten beendet.

Für große K entspricht dieses Vorgehen dem sequentiellen Algorithmus 1. Je größer K , desto länger dauert auch die Landmarkenauswahl. Je kleiner K gewählt wird, desto öfter wird ein Knoten zu einer Landmarke gemacht, obwohl er bereits in der r -Umgebung einer anderen Landmarke liegt und eine „Unterdrückungsnachricht“ zu ihm unterwegs ist.

Definition 4 ((α, β)-Regularität) *Eine Verteilung der Sensorknoten im Netzwerk ist (α, β)-regulär, wenn in jeder r -Umgebung jedes Knotens mindestens αr^2 und maximal βr^2 Knoten vorhanden sind.*

$$\forall c \in S \forall r \in \mathbb{R} : \alpha r^2 \leq |B(c, \lfloor r \rfloor)| \leq \beta r^2$$

Satz 2 *Ist die Verteilung der Knoten (α, β)-regulär, existieren in jeder r -Umgebung maximal*

$$\frac{9\beta r^2}{K} + 16\beta - 1$$

Landmarken.

Zum Beweis sei nur soviel gesagt: Es wird gezeigt, dass die Landmarken aufgrund der unterschiedlich langen Wartezeiten mit einer gewissen Wahrscheinlichkeit bestimmte Mindestabstände haben und sich daraus eine Obergrenze der Landmarkenanzahl in der Umgebung eines Knoten herleiten lässt. Der Beweis im Anhang von [Ngu07] sowie das genaue Ergebnis von Satz 2 ist nach Aussage eines der Autoren auch nicht ganz korrekt. Es besteht jedoch kein Zweifel, dass sich die Anzahl mit einem ähnlichen Ergebnis nach oben beschränken lässt.

Durch geeignete Wahl des Parameters K kann nun versucht werden, das Optimum zwischen beiden Algorithmen zu finden. Für $K = \Theta(r)$ steigt die Anzahl zu viel erzeugter Landmarken linear mit r . Wird K stattdessen aus $\Theta(r^2)$ gewählt, kann eine konstante, von der Größe der r -Umgebung unabhängige Anzahl an zu viel erzeugten Landmarken realisiert werden.

3 Routing mittels Landmarken

Einführung

Eine wesentliche Anforderung an Ad-hoc- und Sensornetzwerke ist, neben der Generierung von Daten, auch deren Transport. In einem einfachen Szenario schicken alle Knoten regelmäßig ihren Status (Temperatur, Heart-Beat, Gaskonzentration oder sonstige Messwerte) an einen Gateway-Knoten (Senke) der die Daten aggregiert und weiterverarbeitet. Betrachtet man die Landmarken aus dem ersten Teil und deren *Shortest-Path-Trees*, fällt auf, dass in diesem speziellen Fall sogar eine einzige Landmarke (die Senke) reicht, damit alle Pakete zu dieser „abfließen“ können. In zukünftigen Anwendungen von Ad-hoc- und Sensornetzen sollen jedoch auch Abfragen, wie sie bei Datenbanken der Fall sind, angewendet werden können. Dazu muss es möglich sein, zwischen beliebigen Knoten im Netzwerk Daten zu verschicken, unabhängig von den Positionen der Landmarken.

Bei geographischem Routing kennt jeder Knoten seine Position mehr oder weniger genau. Pakete können nun jeweils an den Knoten weitergeleitet werden, der eine Potentialfunktion (etwa den euklidischen Abstand zum Ziel) minimiert. In dieser Ausarbeitung soll allerdings nur nichtgeographisches Routing betrachtet werden, welches auf den bereits vorgestellten Landmarken basiert. Die Hauptidee landmarkenbasierten Routings ist es, die Landmarken in der Nähe des Zielknotens zu identifizieren und die Pakete über deren *Shortest-Path-Trees* in Richtung des Zielknotens zu bewegen.

Die Anforderungen an landmarkenbasiertes Routing sind dabei die gleichen wie für jeden Routingalgorithmus in Ad-hoc- und Sensornetzwerken:

- garantierte Übertragung der Pakete
- kurze Wege (wenig Abweichung vom direkten Pfad)
- geringer Speicherverbrauch
- gleichmäßiger Energieverbrauch (der einzelnen Knoten)
- möglichst selten Fluten

Bekannt Methoden zum Routing mittels Landmarken

GLIDER In [Fan05] kennt jeder Knoten seine „nächste“ Landmarke. Global betrachtet induziert diese Beziehung ein Voronoi-Diagramm auf dem Netzwerk, bei dem alle zu

einer Landmarke gehörenden Knoten ein Gebiet bilden. Das Routing wird nun in zwei Teilschritte eingeteilt:

1. **globales Routing:** Routing zum Zielgebiet
2. **lokales Routing:** Knoten im Zielgebiet finden

Um ein Paket zum Zielgebiet zu bringen, wird in der Initialisierungsphase ein Atlas über alle Gebiete (inkl. Nachbarschaftsbeziehungen) erstellt und der kürzeste Weg jeder Gebietskombination vorberechnet. Dieser Atlas muss allen Knoten zur Verfügung gestellt werden. Zum lokalen Routing wird der Wert einer Potentialfunktion minimiert, welche die Abstände zu den Landmarken der angrenzenden Voronoi-Gebiete betrachtet.

Beacon Vector Routing (BVR) Beim *BVR* wird der lokale Teil aus *GLIDER* auf das gesamte Routing angewandt. Für jeden Routingschritt werden die k nächsten Landmarken des Zielknotens betrachtet. Landmarken, die näher am Ziel als am aktuellen Knoten sind, erzeugen in der Potentialfunktion eine „anziehende“ Kraft. Ist eine Landmarke weiter vom Ziel entfernt als vom aktuellen Knoten, entspricht dies einer „abstoßenden“ Kraft.

Bei diesem Ansatz muss jeder Knoten die Entfernungen aller seiner Nachbarn zu allen Landmarken kennen. Diese werden benötigt, um jeweils den dem Ziel nächsten Knoten für den kommenden Routingschritt zu ermitteln. Je dichter das Netz, desto mehr Knotenentfernungen müssen also bekannt sein. Außerdem sind in jedem Routingschritt umfangreiche Berechnungen durchzuführen um den nächsten Knoten auszuwählen. Das Verfahren funktioniert zwar bereits sehr gut, ist aber keineswegs speicher- und berechnungsparend.

3.1 Die Greedy-Regel

Die in [Ngu07] vorgestellte zentrale Routingregel des *Greedy-Landmark-Descent-Routing* lautet wie folgt:

Leite Pakete immer in Richtung der Landmarke weiter, welche das Verhältnis des Abstandes Start-Landmarke zum Abstand Ziel-Landmarke maximiert.

Ein Paket wird nun entlang des *Shortest-Path-Trees* der aktuellen Landmarke, in Richtung dieser geschickt. Dies ist rein mechanisch und ohne größeren Aufwand, im Gegensatz zum *BVR*, möglich. Im Speicher eines Knotens müssen nun nur noch die Landmarkenentfernungen, sowie der Knoten welcher sich „abwärts“ im *Shortest-Path-Tree* jeder Landmarke befindet, gehalten werde.

Definition 5 (Extremknoten) *Ein Knoten, der von der aktuell betrachteten Landmarke genau so weit entfernt ist wie der Zielknoten heißt Extremknoten.*

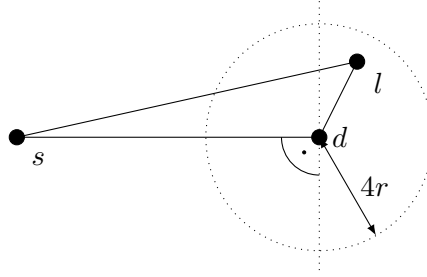


Abbildung 3.1: Ist das Ziel ein innerer Punkt, findet sich immer eine Landmarke, die eine anziehende Kraft ausüben kann (Abbildung aus [Ngu07]).

Jedes Paket wird nun so lange in Richtung der aktuellen Landmarke weitergeleitet, bis ein Extremknoten erreicht wird. Somit sind auf keinem Transitknoten aufwändige, energiehungrige Berechnungen nötig. An jedem Extremknoten beginnt der Algorithmus wieder von vorn. Dabei werden die letzten acht Extrempunkte im Paket gespeichert, um eventuelle Kreisläufe zu erkennen. Trifft ein Paket an einem bekannten Extremknoten ein, so ist die *Greedy-Regel* fehlgeschlagen. Das Paket wird nun an denjenigen Nachbarn weitergeleitet, der bezüglich der L_1 -Norm dem Ziel am nächsten ist.

$$\text{Min}_i (|v_i - v_t|)$$

Funktioniert das ebenfalls nicht, wird die L_∞ -Norm verwendet.

$$\text{Min}_i |v_i - v_t|_\infty$$

Kann das Paket dadurch immer noch nicht weiter vermittelt werden, wird das Netzwerk in Richtung des Zielknotens mit der Nachricht geflutet.

3.1.1 Übertragungsqualität im stetigen Definitionsbereich

Lemma 3 Sei s die Quelle eines Paketes, d ein innerer Punkt und $s \neq d$, dann existiert eine Landmarke l für die gilt $|sl| > |dl|$.

Beweis: Wir nehmen an, dass die Landmarken ein r -Sampling bilden und betrachten die vom Startknoten abgewandte $4r$ -Halbkugel um den Zielknoten. In dieser Halbkugel befindet sich mindestens eine Landmarke. Für jede Landmarke l in dieser Halbkugel gilt, wie in Abbildung 3.1 zu erkennen, dass $\angle sdl > 90^\circ > \angle dsl$ und damit auch $|sl| > |dl|$. ■

Es existiert also stets eine „anziehende“ Landmarke, wenn der Zielpunkt ein innerer Punkt ist. Somit kann auch immer in Richtung dieser Landmarke navigiert werden. Im folgenden Abschnitt soll gezeigt werden, dass eine Weiterleitung in Richtung dieser Landmarke das Paket auch tatsächlich näher an seinen Zielpunkt heranbringt.

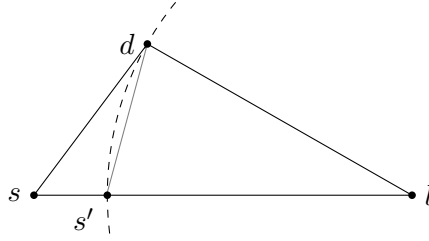


Abbildung 3.2: Landmark-Descent bringt das Paket mit jedem Schritt näher an das Ziel (Abbildung aus [Ngu07]).

Lemma 4 Sei $|sl| > |dl|$, dann kommt das Paket im nächsten Schritt (zu s') näher an das Ziel heran und es gilt $|s'd| < |sd|$.

Beweis: Von s wird das Paket in Richtung l weitergeleitet bis $|s'l| = |dl|$ und damit $\triangle s'ld$ ein gleichschenkliges Dreieck ist. Insbesondere gilt $\angle ds'l < 90$ und auch $\angle ss'd > 90$. Daraus folgt, dass $\angle dss' < 90$ und damit auch $|s'd| < |sd|$. ■

Die gemeinsame Betrachtung der Lemma 3 und 4 zeigt, dass jedes Paket durch *Greedy-Landmark-Descent-Routing* letztendlich zu seinem Zielpunkt findet. Die beiden Ergebnisse können also zu folgendem Satz zusammengefasst werden:

Satz 5 Ist das Ziel d ein innerer Punkt, so bringt *Greedy-Landmark-Descent-Routing* jedes Paket ans Ziel.

3.1.2 Obere Schranke der Pfadlänge im stetigen Definitionsbereich

Es soll nun theoretisch betrachtet werden um welche Größenordnung sich der durch *Greedy-Landmark-Descent-Routing* gefundene Weg vom kürzestmöglichen Weg unterscheidet. Ziel ist die Angabe einer oberen Schranke für die maximale Pfadverlängerung.

Definition 6 (Ein-Schritt-Dilatation) Das Verhältnis zwischen der zurückgelegten Strecke und der realen Annäherung an das Ziel in einem Routingschritt ist die Ein-Schritt-Dilatation κ .

$$\kappa = \frac{|ss'|}{|sd| - |s'd|}$$

Im folgenden Lemma soll κ zum minimalen Abstand von einem Punkt d und der ihn adressierenden Landmarken ins Verhältnis gesetzt werden.

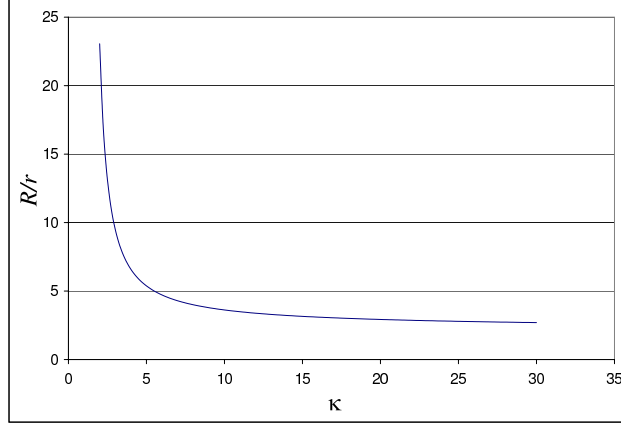


Abbildung 3.3: obere Schranke für R/r bei verschiedenen κ (Abbildung aus [Ngu07]).

Lemma 6 Gegeben seien ein gültiges r -Sampling und der Abstand zwischen einem Knoten und der ihn adressierenden Landmarken sei mindestens R . Dann gilt $f(\kappa) \geq \frac{R}{r}$, wobei

$$f = \frac{4}{1 + u - \sqrt{w}}, \quad u = 1 - \frac{1}{\kappa^2}, \quad v = 1 - \frac{1}{\kappa}$$

$$w = (1 - u^2) + 2u^2(1 - v^2) + 2u\sqrt{(1 - v^2)(1 - u^2v^2)}$$

Offensichtlich ist $\kappa > 1$ und $f(\kappa)$ für $\kappa > 1$ streng monoton fallend mit Wertebereich $(2, \infty)$. Die Umkehrfunktion f^{-1} existiert also und für $R > 2r$ gilt $\kappa \leq f^{-1}(R/r)$. Je kürzer der Routingpfad also sein soll, desto weiter müssen die adressierenden Landmarken vom Zielpunkt entfernt sein. Dieser Beweis ergibt folgenden Satz:

Satz 7 Für $R > 2r$ besitzt κ eine obere Schranke.

Unter Annahme der (α, β) -Regularität kann nun die Anzahl der adressierenden Landmarken nach oben beschränkt werden. Dieses Ergebnis ist wichtig für die Skalierbarkeit des Systems.

Satz 8 Ist die Knotenverteilung (α, β) -regulär, so beträgt die erwartete Anzahl adressierender Landmarken eines Knotens maximal $144\frac{\beta r^2}{K} + 16\beta - 1$.

$$144\frac{\beta r^2}{K} + 16\beta - 1 = 9\frac{\beta(4r)^2}{K} + 16\beta - 1$$

3.2 Der Algorithmus

3.2.1 Adressaufbau

Jeder Knoten verwendet als Adresse den Vektor der Entfernungen zu den m nächsten Landmarken. In [Ngu07] wurde durch Simulationen gezeigt, dass m recht klein sein kann.

Sechs Landmarken reichen bereits aus, um mit hoher Wahrscheinlichkeit die Pakete erfolgreich auszuliefern. Wird der Adressvektor größer gewählt, kann die Richtung des zu routenden Paketes genauer bestimmt werden. Allerdings verbraucht das Routing weniger Ressourcen (etwa durch kleinere Paketheader), wenn der Adressvektor aus weniger Landmarkenentfernungen besteht.

3.2.2 Greedy–Landmark–Descent–Routing

Die Kernidee dieses Routingalgorithmus besteht darin, die Landmarke mit der höchsten „Anziehungskraft“ auszuwählen und Pakete absteigend über deren *Shortest-Path-Tree* ihrem Ziel anzunähern. Zuerst soll hier zusammengetragen werden, welche Daten jedes Paket während des Routings mit sich führen muss.

Im Allgemeinen kann nicht verhindert werden, dass mehrere Knoten über die gleichen Entfernungen zu allen Landmarken verfügen. Deswegen muss die Adresse des Zielknotens außer den Entfernungen des Zielknotens zu seinen nächsten Landmarken auch die ID des Zielknotens enthalten. Um die Paketheader klein zu halten, soll außerdem nicht jeder Extremknoten gespeichert werden. In [Ngu07] reichten bereits $e = 8$ letzte Extremknoten aus, um zuverlässig lokale Minima oder Kreisläufe zu erkennen.

Der Paketheader jeder Nachricht sieht dann wie folgt aus:

- ID des Zielknotens
- Adresse des Zielknotens
- aktuelle Ziellandmarke
- Liste der e letzten Extremknoten
- Nutzlast

Außer an Extremknoten lassen sich die Pakete quasi mechanisch über die *Shortest-Path-Trees* weiterleiten. Pakete müssen also nur an Extrempunkten bearbeitet werden (Wahl der neuen Ziellandmarke, Extremknotenliste updaten). Dadurch kann allgemein der Aufwand und Energieverbrauch niedrig gehalten werden.

3.3 Randeffekte

Bisher bezogen sich alle Routingergebnisse auf Ziele, die innere Punkte sind. Für Knoten in Randnähe gestaltet sich das Routing deutlich schwieriger, da hier weniger Landmarken vorhanden sind und diese den Zielknoten nicht notwendigerweise umschließen. Dieses Problem tritt bei allen Routingansätzen mit virtuellen Koordinatensystemen auf. In [Ngu07] wird jedoch eine Idee eingeführt und theoretisch betrachtet, die dieses Problem vermindern kann.

```

ROUTE(Paket p)
1  if p.mode = GLDR
2    then if dist(p.bestBase) > dist(p.dest)
3      then FORWARD p toward p.bestBase
4      else if current node is in p.extremeNodes
5        then SET p.mode  $\leftarrow L_1$ 
6        else RECORD current node into p.extremeNodes
7          COMPUTE new base p.bestBase
8          if p.bestBase = NULL
9            then SET p.mode  $\leftarrow L_1$ 
10           else FORWARD p toward p.bestBase
11 if p.mode =  $L_1$ 
12   then if some neighbor nb is closer to p.dest using  $L_1$ 
13     then FORWARD p to nb
14     SET p.mode  $\leftarrow L_\infty$ 
15 if p.mode =  $L_\infty$ 
16   then if some neighbor nb is closer to p.dest using  $L_\infty$ 
17     then FORWARD p to nb
18     FLOOD to find p.dest
19

```

Abbildung 3.4: Pseudocode des Routingalgorithmus (aus [Ngu07])

Die Idee besteht in der Einführung virtueller Landmarken, welche das Netz nach außen erweitern und so auch „anziehende“ Kräfte bis an den Rand des Netzwerkes ermöglichen. Dazu wird der gesamte, nicht-innere Teil des Netzwerkes dupliziert (der $4r$ -Rand) und entlang der Netzwerkergrenze nach außen „geklappt“. Jeder Pfad zwischen einem realen und einem virtuellen Knoten muss nun über einen (realen) Knoten an der Netzwerkergrenze verlaufen und wird als „reflektierter Pfad“ bezeichnet. Analog ergibt sich die Bezeichnung der „reflektierten Entfernung“ als kürzester reflektierter Pfad zwischen einem realen und einem virtuellen Knoten.

Die virtuellen Landmarken propagieren ihre Entfernungen nun über die Knoten entlang der Netzwerkergrenze. Wenn die Netzwerkergrenze nicht besonders stark gekrümmt ist, kann sie lokal als Gerade betrachtet werden. Dann sind die Grenzknoten von den virtuellen Landmarken genau so weit entfernt, wie von den zugehörigen realen Landmarken. Diese Entfernungen können die Grenzknoten nun in der Initialisierungsphase weitergeben. Die Informationen über virtuelle Landmarken brauchen nur $12r$ weit ins Netzwerkinnere geschickt werden, denn alle weiteren Knoten sind innere Knoten und verfügen damit über genügend reale Landmarken zur Navigation.

3.4 Simulation

Der vorgestellte Routingalgorithmus wurde in [Ngu07] auch ausführlich durch Simulationen untersucht. Der Aufbau bestand aus 3200 zufällig verteilten Knoten. Die Landmarken bildeten ein 10-Sampling des Netzwerkes. In 100 Durchläufen des Landmark-Selection-Algorithmus wurde der Routerfolg zwischen jeweils 320 zufälligen Knotenpaaren geprüft. Zum Vergleich wurden die gleichen Tests auch mit *BVR* durchgeführt.

Im Mittel wurden 28,2 Landmarken erzeugt und der Routerfolg lag bei 90,9%. Die Knoten bei denen das Routing fehlschlug, befanden sich oft in Randnähe. Existieren nur wenige Landmarken, so ist *GLDR* gegenüber *BVR* im Vorteil. Dieser verschwindet jedoch bei einer höheren Anzahl an Landmarken.

Die Dilatation der Pfade war recht gering. So betrug die durchschnittliche Weglänge nur das 1,04- bis 1,09-fache des kürzesten Pfades.

4 Zusammenfassung

Die Autoren des Papers stellen sich der Frage des Routings in Ad-hoc- und Sensornetzwerken ohne Informationen über die Positionen der Knoten. Sie liefern einen Lösungsansatz für nichtgeographisches Routing anhand von Landmarken. Dabei setzen sie deutlich früher an als Vorgängerarbeiten zu diesem Thema, nämlich schon bei der Auswahl der Landmarken. Sie schlagen ein Verfahren zur Landmarkenauswahl vor, welches eine Menge Landmarken mit garantierter Güte (gleichmäßige Verteilung, geringe Anzahl) in $O(1)$ erzeugt.

Der darauf aufbauende Routing-Algorithmus basiert auf einer sehr einfachen Regel und liefert leicht bessere Ergebnisse als bereits bekannte und kompliziertere Algorithmen aus diesem Gebiet (wie BVR oder GLIDER). Seine besondere Stärke liegt jedoch im deutlich geringeren Ressourcenverbrauch. So sind aufwendige Berechnungen nur auf wenigen der Transitknoten nötig und der für die Routinginfrastruktur benötigte Speicher auf jedem einzelnen Knoten ist gering.

Die Algorithmen eignen sich jedoch nicht für alle Typen von Ad-hoc- und Sensornetzen gleichermaßen. Knotenbewegungen werden gar nicht betrachtet und ebenso können sich die Verfahren nicht auf abgebrochene oder neu entstandene Funkverbindungen einstellen. Damit sind beide Algorithmen nur für statische Sensornetze interessant.

Das Problem der Erreichbarkeit von Knoten in Randnähe ist mit [\[Ngu07\]](#) noch nicht abschließend gelöst. Die Autoren betrachten es als weiteres interessantes Forschungsgebiet.

Literaturverzeichnis

- [Car05] Caruso, A., et al.: *GPS free coordinate assignment and routing in wireless sensor networks*, in: *24th Conference of the IEEE Communication Society (INFOCOM)*, 2005. [2.1](#)
- [Fan05] Fang, Q., et al.: *GLIDER: Gradient landmark-based distributed routing for sensor networks*, in: *24th Conference of the IEEE Communication Society (INFOCOM)*, 2005. [3](#)
- [Ngu07] Nguyen, An, et al.: *Landmark Selection and Greedy Landmark-Descent Routing for Sensor Networks.*, in: *26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, S. 661–669, 2007. [1.1](#), [2.1](#), [2.2.3](#), [2.2.3](#), [3.1](#), [3.1](#), [3.2](#), [3.3](#), [3.2.1](#), [3.2.2](#), [3.3](#), [3.4](#), [3.4](#), [4](#)
- [Rao03] Rao, A., et al.: *Geographic routing without location information*, in: *9th Annual International Conference on Mobile Computing and Networking (MobiCom 2003)*, S. 96–108, ACM Press, San Diego, 2003. [2](#), [2.1](#)