Vertex Cover Problem

Svetlana Mareva

Fachbereich Informatik, Freie Universität, Berlin
mareva@inf.fu-berlin.de

Drahtlose Sensornetze (engl. Wireless Sensor Networks) werden heute in vielen Gebieten eingesetzt: Risiko- und Umweltüberwachung, Militär und Technik. Eine der größten Herausforderungen ist das Leben des Sensornetzes optimal zu verlängern und dabei eine gute Abdeckung des überwachten Gebietes zu erhalten. In diesem Paper werden Algorithmen untersucht, die die Knotenüberdeckung (engl. Vertex Cover) bei der Verwendung von neu hinzugefügten Sensoren verbessern. Dabei werden verschiedene Ansätze beschrieben, die das beste Vertex Cover mit Hilfe der Graphentheorie finden.

***Schlüsselwörter*— Überdeckung, Graphentheorie, Sensornetz.**

I. Einführung

Mit der Entwicklung der modernen Technologien, nimmt die Rolle der Sensornetze zu. Sensornetze präsentieren eine Vielfalt von Netzbetrieb-Modellen für Datenablage und Datenabwicklung. Sie spielen eine große Rolle bei der Überwachung im Militär, Gebäudesicherung, Target Tracking, usw. Die billigen Sensoren, die von einer kleinen Batterie betrieben werden, sind leicht in großen Mengen zu produzieren. Die große Herausforderung für die Designer der Sensornetze ist die Verlängerung der Lebensdauer der Netze. In manchen Gebieten der Anwendung, wie z.B. Gebäudeüberwachung bei der Feuerwehr, ist es kritisch, wenn ganze Flächen nicht überdeckt sind. Man muss die Arbeit der kleinen Sensoren so optimieren, dass das Netz in einem längeren Zeitintervall überlebt, wobei eine maximal große Fläche überdeckt wird. Die Überdeckungsprobleme in Sensornetzen, die erforscht werden, umfassen worst-case und best-case Überdeckung, Flächenüberdeckung und Instandhaltung der Konnektivität. Das Quality of Service (QoS) wird von der Worst-Case Überdeckung (definiert in Meguerdichian u.a [1]) gemessen, indem man den Pfad mit der niedrigsten Beobachtbarkeit unter allen möglichen Pfaden zwischen einem Anfangs- und einem Endknoten findet. Man nennt diesen Pfad den maximalen Lückenpfad (Maximum Breach Path), da er am Weitesten von allen Sensoren liegt. Mit der Best-Case Überdeckung werden Bereiche mit hoher Betreuung und Beobachtbarkeit von Sensoren gefunden.

Die präsentierten Lösungen laufen in polynomieller Zeit und kombinieren Methoden der Geometrie und graphentheoretische Algorithmen wie: Voronoi Diagramme, Delaunay-Triangulation . Der erste vorgestellte Algorithmus von Meguerdichian u.a. [1] ist eine Heuristik zur Verbesserung der Überdeckung beim Hinzufügen von neuen Sensoren in dem Netz. Die zweite Heuristik von Hou u.a. [2] findet die optimale Stationierung mit der maximalen Überdeckung in polynomielle Zeit.

Der Rest des Papers wird folgendermaßen aufgeteilt: im 3.Abschnitt wird ein Teil des aktuellen Forschungsstandes vorgestellt, indem man manche Algorithmen, die sich mit dem Thema beschäftigen kurz erläutert. Im 3.Abschnitt werden die Begriffe Voronoi Diagramme und Delaunay- Triangulation näher betrachtet und erklärt. Im 4.Abschnitt wird die Heuristik von Meguerdichian u.a. [1] vorgestellt. Der Algorithmus von Hou u.a. [2] wird im 5.Abschnitt gezeigt. Im 6.Abschnitt werden experimentelle Ergebnisse der Algorithmen präsentiert.

II. Forschungsstand

Das Überdeckungsproblem (engl. Coverage Problem) wurde von mehreren Forschern studiert. Es wurden viele Algorithmen entwickelt, die verschiedene Themengebiete umfassen. In diesem Paper hat man sich auf graphentheoretische Algorithmen konzentriert.

Ein wichtiges Problem in Sensornetzen ist eine bestmögliche Überdeckung zu haben und gleichzeitig Energie zu sparen, da die Batterien von Sensoren eine relativ kurze Lebensdauer haben. Die Idee der entwickelten Algorithmen ist Sensoren abwechselnd arbeiten zu lassen, in dem manche Sensoren aktiv sind, während andere in einem Sleep-Modus warten. Der Algorithmus von Wang u.a. [3] versucht dieses Problem zu lösen ohne die genaue Position der Sensoren zu kennen. Falls es Überdeckungslücken im Sensornetz gibt, findet der Algorithmus zusätzlich noch diese Lücken und stellt eine Lösung dazu vor.

Der Algorithmus von Cheng u.a. [4] löst das gleiche Problem ohne die genaue Position der Sensoren zu kennen. Dadurch dass man sich auf die Distanzinformationen anstatt genauer Positionsinformationen beschränkt, kann man den Overhead, der bei Positionsbestimmung anfällt, vermeiden und dadurch die Lebensdauer erhöhen. Damit werden überflüssige Berechnungen gespart.

Der Algorithmus von Bahi u.a. [5] löst das Überdeckungsproblem mit Hilfe von sogenanten Mobile Beacons. Beacons sind Sensoren, die einen GPS-Receiver haben und ihre Position kennen. Da es ziemlich umständlich ist, alle Sensoren als Beacons zu haben, versucht man die Anzahl der Beacons zu minimieren. In diesem Algorithmus wird ein Schema mit Mobile Beacons vorgestellt, die ihre Position ändern. Die mobilen Beacons schicken Pakete zu

ihren Nachbarsensoren mit Information über ihre Position und somit helfen sie den anderen Sensoren, ihre Posistion zu bestimmen. Es wird gleichzeitig Energie gespart, da die Sensoren nur mit den mobilenn Beacons kommunizieren.

III. Begriffseinführung

A. Geometrische Begriffe

Voronoi Diagramme und Delaunay-Triangulation werden in unterschiedlichen Gebieten der Wissenschaft eingesetzt. In diesem Paper werden Algorithmen vorgestellt, die die dazugehörigen Verfahren benutzen. Dafür werden sie näher betrachtet.

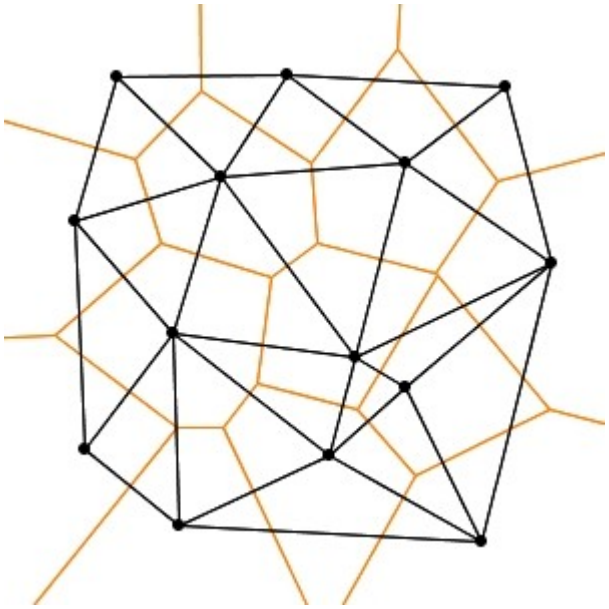
1) Voronoi Diagramm

Gegeben sei ein Sensornetz in einer zweidimensionalen Ebene mit einer Menge von Sensoren S, wobei |S|=n. Aus dem gegebenen Sensornetz kann man eine Partition von S bilden, indem man jedem Punkt aus der Ebene seinen benachbarten Sensor zuordnet. Alle Punkte, die einem Sensor p zugewiesen sind, bilden die Voronoi Region, V(p). Das Voronoi Diagramm,V(S), besteht aus allen Voronoi Regionen, V(p1), ..., V(pn) [6].

Voronoi Regionen sind konvex. Die Kanten in einem Voronoi Diagramm heißen Voroinoi Kanten und die Knoten- Voronoi Knoten. Jede Voronoi Kante muss mit zwei am nächst liegenden Sensoren assoziiert werden und jeder Voronoi Knoten hat mindestens drei am nächst liegende Sensoren. Figur 1 zeigt ein Voronoi Diagramm (orange) und eine Delaunay-Triangulation (schwarz). Eigenschaften des Voronoi Diagramms sind: es gibt keine 4 Punkte (Sensoren) aus S, die auf einem Kreis liegen; jeder Voronoi Knoten ist von Grad drei.

2) Delaunay- Triangulation

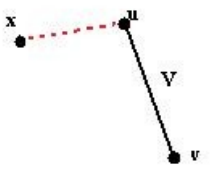
Die Delaunay- Triangulation von S, D(S), ist der duale Graph von dem Voronoi Diagramm. Man konstruiert die Delaunay-Triangulation, indem man die Knoten aus S verbindet, die einer gemeinsamen Voronoi Kante zugeordnet sind. Eine wichtige Eigenschaft der Delaunay- Triangulation ist, dass dessen Eckpunkte gleichzeitig die Umkreismittelpunkte der Voronoipolygone bilden. Das Voronoi Diagramm eines Voronoi Diagramms ist folglich die Delaunay- Triangulation [7].



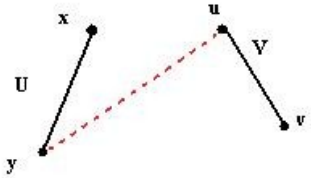
Figur 1: Illustration eines Voronoi Diagramms und einer Delaunay- Triangulation [1]

B. Notation und Definitionen

Als erstes wird die verwendete Notation vorgestellt. Eine zweidimensionale Ebene mit einem Sensornetz sei gegeben. Sei die Menge aller Sensoren S mit n Knoten. Die Euklidische Distanz zwischen je zwei Knoten x und y wird mit |xy| bezeichnet. Sei V eine beliebige Teilmenge von S und x ein Knoten aus S, mit dist (x,V) wird die kleinste Distanz zwischen x und jedem anderen Knoten aus V bezeichnet. Gegeben zwei Mengen von Knoten U und V, mit *dist (U,V)* sei die kleinste Distanz zwischen jedem Knoten aus U und jedem Knoten aus V definiert und mit *cover (U,V)* die Überdeckungsdistanz $max_{x \in U}\ dist(x,V)$. Figuren 2 und 3 zeigen die Definitionen von *dist (x,V)* und *cover (U,V)*.



Figur 2. Illustration von dist *(x,V)*



Figur 3. Illustration von *cover(U,V)*

Gegeben ein Pfad P(s,t) zwischen einem Startpunkt s und einem Endpunkt t. Das Support von P(s,t) ist *cover(P(s,t),S)*, das Support misst die niedrigste Beobachtbarkeit eines Punktes auf P. Breach von P(s,t) ist definiert als *dist(P(s,t),S)* und misst die beste Beobachtbarkeit eines Punktes auf P. Sei *support(P)* die Support-Grösse von P und *breach(P)* die Breach-Grösse von P. Wir definieren die beste Überdeckung und die schlechteste Überdeckung für ein Sensornetz.

Definition: Ein Pfad, der das Minimum Support hat, nennt sich der beste Support Pfad (Figur 4), d.h. alle Punkte die auf dem Pfad liegen, haben insgesamt die durschnittlich beste Beobachtbarkeit.

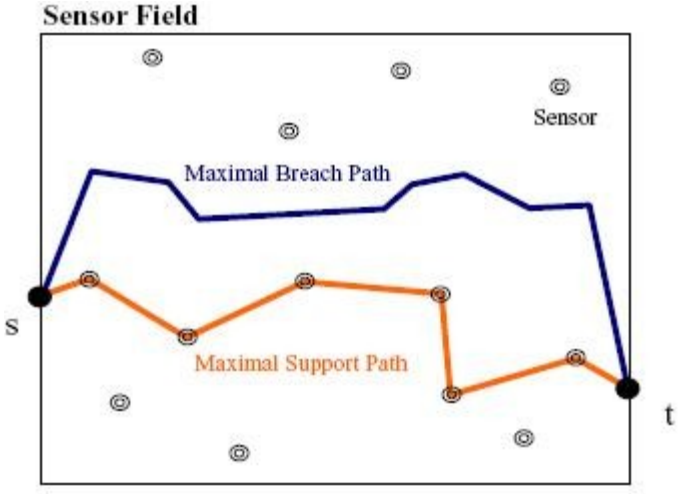
Definition: Ein Pfad, der das Maximum Breach hat, nennt sich Maximum Breach Pfad (Figur 4), d.h. alle Punkte die auf dem Pfad liegen, haben insgesamt die durchschnittlich schlechteste Beobachtbarkeit.

Gegeben einen Startpunkt s und einen Endpunkt t in einem Sensornetz. Support des besten Support Pfades nennt sich das Support des Sensornetzes. Und Breach des Maximum Breach Pfades- Breach des Sensornetzes. Das Überdeckungsproblem für ein Wireles Sensornetz ist den besten Support Pfad und den Maximum Breach Pfad zu finden.

Wenn wir die Überdeckung eines Sensornetzes verbessern wollen, ergibt sich das Problem, wie wir neue Sensoren einsetzen, damit die Überdeckung optimal wird. Daraus ergeben sich zwei weitere Probleme.

Definition: Das Problem der besten Überdeckung bei Einsatz neuer Sensoren, ist die besten Einsatzstellen für k zusätzliche Sensoren zu finden, so dass das Support des Netzes am meisten verbessert wird.

Definition: Das Problem der schlechtesten Überdeckung bei Einsatz neuer Sensoren, ist die Einsatzstellen für k zusätzliche Sensoren zu finden, so dass das Breach des Netzes sein Maximum erreicht, d.h. die Beobachtbarkeit des Sensornetzes ist am schlechtesten.



Figur 4. Illustration von Maximal Breach Path und Maximal Support Path [1]

Die folgenden zwei Definitionen unterscheiden sich von den obigen zwei nur darin, dass keine zusätzliche Sensoren im Netz hinzugefügt werden.

Definition: Das Problem der besten Überdeckung, ist den Maximal Support Pfad zu finden, um das beste Support des Netzes zu bestimmen.

Definition: Das Problem der schlechtesten Überdeckung, ist den Maximal Breach Pfad zu finden, um die schlechteste Beobachtbarkeit des Netzes zu bestimmen.

In folgendem Absatz werden zwei Algorithmen vorgeschlagen. Der erste Algorithmus löst das Problem der besten Überdeckung in einem Sensornetz und der zweite Algorithmus löst das Problem der besten Überdeckung bei Einsatz von zusätzlichen Sensoren in einem Sensornetz.

IV. Das Problem der besten Überdeckung.

A. Der Algorithmus von Meguerdichian u .a.[1].

Gegeben ein Sensornetz A mit Sektoren I und F. Finde den maximalen Support Pfad P, der in I startet und in F endet. Die Idee des Algorithmus ist einen Pfad zwischen I und F zu finden, so dass der Support von diesem Pfad das Minimum unter allen möglichen Pfäden ist. Der minimale Wert des Supports zeigt die Überdeckung im schlechtesten Fall und maximiert die Beobachtbarkeit entlang des Pfades. Dabei wird die Delaunay Triangulation verwendet. Eine wichtige Eigenschaft der Delaunay Triangulation ist, dass jede Kante in den konstruierten Dreiecken minimale Länge hat. Daraus folgt, dass der minimale Support Pfad auf diese Kanten liegt.

Algorithmus:

1. Generiere Delaunay Triangulation.
2. Generiere Graph G aus der Delaunay Triangulation :
   - jede Linie in der Triangulation ist eine Kante in G mit Gewicht Länge der Linie in der Triangulation
   - jeder Sensor in der Triangulation ist ein Knoten in G
3. Suche nach Support Pfad Ps
   - prüfe ab, ob ein Pfad zwischen I und F existiert, verwende dabei Breitensuche und Binäre Suche
   - führe Binäre Suche zwischen den Kanten mit dem kleinsten und dem grössten Gewicht aus
   - überprüfe in jedem Schritt der Binären Suche, falls ein Pfad existiert zwischen Kanten mit Gewicht kleinergleich dem spezifizierten Wert von support_width
   - P ist der Minimale Support Pfad

Anmerkung: Jede Kante in P hat Gewicht kleinergleich support_width. Support_width entspricht der oberen Schranke für den Gewicht aller Kanten auf dem minimum Support Pfad, d.h. es gibt mindestens eine Kante mit Gewicht gleich support_width.

B. Komplexität des Algorithmus von Meguerdichian u.a.

[1]

- Konstruktion von Delaunay Triangulation: O (n log n)
- Graph Konstruktion und Gewichtung der Kanten : O(n)
- Breitensuche: O(m), wobei m die Anzahl der Kanten ist. Komplexität hängt von der Dichte des Graphen. O(n) bei dünnbesetzten und $O(n^2)$ bei dichten Graphen.
- Binäre Suche : O(log n)

Total: O(n log n) für dünnbesetzte Graphen und $O(n^2 \log n)$ für dichte Graphen.
Beweis der Korrektheit wird der Einfachheit halber weggelassen.

V. Das Problem der besten Überdeckung bei Hinzufügen von neuen Sensoren zu einem bestehenden Netzwerk

A. Der Algorithmus von Hou u. a. [2].

Um die Überdeckung des Sensornetzes am meisten zu verbessern, versucht der Algorithmus das Support des maximalen Support Pfades im Netz zu vergrößern. Die Idee des Algorithmus ist, einen Kandidat-Pfad zu finden mit dem größten Support und den Support des Pfades danach am meisten zu verbessern. In diesem Algorithmus wird Dijkstra's Algorithmus [5] zum Finden der kürzesten Wege in Graphen modifiziert und verwendet. Statt die Summe der Kantengewichte auf dem Pfad zu verwenden, wird das maximale Gewicht des Pfades als Pfadkosten genommen.

Dafür wird der Algorithmus Pfad-Kosten (P,k) verwendet, wobei weight[e] das Gewicht von der Kante e ist, n[e] die Anzahl von neuen Sensoren ist, die auf der Kante e platziert werden und key[e] das Gewicht der Kante e bei Einsatz zusätzlicher Sensoren auf e beobachtet. Das maximale Gewicht des Pfades vergrößert sich dann, wenn ein neuer Sensor auf der Kante des Pfades mit dem maximalen Gewicht platziert wird. Die neuen Sensoren werden dann auf diesem Pfad eingesetzt.

Pfad-Kosten (P,k)
1. Für jede Kante e auf dem Pfad P
2. do { key [e] = weight (e)
3. n [e] = 0 }
4. für i =1 bis k
5. do { e= Kante mit dem maximalen Key-Wert auf dem Pfad P
6. n[e]= n[e]+1
7. key[e]= weight[e] / (n[e]+1) }
8. e = Kante mit dem maximalen Key-Wert auf dem Pfad P
9. return key[e]

Bei Pfad-Kosten (P,k) wird das Gewicht der Kante e, weight[e] am Anfang auf die Hälfte der Euklidischen Dystanz zwischen u und v (1/2 |uv|) gesetzt. Falls wir n[e] Knoten zwischen den Endknoten des Pfades P mit k Knoten insgesamt gleichmäßig auf die Kante e verteilen, der Wert, der von dem Pfad-Kosten Algorithmus, das maximale Gewicht der Kanten in dem neuen Pfad wird gleich :

$$\max_{e \in P} (\text{weight}[e] / (n[e] + 1)).$$

Der oben angegebene Wert ist minimal und kann per Induktion bewiesen werden. Im unten vorgeschlagenen Algorithmus für das Problem der besten Überdeckung bei Einsatz neuer Sensoren wird dieser Wert benutzt. Es wird davon ausgegangen, dass der Anfangs- und der Endpunkt des maximalen Support Pfades Sensoren sind.

BEST-COVERAGE-DEPLOY
1. Für jedes Paar von Sensoren u und v, konstruiere die Kante uv mit Gewicht ½ |uv|.
2. Wende das modifizierte Dijkstra Algorithmus an, um den kürzesten Pfad Pc, zwischen s und t zu finden.
3. Füge k neue Sensoren auf dem Pfad hinzu, der im 2.Schritt gefunden wurde:
   Für jede Kante e in Pc, füge n[e] Knoten gleichmäßig verteilt zwischen den Endknoten von e hinzu.

Beweis der Korrektheit wird der Einfachheit halber weggelassen. Siehe [2].

B. Zeikomplexität des Algorithmus von Hou

Gegeben ein Sensornetz mit n Sensoren. Im ersten Schritt des Algorithmus, braucht er $O(n^2)$ Zeit um alle Kanten zu konstruieren und $O(n^2)$ um das Gewicht aller Kanten zu bestimmen. Im zweiten Schritt wird der modifizierte Dijkstra Algorithmus angewendet, um den kürzesten Pfad zu finden. Das braucht $O(n \log n + n^2) = O(n^2)$ Zeit. Im dritten Schritt wird O(k) Zeit benötigt, um die neuen k Knoten hinzuzufügen. Der Algorithmus braucht insgesamt $O(n^2)$ Zeit.

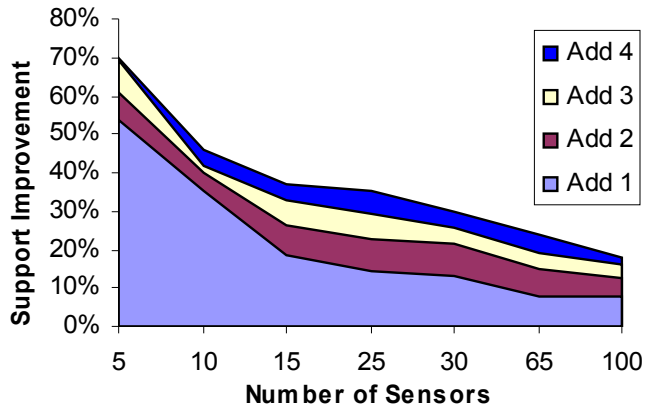
VI. Experimentelle Resultate

A. Algorithmus von Meguerdichian

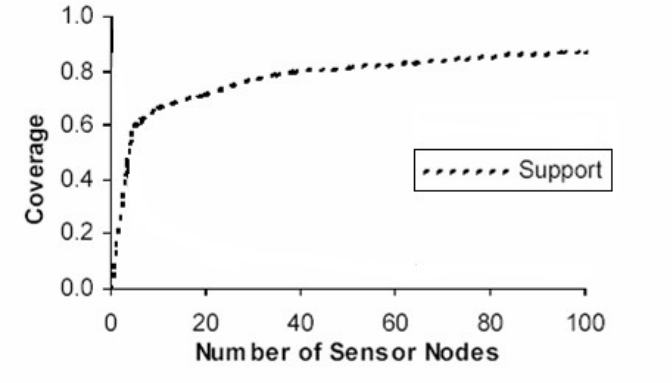
Die vorgeschlagene Heuristik wurde evaluiert. In Figur 5 wird die Verbesserung des Supports in % gezeigt bei Hinzufügen von 1,2,3 und 4 Sensoren auf die Kanten des besten Support Pfades im Sensornetz, wobei die Anzahl von Sensoren von 5 bis 100 steigt. Figur 6 zeigt das asymptotische Verhalten bei chaotischem Einsatz von 100 neuer Sensoren in einem Sensornetz mit 100 Sensoren.

Support=1-support_weight (support_weight entspricht der Länge der jeweiligen Kante, d.h. je kleiner support_weight ist, desto besser das Support). Figur 6 zeigt, dass das Support des Sensornetzes sogar dann steigt, wenn neue Sensoren chaotisch

hinzugefügt werden, angenommen es gibt genug solche Sensoren zur Verfügung.



Figur 5. Verbesserung des Supports im Algorithmus von Meguerdichian [1]

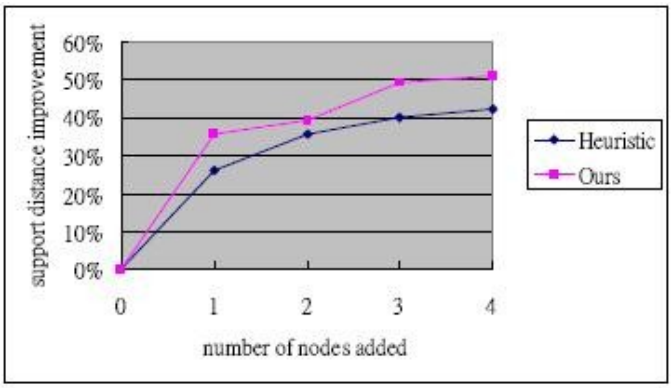


Figur 6. Verbesserung des Supports bei chaotischem Einsatz von zusätzlichen Sensoren im Algorithmus von Meguerdichian [1]

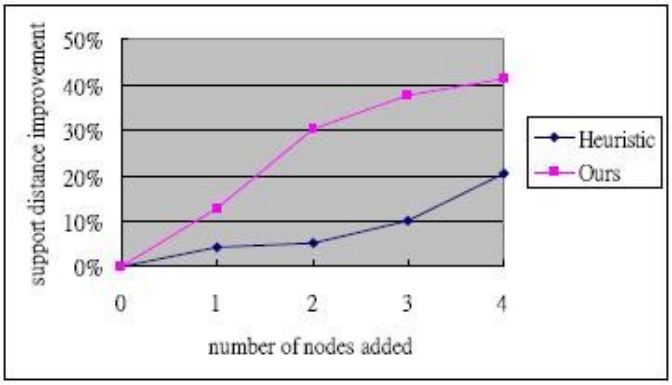
B.Algorithmus von Hou u.a. [2]

Der Algorithmus von Hou wurde evaluiert und mit dem Algorithmus von Meguerdichian u.a. [1] verglichen. Dabei werden zwei Sensornetze getestet: mit 25 und 80 Sensoren am Anfang.

Figur 7 zeigt die Support- Verbesserung bei Einsatz von 1 bis 4 Sensoren in einem Sensornetz mit 25 Sensoren. Aus dem Diagramm ist es leicht zu sehen, dass der Algorithmus von Hou bessere Resultate aufweist. Figur 8 zeigt die Support-Verbesserung in einem Sensornetz mit 80 Sensoren. Der Algorithmus von Hou ist offensichtlich in diesem Test auch besser.



Figur 7. Verbesserung des Supports in einem Sensornetz mit 25 Sensoren [2]



Figur 8. Verbesserung des Supports in einem Sensornetz mit 80 Sensoren [2]

VII. Abschluss

In diesem Paper wurden zwei Algorithmen vorgestellt und ihre Resultate experimentell verglichen. Der Algorithmus von Meguerdichian u.a. [1] bestimmt den maximalen Support-Pfad in einem Sensornetz, wobei das Maximum Support des Netzes bestimmt wird. Der Algorithmus kann leicht modifiziert werden, um den maximalen Breach-Pfad zu finden. Daraus ergibt sich der Pfad mit der schlechtesten Beobachtbarkeit unter allen, d.h. man kann versuchen diesen Pfad zu verbessern in dem man neue Sensoren dem Pfad entlang hinzufügt.

Der zweite Algorithmus von Hou u.a. [2] findet das

bestmögliche Support beim Hinzufügen von Sensoren. Der Algorithmus findet einen Kandidat-Pfad mit dem besten (minimalen) Support und fügt Sensoren auf dem Pfad hinzu. Dabei wird das Support des Sensornetzes verbessert. Der Algorithmus von Hou zeigt wesentlich bessere Resultate als der Algorithmus von Meguerdichian.

Viele andere Algorithmen ergeben sich als hilfreich in dem Bereich der Sensornetz-Technik. Die Sensornetze werden häufig in Gebieten eingesetzt, wo es sehr schwer möglich ist, die Batterien der Sensoren zu wechseln. Deswegen braucht man effiziente Algorithmen zur Verlängerung der Lebensdauer der Sensoren. Dabei werden spezielle Sleep-Awake Mechanismen entwickelt, die bestimmte Sensoren nach einer Zeitspanne ausschalten und wieder einschalten. Solche Algorithmen sind sehr wichtig für die Überwachung im Militär.[10]

VIII. Referenz

[1] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M.Srivastava "Coverage Problems in Wireless Ad-Hoc Sensor Network," Proc. IEEE INFOCOM '01, pp. 1380-1378, 2001.

[2] Yung-Tsung Hou, Tzu-Chen Lee, Chia-Mei Chen, Bingchiang Jeng "Node Placement for Optimal Coverage in Sensor Networks" Proc. of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)

[3] Wei-Tong Wang, Kuo-Feng Ssu, Hewijin Christine Jiau „Density Control without Location Information in Wireless Sensor Networks"

[4] Yang-Min Cheng ,Li-Hsing Yen „Range-Based Density Control forWireless Sensor Networks " Proc. of the 4th Annual Communication Networks and Services Research Conference (CNSR'06)

[5] Jacques M. Bahi, Abdallah Makhoul and Ahmed Mostefaoui "Localization and Coverage for High Density Sensor Networks" Proc. of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops(PerComW'07)

[6] Geometrische Algorithmen − SS 2001 − Th. Ottmann

[7] Fortune, Steven: Voronoi diagrams and Delaunay triangulation. In: Du, Ding-Zhu (Hrsg.) ; Hwang, Frank (Hrsg.): *Computing in Euclidean Geometry (Lecture Notes Series on Computing 1)*, World Scientific, 1992, S. 193-230

[8] T.J. Cormen, C.E. Leiserson, and R.L. Rivst, Introduction to Algorithms. MIT Press and McGraw-Hill, 1990.

[9] S. Fortune, "Voronoi Diagrams and Delaunay Triangulations," Computing in Euclidean Geometry, F.K. Hwang and D.-Z. Du, eds., pp. 193-233, Singapore: World Scientific, 1992.

[10] Chao Gui and Prasant Mohapatra, "Power Conservation and Quality of Surveillance in Target Tracking Sensor Networks"

[11] D.P. Dobkin, S.J. Friedman, and K.J. Supowit, "Delaunay Graphs are almost as good as complete graphs," Discrete Computational Geometry, 5:399-407, 1990.