Distributed Covering by Ant-Robots Using Evaporating Traces

27. Januar 2008

by Israel A. Wagner, Michael Lindenbaum und Alfred M. Bruckstein
veröffentlicht: Oktober 1999

aufbereitet von Benjamin Reichelt (uni@benreichelt.de)

für das Seminar: "Ausgewählte Kapitel der Informatik" von Herrn Professor Dr. R. Schrader an der Universität zu Köln

Inhaltsverzeichnis

1 Einführung

Ameisen (engl. ants) und andere in Kolonien lebende Insekten benutzen bestimmte Chemikalien zur Kommunikation und Koordination von bestimmten Aufgaben. Diese Chemikalien werden Pheromon genannt. Die Autoren des zugrunde liegenden Papers untersuchen die Möglichkeit, ob ein System von unabhängigen Robotern auch auf diese Art und Weise kommunizieren und in angemessener Zeit gegebene Problem lösen kann. Als Einsatzgebiete könnte man sich die Säuberung, Erschließung oder Überwachung eines unbekannten, sich mit der Zeit verändernden Gebietes vorstellen. Die eingesetzten Roboter, im folgenden meist Agenten genannt, sollen nur über abgelegte, mit der Zeit verdunstende Geruchsspuren kommunizieren. Die Agenten sollen in der Lage sein, Kanten bzw. Knoten der zugrunde liegenden Struktur mit Duftstoffen, Farben oder Temperaturen zu markieren. Es wird angenommen, dass die Intensität dieser Markierungen mit der Zeit abnimmt. So sind die Agenten in der Lage festzustellen, ob und wann welcher Ort bereits besucht wurde. Als abstraktes Modell wird ein sich unabhängig voneinander auf einem Graphen G mit Kanten E und Ecken V bewegendes System von Multiagents mit geteiltem Speicher verwendet. Als Knoten werden rechteckige Bodenfliesen und als Kanten deren Kanten angenommen. Im Gegensatz zur bereits existierenden Lösungsmethode depth first search (siehe Abschnitt 3: Algorithmus 2) ist der hier vorgestellte Algorithmus Ant-Walk-2 dynamisch, das bedeutet, er löst das Problem auch, wenn einer der Agenten ausfällt oder die zugrunde liegende Struktur verändert wird (Ecken oder Kanten könnten während des Suchvorgangs hinzugefügt oder entfernt werden). Im konkreten Beispiel bedeutet dies, dass zum Beispiel Möbel auf der zu reinigenden Fläche versetzt werden oder Türen geöffnet bzw. geschlossen werden. Zusätzlich können die auf Ant-Walk-2 basierenden Agenten auch bei Einflüssen durch äußere Störungen (siehe Abschnitt 2.2) ihre Aufgabe vollenden, natürlich nimmt die dafür benötigte Laufzeit mit der Größe der Störung zu. Das System von selbstständig agierenden Agenten könnte zum Beispiel nach einem nuklearen Zwischenfall eingesetzt werden, falls es durch Einflüsse der radioaktiven Strahlung unmöglich ist, mittels elektromagnetischer Wellen zu kommunizieren. Die Autoren haben drei Algorithmen, die dieses Problem mit und ohne Einflüsse durch Störungen von außen lösen, entwickelt, auf Konvergenz und worst-case Laufzeit geprüft und experimentell getestet.

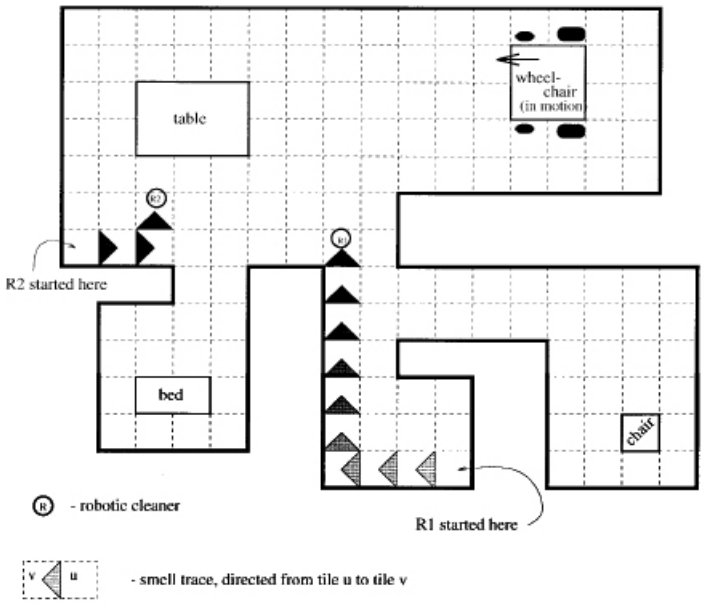

Abbildung 1: Dynamisches Modell einer zu überdeckenden (zu reinigenden) Fläche

1.1 Definition des Problems

Ein System von Agenten, die nach dem von den Autoren entwickelten Algorithmen, soll in der Lage sein, ein unbekanntes, sich veränderndes System (in Form eines zusammenhängendes Graphen) überdecken, ohne dass von außen in irgendeiner Form eingegriffen werden muss. Die Agenten steuern selbstständig anhand der gespeicherten Bewegungsregeln ihre Bewegungen und überdecken den zugrunde liegenden Graphen vollständig innerhalb der vorgestellten Laufzeitschranke (siehe Theorem 2). Auch Einflüsse durch Störungen von außen z.B. in Form von Messfehlern hindern die Agenten nicht an der Lösung des gestellten Problems, sondern zögern diese nur eventuell hinaus.

2 Ant-Walk-1

Algorithm 1 Ant-Walk-1

.	Rule Ant-Walk-1(u vertex)
A)	$t := 1 + t$
B)	Finde eine Kante (u, v) ausgehend von u, so dass $s(u, v) = \min_{w \in N(u)} \{s(u, w)\}$
.	/Falls mehrere solchen Kanten existieren, triff eine heuristische Entscheidung/
C)	Gehe zu v und setze $s(u, v) := t$;
D)	Beende Ant-Walk-1

Als Modell für die zu überdeckende Fläche nimmt man den Graphen $G(V, E)$ an. Jeder Knoten u des Graphen entspricht einer zu überdeckenden Kachel, jede Kante (u, v) entspricht der Abgrenzung der Kacheln u und v. Für jede Kante bezeichnet man die Markierung der Kante (u, v) mit $s(u, v)$ (initialisiert mit: $s(u, v) := 0$). Die Markierung (z.B. Pheromon oder Temperatur) nimmt exponentiell mit der Zeit ab. Daher ist es sinnvoll, die Markierung als Zeitmarkierung t im folgenden zu verwenden. Der Ant-Walk-1 Algorithmus basiert nicht auf der internen Speicherung von Daten der Agenten. Die einfachen Verhaltensregeln, nach denen sich die Agenten richten, sind folgende:

1. Suche ausgehend von u die Kante, die die geringste Markierung aufweist, da diese Kante am längsten nicht besucht worden war.
2. Besuche den auf an die Kante angrenzenden Knoten v und markiere die Kante (u, v) neu.

Die Größe der Agenten bzw. der Knoten lässt es zu, dass mehrere Agenten zur gleichen Zeit den gleichen Knoten besuchen. Wer den Knoten zuerst verlässt, kann zum Beispiel durch eine ID-Kodierung vorgeschrieben werden. Dies ist nötig, um mögliche Kollisionen unter den Agenten zu vermeiden.

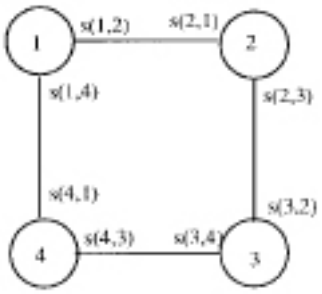

Abbildung 2: Visualisierung für einen Graphen mit vier Knoten und einen Agenten.

2.1 Analyse von Ant-Walk-1

In diesem Abschnitt wird eine obere Schranke für die Laufzeit des Ant-Walk-1 Algorithmus (ohne angenommene Störungen und Messfehler s. Abschnitt 2.2) gesucht.

Für den Fluss $f(u,v)$ durch eine Kante (u,v), der definiert ist als die Anzahl der Überquerungen durch Agenten von u nach v. Es gilt:

Korollar4

$$\forall_{(u,v),(x,y)\in E} : |f(x,y) - f(u,v)| \leq k\rho(G) + n$$

Der Beweis dieses Korollars findet sich im Original Paper. Wir nehmen also als gegeben an, dass sich der Fluss durch zwei beliebige Kanten nicht um mehr als $k\rho(G)+n$ unterscheiden kann. Wobei $\rho(G)$ als Schnitt-Widerstand bzw. Dichte von G bezeichnet werden kann.
Das folgende Theorem gibt eine obere Laufzeitschranke für die Überdeckung eines Graphen G durch k Agenten an:

Theorem1

$$t_k \leq n\Delta\left(\frac{n}{k} + \rho(G)\right)$$

mit $\Delta :=$ maximale Grad der Knoten in G, $n := |V(G)|$, $\rho(G)$ Dichte oder Schnitt-Widerstand von G.

Beweis Wir nehmen an, dass die in Theorem 1 angegebene Zeit verstrichen ist und dennoch eine Kante unbesucht, der Fluss durch diese Kante $f(x,y) = 0$. Nach der Zeit $t \geq t_k$ ist also der Gesamtfluss in G $F = \sum_{(u,v)\in E} f(u,v) = kt$. Nach Korollar 4 kann sich der Fluss durch zwei besuchte Kanten nicht um mehr als $k\rho(G)+n$ unterscheiden.

$$\begin{aligned} F = kt &= \sum_{(u,v)\in E} f(u,v) = \sum_{(u,v)\in E\setminus\{(x,y)\}} f(u,v) + f(x,y) \\ &\overset{K4}{\leq} (|E(G)| - 1)\cdot(k\rho(G)+n) + 0 \\ &\leq (|E(G)|)\cdot(k\rho(G)+n) \end{aligned}$$

Die Anzahl der Kanten kann durch Anzahl der Ecken n und der maximalen Kantenanzahl Δabgeschätzt werden: $|E(G)| \leq \frac{n\Delta}{2}$. Es gilt also:

$$F = kt \leq (|E(G)|) \cdot (k\rho(G) + n) \leq \frac{n\Delta}{2} (k\rho(G) + n)$$

Löst man diese Ungleichung nach t auf, erhält man:

$$t \leq \frac{n\Delta}{2} \left(\rho(G) + \frac{n}{k}\right)$$

Nach Annahme gilt also:

$$n\Delta \left(\frac{n}{k} + \rho(G)\right) \geq t_k > t \leq \frac{n\Delta}{2} \left(\rho(G) + \frac{n}{k}\right)$$

Aus diesem Widerspruch folgt die Behauptung, also Theorem 1. □

Die berechnete obere Schranke ist fest bis auf eine Konstante, die auftritt, falls ein Agent mehrere Male vor- und zurückgeht, bis er den ganzen Graphen überdeckt. Dieses Problem tritt möglicher Weise aufgrund von heuristisch getroffenen Entscheidungen auf. Intuitiv lässt sich der Teil $\left(\frac{n}{k} + \rho(G)\right)$ aus Theorem1 - wie folgt - erklären: Liegt der Graph G dicht, so ist der Durchflusswiderstand $\rho(G)$ gering und der Term $\frac{n}{k}$ signifikant (z.B. die Vergrößerung der Agentenanzahl k erzeugt eine schnellere Überdeckung). Ist auf der anderen Seite die Dichte des Graphen G gering und damit $\rho(G)$ groß, so macht der Einsatz von vielen Agenten nicht unbedingt Sinn.

2.2 Auswirkungen von Störungen und Messfehlern auf Ant-Walk-1

Erfahrungsgemäß funktionieren die Sensoren und Markierer der verwendeten Roboter nicht immer einwandfrei. Durch Störungen in der Umgebung und Messfehler kann es je nach verwendetem Algorithmus zu abnormen Verhalten der Roboter kommen. Ist die Ausfallwahrscheinlichkeit eines Roboters p. Die Wahrscheinlichkeit, dass keiner der k Roboter ausfällt, ist also $(1-p)^k$. Ein Algorithmus, der das fehlerfreie Verhalten aller k Roboter verwendet, hat also eine Ausfallwahrscheinlichkeit von $1-(1-p)^k$. Es gilt $\lim_{k\to\infty} 1-(1-p)^k = 1$.
Sei α die Störung bzw. der Messfehler der Sensoren, dann weicht die Messung der Markierung $s(u,v)$ höchstens und $\frac{\alpha}{2}$ nach oben bzw. nach unten ab. Es gilt also bei einer Markierung der Kante (u,v):

$$s(u,v) - \frac{\alpha}{2} \leq s_{Mess}(u,v) \leq s(u,v) + \frac{\alpha}{2}$$

Ein Algorithmus, der auf der Funktion dieses Sensors basiert, kann also nicht zwischen einer Markierung, die zum Zeitpunkt t und zum Zeitpunkt $t+\alpha$ hinterlassen wurde, unterscheiden. Die bisher gemachten Vorhersagen müssen also korrigiert werden. Theorem1 muss also bezüglich der Fehler korrigiert werden zu:

Theorem1α $t_k \leq n\Delta\left(\frac{(1+\alpha)n}{k} + \rho(G)\right)$

Man beachte, dass hier die Zeit zur Überdeckung des gesamten Graphen durch Erhöhung der Anzahl von eingesetzten Roboter verringert werden kann. Zusammenzufassend lässt sich sagen: Bei großen Störungen ist es sinnvoll, mehrere Agenten einzusetzen, um das Problem in einer angemessenen Zeit zu lösen.

3 Ant-Walk-2

Da Ant-Walk-1 bei "ungünstiger" Graphen in gewisse Fallen tappen kann, die zu unnötigen Wiederbesuchen einiger Knoten führen können, gilt es diesen nun zu optimieren. Ein effizienterer Algorithmus als Ant-Walk-1 ist Ant-Walk-2. Wie zu erwarten muss man dazu natürlich etwas investieren. Die für Ant-Walk-1 noch recht einfach gehaltene Hardware wird nun komplexer. Ant-Walk-2 basiert ist im Grunde eine Verallgemeinerung des depth-first-search (kurz: DFS) Algorithmus. Die grundlegende Idee von DFS ist folgende:

- Suche einen Nachbarn, der noch nicht besucht wurde, und setze die Suche bei diesem Nachbarn fort.
- Falls kein solcher Knoten existiert, verfolge die Spuren entlang der Kante zurück, die die Eingangskante zum aktuellen Knoten ist.
- Falls keine solche Kante existiert (z.B. der Knoten ist der erste in der Suchreihenfolge), beende die Suche und notiere, dass alle Knoten von G bereits besucht wurden.

Nun zur Formalisierung und Einbindung dieses Verfahrens in die Optimierung von Ant-Walk-1:

Definition Für $u \in V$ sei: $t_{in}(u) := \min_{v \in N(u), s(v,u)>0}\{s(v,u)\}$ die Zeit des ersten Besuches von u.

Definition Für $u \in V$ sei: $t_{out}(u) := \min_{v \in N(u), s(u,v)>0}\{s(u,v)\}$ die Zeit des ersten Verlassens von u.

Da bei DFS ein Knoten nicht zweimal in der gleichen Richtung besucht wird, verändern sich die Werte von $t_{in}(u)$ und $t_{out}(u)$ nicht mehr, wenn sie einmal gesetzt wurden. Beide Werte werden mit Null initialisiert. Falls $t_{in}(u) = t_{out}(u) = 0$ ist, bedeutet dies also, dass u ein neuer Knoten ist, der noch nicht besucht wurde. Falls $t_{in}(u) > t_{out}(u) > 0$, ist u ein Startknoten, da er zuerst verlassen und dann besucht wurde.

Algorithm 2 depth first search

.	Rule DFS(u vertex)
A)	$t := 1 + t$
B)	Falls ein Nachbar v von u existiert mit $s(u,v) < 1$, dann setze $s(u,v) := t$
.	Falls $t_{in}(v) < 1$, gehe zu v
.	/Gehe ich zum Ursprung zurück?/
C)	Falls $t_{in}(v) > t_{out}(v) > 0$, dann stoppe
.	/Alle Nachbarn sind alt, der Graph ist vollständig besucht/
D)	Finde eine Kante (u,v) mit $s(v,u) = t_{in}(u)$
E)	Setze $s(u,v) := t$
F)	Geh zu v
G)	Beende DFS

Falls die Markierung der Eingangskante aus irgendwelchen Gründen verloren geht, wird das Programm nie zu Ende gebracht. Außerdem ergibt sich die Schwierigkeit, DFS für mehrere Roboter anzuwenden, da ein Roboter der seinen Startpunkt erreicht hat, dort verharrt und nicht weiter arbeitet. Um die Nachteile von multilevel DFS zu umgehen, wird im Fall des Abbruchs von DFS ein neues Suchlevel gestartet. Die Suchlevel werden individuell von jedem Agenten gespeichert. Befindet sich ein Agent r in der Situation, dass er aus seiner Sicht alle Kanten bereits besucht hat, wird der Wert $search - level(r)$ um eins erhöht. Man kann sagen, dass diese Variable die Zeit speichert, wenn die Suche für diesen Agenten neu gestartet wird. Alle bisher besuchten Knoten gelten nur als unbesucht. Gilt also $t_{in}(u) < search - level(r)$, ist dies ein unbesuchten Knoten für den Agenten r. Alle Suchlevel $search - level(r)$ werden mit dem Wert 1 initialisiert.
Die Einführung des Suchlevels bringt folgende Vorteile:

1. Falls ein Agent zum Ausgangsknoten seiner Suche zurückkehrt, stoppt dieser nicht wie in der obigen Version von DFS, sondern startet eine neue Suche, indem er sein Suchlevel um 1 erhöht. Dieser vorher unnützte Agent greift jetzt ein und kommt den anderen Agenten zur Hilfe.

2. Falls sich der Graph G durch Hinzufügen oder Löschen einen Knotens u ändert, kann das Backtracking nicht nochmal angewendet werden.

3. Falls eine Störung des Sensors den Agenten glauben lässt, Punkt 1) wäre wahr, startet er wie in 1) eine neue Suche mit erhöhtem Suchlevel.

Algorithm 3 Ant-Walk-2

.	Rule Ant-Walk-2($search - level(r)$ integer, u vertex)
A)	$t := 1 + t$
B)	Falls ein Nachbar v von u existiert mit $s(u,v) < search - level(r)$, dann setze $s(u,v) := t$
.	Falls $t_{in}(v) < search - level(r)$ gehe zu v
.	/Habe ich das aktuelle Suchlevel ausgereizt?/
C)	Falls $t_{in}(u) > t_{out}(u) \geq search - level(r)$ dann setze $search - level(r) := t$
.	/Alle Nachbarn sind alt! Erhöhe das Suchlevel!/
D)	Finde eine Kante (u,v) mit $s(v,u) = t_{in}(u)$
E)	Setze $s(u,v) := t$
F)	Geh zu v
G)	Beende Ant-Walk-2

Die schon zu Beginn erwähnten Nachteile sind neben der Speicherung des Suchlevels jedes Agenten auch die erhöhte Anzahl von Berechnungen, die durchgeführt werden müssen. Dennoch ist die Zeit, die der mit Ant-Walk-2 gewählte Ansatz benötigt, kürzer als die seines Vorgängers Ant-Walk-1, was die folgende Analyse zeigt.

3.1 Analyse von Ant-Walk-2

Zuerst gilt es zu bemerken, dass für $k = 1$ also einen Agenten Ant-Walk-2 nur eine Variation von DNS ist, die sich durch $2|E|$ nach oben abschätzen lässt (der Roboter startet eine neue Suche mit erhöhtem Suchlevel, dennoch ist das Problem gelöst). Für den Fall $k > 1$ also mehr als ein Agent ist das worst-case Szenario das folgende:
Alle Agenten führen die gleichen Schritte durch und erzeugen k Suchlevel, bevor der Graph komplett besucht wurde.
Ein Beispiel für ein worst-case Szenario ist durch einen langen schmalen Gang gegeben, bei dem alle Roboter nebeneinander oder auf demselben Knoten starten. In diesem Fall ist keine Verbesserung durch den Einsatz von mehreren Agenten zu erwarten, jedoch kann die große Anzahl von eingesetzten Agenten die Effekte durch Störungen minimieren. Es gilt:

Theorem2 k Agenten, die nach Ant-Walk-2 funktionieren und deren Sensoren einen Fehler von maximal α haben, überdecken einen Graphen G mit

n Knoten, deren maximaler Grad Δ ist, in einer Zeit von höchstens:

$$t_k \leq \frac{n\Delta}{2}\lceil\frac{1+\alpha}{k}\rceil$$

4 Vertex-Ant-Walk

Nicht immer ist es notwendig oder gewünscht, alle Kanten eines Graphen G zu besuchen, sondern es reicht, alle Knoten zu besuchen. Für dieses spezialisierte Problem wurde ein neuer Algorithmus entwickelt. Die Markierungen werden hierbei nicht auf den Kanten, sondern auf den Knoten hinterlassen:

Algorithm 4 Vertex-Ant-Walk

.	Rule Vertex-Ant-Walk(u vertex)
A)	$t := 1 + t$
B)	Falls ein Nachbar v von u existiert mit $s(v) = \min_{w \in N(u)} \{s(w)\}$,
.	/Falls mehrere solchen Knoten existieren, triff eine heuristische Entscheidung/
C)	Setze $s(u) := t$
D)	Gehe zu v
E)	Beende vertex-Ant-Walk

4.1 Analyse von Vertex-Ant-Walk

Theoretisch wird eine Laufzeitschranke, die exponentiell zum Durchmesser des Graphen G, ist vorhergesagt. Allerdings glauben die Autoren, dass die worst-case Laufzeitschranke wesentlich größer ist als die Laufzeitschranke für den average-case. Die Simulationen zeigten, dass die Laufzeit des Vertex-Ant-Walk Algorithmus nahe an der Laufzeit des kantenbasierten Algorithmus liegt. Eine obere Laufzeitschranke zeigt das folgende Theorem:

Theorem3 k Agenten, die nach Vertex-Ant-Walk funktionieren, überdecken einen Graphen G mit Durchmesser d und n Knoten, deren maximaler Grad Δ ist in einer Zeit von höchstens:

$$t_k \leq \frac{n\Delta^d}{k}$$

5 Simulationen und Experimente

Im Gegensatz zur den oben theoretisch entwickelten worst-case Laufzeitschranken interessiert uns in der Praxis das Durchschnittslaufzeitverhalten, diese wur-

de anhand von Simulationen untersucht. Es wurden Untersuchungen für verschiedene Störungseinflüsse ($\alpha \in \{0; 10; 20\}$) gemacht. Außerdem wurde die Anzahl von eingesetzten Agenten variiert. Es stellte sich heraus, dass Ant-Walk-1 und Ant-Walk-2 bei Störungseinflüssen besser funktionieren als Vertex-Ant-Walk, während bei störungsfreien Problemen Vertex-Ant-Walk das gegebene Problem schneller löst. Für alle drei Algorithmen ergibt sich ein asymptotisches Verhalten für wachsendes k. Die Simulationen zeigen allerdings auch, dass in manchen Fällen mehr als 50 % der zu überdeckenden Fläche in weniger als 20 % der gesamten Suchzeit überdeckt werden konnten. In einigen Fälle kann es also sinnvoller sein, einen Algorithmus zu verwenden, der einen signifikanten Prozentsatz der zu überdeckenden Fläche geringer Zeit überdeckt.
Zur Zeit arbeiten die Autoren daran, ein programmierbares Fahrzeug zu entwickeln, das mit den entwickelten Algorithmen programmiert wird und so in der Realität mit Hilfe von abgesetzten Spuren in unbekannten Bereichen navigieren soll.

6 Diskussion

Israel A. Wagner, Michael Lindenbaum und Alfred M. Bruckstein haben sich in ihrer wissenschaftlichen Arbeit mit einem sehr interessanten und anwendungsbezogenen Thema beschäftigt. Sie haben einen Algorithmus entwickelt, der ein dynamisches und unbekanntes Areal überdecken kann und die Laufzeit nach oben begrenzt. Die verwendeten Agenten kommunizieren nur über abgelegte Spuren und lösen das Problem selbst unter gewissen Wahrnehmungsstörungen durch Messfehler der Sensoren. Es sollte weiterhin untersucht werden, wie das Speichern und Teilen der gesammelten Daten zur Optimierung der Problemlösung beiträgt.
Wie sooft untersucht die Wissenschaft ein in der Natur auftretendes Phänomen und versucht dieses zu kopieren. Natürlich ist es noch ein großer Schritt von der theoretischen Lösung des gegebenen Problems zur praktischen Umsetzung, dennoch denke ich, dass dies ein Schritt in die richtige Richtung ist.