Game Theory Evolving **WS 2004/2005**
Nina Lehmann, Leo Petrak **Universität Tübingen**
Michael Kaufmann, Klaus Wehrle **19.10.2004**

Game Theory Evolving - Ein Studienprojekt

Begleitend zum Reading Kurs mit dem Buch 'Game Theory Evolving' wollen wir mit Euch eine eigene Spielsituation modellieren, die wir aus unseren eigenen Forschungen heraus entwickelt haben. Im Folgenden werden wir die Problemstellung formalisieren. Eure (und unsere) Aufgabe ist, für das unten vorgestellte Modell geeignete Evolutionsmodelle zu entwickeln, diese zu implementieren und die Ergebnisse zu protokollieren.

1 Die Situation

Wir haben die folgende Situation: Knoten befinden sich in einem geometrischen Raum und sind untereinander verbunden durch ein Netzwerk. Jeder Knoten in diesem Netzwerk möchte möglichst, dass alle anderen Knoten nicht allzu weit von ihm weg sind. Dies kann er natürlich dadurch erreichen, dass er einfach Kanten zu allen anderen Knoten aufbaut, sich also insgesamt eine Clique bildet. Natürlich passiert das in realen Situationen so gut wie nie, und meistens hängt das wiederum damit zusammen, dass sowohl das Aufbauen als auch das Erhalten von Kanten Energie, Geld oder sonstige Resourcen kostet. Gesucht ist also eine Netzwerkstruktur, die mit wenig Kantenkosten versucht, eine gute Erreichbarkeit der Knoten untereinander zu realisieren.

Ein Knoten, der zu weit weg von anderen Knoten ist, wird versuchen sich näher an das Zentrum des Netzwerkes zu bringen, indem er neue Kanten aufbaut. Der Knoten, zu dem dieser unzufriedene Knoten eine Kante aufbauen will, ist unter Umständen schon recht zufrieden mit seiner Situation im Netzwerk und wird die neue Kante unter Umständen nicht annehmen wollen. Überhaupt wird ein Knoten, der keinen allzu großen Abstand zu den anderen Knoten hat, immer versuchen, einige seiner Kanten loszuwerden, weil er dadurch die Kosten senken kann.

Wir werden im Folgenden zuerst einige graphentheoretische Grundbegriffe definieren, und dann das Modell der oben dargestellten Situation konkretisieren.

2 Graphentheoretische Grundbegriffe

Gegeben sei ein k-dimensionaler, euklidischer Raum und eine Menge von Knoten V, die in irgendeiner Weise in diesem Raum verteilt sind. Jeder Knoten v_i hat einen Positionsvektor $\vec{p} = (x_{i1}, x_{i2}, \ldots, x_{ik})$. Die Distanz $C(v_i, v_j)$ zweier Knoten v_i und v_j im Raum ist gegeben durch die euklidische Distanz ihrer Position. Knoten können miteinander verbunden sein, ausgedrückt durch das ungeordnete (v_i, v_j). Diese Verbindungen sind symmetrisch, und die Menge E aller Verbindungen (oder Dupel) ist die Kantenmenge zwischen den Knoten. Somit ergibt sich ein *gewichteter Graph* $G = (V, E, C)$, wobei die Gewichtungsfunktion jedem Dupel (v_i, v_j) die euklidische Distanz zwischen diesen Punkten zuweist. Diese Distanz werden wir im weiteren auch als die *Kosten* einer (potenziellen) Kante bezeichnen, da intuitiv die Kosten einer Kante in einem euklidischen Raum proportional zu der Distanz sind, die sie überbrücken muss.

Ein *Pfad* $W(v_0, v_p)$ von Knoten v_0 zu Knoten v_p in einem Graphen ist definiert als eine geordnete Menge von Knoten $(v_0, v_1, v_2, \ldots, v_p)$ mit den folgenden Eigenschaften: Kein Knoten in dieser Menge kommt doppelt vor und für alle v_i, v_{i+1} ist das Dupel (v_i, v_{i+1}) Element der Kantenmenge. Die Länge eines Pfades entspricht der Anzahl von Kanten, die traversiert müssen, um von v_0 zu v_p zu kommen, in diesem Fall also $p-1$. Ein Pfad ist ein kürzester Pfad zwischen v_0 und v_p, wenn er von allen Pfaden zwischen diesen Knoten die minimale Länge hat. Die minimale Pfadlänge zwischen zwei Knoten v_0 und v_p wird auch als *hopping distance* $d(v_0, v_p)$ bezeichnet. Die *eccentricity* $e(v_i)$ eines Knotens ist definiert als die maximale Distanz die dieser Knoten zu irgendeinem anderen Knoten aus V hat:

$$e(v_i) = \max_{v_j \in V} d(v_i, v_j) \tag{1}$$

Eine Kante (v_i, v_j) ist *inzident* zu ihren beiden Endknoten v_i und v_j. Die beiden Knoten v_i und v_j sind zueinander *adjazent*. Der *Grad* $d(v_i)$ eines Knotens v_i ist definiert als die Anzahl der Kanten, die zu ihm inzident sind. Die Kosten $C(v_i)$ eines Knotens sind definiert als die Summe der Kantenkosten seiner inzidenten Kanten:

$$C(v_i) = \sum_{(v_i, v_j) \in E} C((v_i, v_j)) \tag{2}$$

3 Das Modell

Die "Umwelt" unsere Problemstellung ist gegeben durch den metrischen Raum, indem die Punkte verteilt sind. Das Spiel besteht darin, dass immer wieder Knoten versuchen, ihre Situation zu verbessern, indem sie Knoten zu anderen Knoten entweder aufbauen oder löschen. Das Löschen können sie ohne Kooperation mit dem verbundenen Knoten lösen, aber eine neue Kante können sie nicht ohne die Kooperation des potenziellen Partners aufbauen.

Wir gehen davon aus, dass die Knoten jederzeit ihre Eccentricity berechnen können. Außerdem wissen sie, welcher Knoten wo seine Position hat. Natürlich können sie auch jederzeit ihre Kosten berechnen.

Die Parameter dieser Situation sind also die Folgenden:

a) Die Zufriedenheitsfunktion ist in irgendeiner Weise eine Kopplung der Eccentricity eines Knotens und seiner Gesamtkosten. Dabei gibt es mehrere Möglichkeiten, die alle das Folgende gemeinsam haben: Ein Knoten ist dann zufrieden, wenn sowohl seine Eccentricity als auch seine Kosten klein sind. Dazu kann man beispielsweise beide Werte normieren, indem man die Eccentricity durch n teilt und die Kosten durch $n-1$ teilt. Dadurch erhält man in beiden Fällen nur noch Werte zwischen 0 und 1. Diese beiden Werte können gewichtet addiert werden und ein threshold-Wert kann die Entscheidung bringen, ob der Knoten zufrieden ist oder nicht. Eine andere Möglichkeit ist, dass man die beiden Werte entkoppelt betrachtet. Dazu gibt man je einen threshold-Wert für die beiden Eigenschaften an, die jeweils getrennt voneinander überschritten werden. Wenn der Knoten unzufrieden mit seiner Eccentricity ist, wird er unter Umständen andere Strategien befolgen, als wenn er unzufrieden mit seinen Kosten ist. Fallen Euch noch weitere Ideen ein, wie man die Zufriedenheit des Knotens in dem oben beschriebenen Netzwerk messen kann? Denkt daran, dass diese Zufriedenheitsfunktion geeignet parametrisierbar sein sollte, so dass sie unter Umständen ebenfalls evolvieren kann! Um hier extreme Fälle auszuschließen, setzen wir die gewünschte Eccentricity allerdings auf $\log n$ fest. Ansonsten würden vermutlich alle Knoten ihre gewünschte Eccentricity auf 1 oder gar 0 evolvieren, weil so die Kantenkosten tatsächlich minimal werden.

b) Eine mögliche Aktion eines unzufriedenen Knotens besteht darin, neue Kanten aufzubauen. Dazu sollten wir eine Wahrscheinlichkeitsfunktion auf dem Raum aller möglichen neuen Kanten aufbauen, der geeignet parametrisiert ist. Die Wahrscheinlichkeit, dass der Knoten versucht, eine neue Kante aufzubauen, sollte im wesentlichen umgekehrt proportional zu der Länge der potenziellen Kante sein. Dieser Wert könnte aber auch durch die Unzufriedenheit des Knotens beeinflusst sein. (Je unzufriedener, desto eher ist er bereit, hohe Kantenkosten in Kauf zu nehmen). Z.B. könnte eine solche Funktion die Kanten in zehn verschiedene Intervalle einteilen, so dass alle Kanten mit Länge 0-0.1 in einem Topf landen. Für jeden dieser Töpfe könnte es eine andere Wahrscheinlichkeit geben, die dann im Laufe der Zeit evolviert.

c) Desgleichen brauchen wir eine Wahrscheinlichkeitsfunktion, die ebenfalls in Abhängigkeit von der Länge einer Kante entscheidet, ob sie wieder gelöscht werden soll. Auch hier soll eine geeignete Parametrisierung erfolgen, so dass wir an dieser Stelle evolvieren lassen könnte.

4 Die Aufgabenstellung

Modelle sind meistens umso besser, je einfacher sie sind. Wir wollen von Euch möglichst viele Ideen, wie man die oben umrissene Situation möglichst sinnvoll modellieren kann. Wir werden daraus gemeinsam einige Implementierungen konstruieren, die dann von uns allen in die Tat umgesetzt werden. Die verschiedenen Modelle werden daraufhin untersucht, ob sich das System stabilisiert (dazu Analyse der verschiedenen Populationen, falls Parameter diskret, Untersuchung der Systemeigenschaften wie mittlerer Durchmesser, mittlere Kantenkosten, vielleicht mittlerer Clustering Coefficient etc.). Ergeben sich interessante Netzwerkstrukturen, wie beispielsweise eine skalenfreie Knotengradverteilung? Haben nachher alle Knoten ungefähr dieselbe Kostenakzeptanz, oder ist die Streuung sehr groß?