Deutsch **Facharbeit** **Markus Winand**

Grundlagen der Künstlichen Intelligenz

mit einem Anwendungsbeispiel aus der Industrie

2., überarbeitete Fassung

HTL Wien 10 29. März 2001 Prof. Kirschner

Vorwort

Meines Erachtens ist die künstliche Intelligenz (KI) das aufregendste Anwendungsgebiet der EDV. Leider ist es aber auch eines der komplexesten, wenn nicht gar das komplexeste.

Mich fasziniert die KI Forschung schon sehr lange, und dementsprechend investiere ich auch immer wieder Zeit in das Studium dieses Fachgebietes. Auf diese Weise habe ich über die Jahre ein *Gefühl* für den Begriff „künstliche Intelligenz“ bekommen. Es war auch mein Anliegen dem Leser dieser Arbeit dieses Verständnis zu übermitteln.

Im Rahmen dieser Arbeit ist es natürlich nicht möglich, umfassend auf das Gebiet der KI einzugehen. Vielmehr stellt diese Arbeit eine groben Überblick dar.

In den letzten Jahren wurden neben der KI eine neue Idee geboren: das künstliche Leben. Auf das künstliche Leben (KL) konnte ich leider gar nicht eingehen. Ebenfalls vernachlässigt habe ich das genetische Programmieren (GP), was den genetischen Algorithmen (GA) übergeordnet ist.

Ich hoffe dennoch, dem Leser die Verfahrensweisen der KI näher zu bringen, und auch aufzuzeigen, daß KI keineswegs ein Spielzeug für Uni-Professoren ist, sondern auch durchaus ernstzunehmende Anwendungsmöglichkeiten in der Industrie bietet, was ich durch das Beispiel in Kapitel 3 aufzeige.

Abschließend möchte ich Sie noch bitten, mich über Fehler, Verbesserungsvorschläge oder einfach nur Meinungen unverblümt in Kenntnis zu setzen.

Markus Winand

Inhaltsverzeichnis

1 Intelligenz

An dieser Stelle halte ich es für wichtig ein paar Gedanken zum Begriff „Intelligenz“ aufzuzeigen. Die Definition des Begriffes Intelligenz ist keineswegs eindeutig. Dennoch verbindet jeder Mensch etwas mit diesem Begriff.

Ich versuche zwischen Intelligenz und Wissen zu unterscheiden: Wissen kann von jedem einzelnen erlernt werden, Intelligenz ist jedoch eine angeborene Eigenschaft eines Menschens. Ich bin mir jedoch sicher, daß weder Wissen noch Intelligenz alleine ein Problem lösen kann.

Aber genau hier liegt das Anwendungsgebiet von KI-Systemen, es sollen Probleme gelöst werden. Natürlich ist nicht jede Lösung interessant, sondern nur „gute“ Lösungen. Zu beachten ist, daß tatsächlich eine „gute“, jedoch nicht die „beste“ Lösung gesucht wird. Dies wirkt sich vorallem auf den Lösungsfindungsprozess aus, es werden nicht mehr alle Lösungen benötigt, um die beste selektieren zu können, vielmehr ist ein möglichst kurzer Lösungsweg zu einer befriedigenden Lösung von Interesse. Diese Auffassung von Intelligenz ist auf genetische Algorithmen anwendbar.

Bei Expertensystemen sieht es jedoch anders aus. Ein Schlagwort zur Beschreibung eines Expertensystems ist „knowledge base system“, was bedeutet, daß ein Expertensystem eine Lösung aufgrund von Wissen findet. Ob das Ziehen eines Schlusses wie es heutige Expertensysteme machen, als intelligent bezeichnet werden kann, sei vorerst dahingestellt. Prinzipiell kann zum Verständnis der Intelligenz eines Expertensystems jedoch das Schließen (vorallem das analoge Schließen) als intelligent angesehen werden.

Zur Quantifizierung der Intelligenz wurde der sog. IQ, der Intelligenz Quotient, eingeführt. Da aber die Intelligenz bislang nicht gemessen werden kann, muß sie getestet werden (auf die Probe gestellt werden). Die geläufigen Intelligenztests versuchen das Testobjekt (den Menschen) vor neue Probleme zu stellen, und die Lösung, aber auch den Lösungsweg zu bewerten. An sich scheint die Bewertung des Lösungsweg-es recht einfach: ein schnellerer Weg ist besser. Dies würde bedeuten, daß der IQ von der Zeit zur Lösungsfindung abhängig ist, was jedoch keines wegs stimmt. Wie lange ein Testobjekt benötigt um eine Lösung zu finden, hängt von sehr vielen Erfahrungen (=Wissen) ab. Natürlich versucht man diese Einflüsse durch die Wahl der Probleme zu minimieren, was jedoch äußerst schwer ist.

Weiters ist zu erwähnen, daß die Zeit zum Finden des Lösungsweges nur sehr beschränkt als Maß für die Qualität eines Lösungsweges herangezogen werden kann. Es gibt Systeme (auch Menschen), die schneller als andere arbeiten (denken) können. Wenn man ein KI System betrachtet, wird deutlich, daß die Intelligenz nicht von der Lösungsfindungsdauer abhängen kann, wenn ein und das selbe System auf einem langsameren Computer und einmal auf einem schnelleren ausgeführt wird.

Ein auf Computer anwendbares Kriterium zu Beurteilung von Lösungswegen ist die Anzahl der Iterationen. Näheres dazu im Kapitel 2.

Intelligenz ist jedoch nicht das Thema dieser Arbeit, die eigentliche Frage sollte lauten „Was ist künstliche Intelligenz?“. Der KI-Forscher **Patrick Henry Winston** definiert Künstliche Intelligenz folgendermaßen: “Artificial intelligence is the study of ideas which enable computers to do the things that make people seem intelligent.”

2 Fuzzy Logik

2.1 Einleitung

Will man Intelligenz auf EDV Systemen simulieren, stößt man zunächst auf das Problem, daß Computer prinzipiell exakt arbeiten. Diese Tatsache gilt allgemein als Stärke von Computern, da Computer daher keine Fehler machen. Im Rahmen der KI ist dies jedoch keineswegs ein Vorteil, da viele Aussagen nicht exakt formuliert sind.

Eine sehr unscharfe Aussage ist z.B.: „Es ist heiß“. Es ist offensichtlich daß der Begriff „heiß“ nicht klar definiert ist. Die Interpretation dieses Begriffes hängt nicht nur vom Empfänger ab, sondern von vielen weiteren Faktoren wie z.B. der Jahreszeit oder vom Aufenthaltsort ab.

Aufgabe der Fuzzy[1] Logik ist es, den Computer den Umgang mit solchen unscharfen Aussagen zu lehren.

2.2 Funktionsweise

Zur Erläuterung der Funktionsweise von „unscharfen“ Systemen soll weiterhin das Beispiel des Begriffes „heiß“ dienen.

Es ist offensichtlich, daß diesem Begriff keine eindeutige Temperatur wie z.B. 60 Grad Celsius zugeordnet werden kann. Vorallem ist die richtige Interpretation des Begriffes vom Kontext abhängig. So meint „heißer Kaffee“ eine andere Temperatur als „die Herdplatte ist heiß“. Diese Unterscheidung ist jedoch nicht Aufgabe der Fuzzy Logik, vielmehr geht es um die subjektive Interpretation einer Aussage wie z.B. „der Kaffee ist heiß“. Daß selbst diese Aussage noch interpretierbar ist, ist das Problem, das die Fuzzy Logik zu lösen versucht.

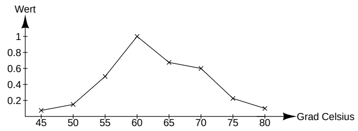
Fuzzy Sets

Hierzu wird keine eindeutige Zuordnung zu einer Temperatur gemacht, sondern eine Wahrscheinlicheitstabelle[2] verwendet. In dieser Tabelle wird z.B. zu jeder Temperatur notiert mit welcher Wahrscheinlichkeit dieser Wert gemeint ist. Diese Tabelle könnte vereinfacht so aussehen (willkürlich gewählter Werte):

[1] Fuzzy(engl.)=Verschwommen.

[2] Der Vergleich der Fuzzy Set Theorie mit der Wahrscheinlichkeitstheorie soll veranschaulichen, wie ein Fuzzy Set aufgefaßt werden kann. Es gibt dennoch grundlegende Unterschiede zwischen der Wahrscheinlichkeitsrechnung und dem Rechnen mit unscharfen Zahlen. In der Fachwelt ist diesbezüglich eine Diskussion entstanden, welche beider Theorien die allgemeinere ist.

Grad Celsius	**Wahrscheinlichkeit**	**Wert**
45	2%	0.066
50	5%	0.166
55	15%	0.500
60	30%	1.000
65	20%	0.666
70	18%	0.600
75	7%	0.233
80	3%	0.100



In der Realität wird diese Wahrscheinlikeit normiert verwendet und *Zugehörigkeitsgrad* genannt. (Spalte *Wert* der Tabelle). Diese normierte Wahrscheinlichkeit ist rechts neben der Tabelle als Graphik dargestellt.[3] Da das Rechnen mit solchen Fuzzy Sets ungleich aufwendiger ist als mit scharfen Zahlen, versucht man ein Fuzzy-Set so einfach wie möglich zu halten, indem man eine Trapezfunktion annähert.

Operatoren

Zum Umgang mit solchen *Fuzzy Sets* hat man Operatoren eingeführt, die auf solche unscharfe Werte angewendet werden können. So gibt es z.B. *UND* und *ODER* Operatoren, die auf ein Fuzzy Set angewendet werden können und wieder ein Fuzzy-Set ergeben. Auf diese Weise können einfache Werte wie z.B. „heiß UND kalt" berechnet werden.

Durch weitere Operatoren können auch sehr komplexe Formeln berechnet werden, die jedoch zum grundlegenden Verständnis nicht notwendig sind.

Die vorgestellten Operatoren UND und ODER sind jedoch wenig einleuchtend, wenn bedacht wird, daß es sich bei einem Fuzzy-Set um eine Zahl, einen einzelnen Wert handelt. Wesentlich einleuchtender sind die üblichen algebraischen Operationen wie „+", „-", „*" und „/". Zur Anwendung dieser algebraischen Operatoren auf ein Fuzzy-Set wurde von Prof. Zadeh 1973 das *Erweiterungsprinzip*[4] vorgestellt.

Durch diese Erweiterung ist es bereits möglich fehlerbehaftete Systeme auf einfache Weise zu berechnen. Z.B. kann eine elektronische Schaltung unter Berücksichtigung der Bauteiltoleranzen berechnet werden. Durch dieses Beispiel wird die Analogie zu bekannten mathematischen Vorgehensweisen

[3] Im Beispiel sind die Tabellen Werte linear verbunden. Bessere Ergebnisse würde z.B. die Annäherung einer Gaußschen Verteilung(=Glockenkurve) bringen, welche aus der Wahrscheinlichkeitsrechnung bekannt ist.

[4] Das Erweiterungsprinzip gilt als Meilenstein in der Fuzzy Set Theorie und bildet die Grundlage vieler Anwendungen.

besonders deutlich, da dieses Beispiel auch klassisch durch die Fehlerrechnung (bzw. Wahrscheinlichkeitsrechnung) gelöst werden kann.

Prädikatenlogik

Der nächste logische Schritt ist es nun, die Fuzzy-Logik auf die *Prädikatenlogik* anzuwenden. Genaueres über die Prädikatenlogik wird im Kapitel 4 besprochen.

Dieser Teil der Fuzzy Logik ist äußerst komplex, und kann hier leider nicht behandelt werden, es ist jedoch für das Verständnis des Kapitels Expertensysteme gut zu wissen, daß auch dort die Fuzzy Logik zum Einsatz kommt.

Mustererkennung

Eines der wichtigsten Einsatzgebiete der Fuzzy Logik ist sicherlich die Mustererkennung. Spezielle Anwendungsgebiete dafür sind die Spracherkennung, Handschrifterkennung aber auch komplexere Dinge wie z.B. Aktienkursanalysen.

Besonders gut kann man sich die Handschrifterkennung mittels eines Fuzzy Sets vorstellen: Man definiert einen Buchstaben nicht mehr als scharf begrenzte Fläche[5], sondern als unscharfes Gebilde mit Grauzonen(=Zugehörigkeitswerte kleiner 1). Mit den vorgestellten Methoden kann nun das unscharfe Referenzobjekt mit dem scharfen Testobjekt verknüpft werden. Das unscharfe Ergebnis kann zu einer Wahrscheinlichkeitsanalyse herangezogen werden.

2.3 Fehlertoleranz

Abschließend will ich noch einmal hervorheben, welche Aufgabe die Fuzzy Logik in der KI, vorallem bei genetischen Algorithmen, hat. Beim Beispiel der Handschrifterkennung ist es sehr deutlich, daß das Arbeiten mit unscharfen Zahlen eine Fehlertoleranz mit sich bringt, die in der KI unerläßlich ist.

Im nächsten Kapitel wird aufgezeigt, daß es nicht nur wichtig ist Eingaben als Fehlerbehaftet anzusehen, sondern auch Ausgaben mit einem Fehler zu beaufschlagen. In der Tat bin ich der Meinung, daß Intelligenz ohne Fehler nicht möglich ist.

[5]Bei der klassischen OCR=(Optical Character Recognition) wird ein Buchstabe als Fläche definiert, und mit einem Testobjekt mittels eines Äquivalenzoperators verglichen. Ein Buchstabe gilt dann als erkannt, wenn die Abweichung einen gewissen Prozentsatz der Buchstabenfläche nicht überschreitet.

Das Wort „Fehler“ ist in der Umgangssprache sehr negativ vorbelastet. Vielleicht beschreiben die Begriffe „Vagheit“ oder „Unsicherheit“ besser, was in der KI oft als „Fehler“ bezeichnet wird. Dieses Kapitel sollte auch aufzeigen, daß jede Aussage als unscharfe Aussage betrachtet werden sollte, also als fehlerbehaftete Aussage. Was jedoch nicht heißen soll, daß eine fehlerbehaftete Aussage eine falsche Aussage ist.

Im Laufe meiner Recherchen zur Fuzzy Logik, und nach langem Überlegen bin ich zu dem Schluß gekommen, daß **erst ein Fehler die Variation bekannter Lösungswege ermöglicht** und daß so neue Lösungen gefunden werden können.

3 Genetische Algorithmen

3.1 Einleitung

Genetische Algorithmen stellen einen wichtigen Teil der KI-Forschung dar. Sie versuchen, wie der Name schon erahnen läßt, Intelligenz am Beispiel der Evolution nachzuahmen. Dabei werden „Lebewesen“ erschaffen, die in ihrer Umwelt überleben müssen, um ihr Erbmaterial (Gene) an die folgende Generation weiterzugeben. Es werden jedoch nicht nur einzelne Lebewesen, sondern ganze Generationen betrachtet, wovon das Erbmaterial des erfolgreichsten Lebewesens die Lösung enthält.

Die Anwendung von genetischen Algorithmen ist dann sinnvoll, wenn es darum geht, binnen kurzer Zeit eine gute Lösung zu erhalten. Die Stärke von genetischen Algorithmen liegt also bei Problemen, wo eine „Straight Forward“ Strategie[6] zu ressourcenaufwendig wird.

Am Ende dieses Kapitels werde ich auf ein praktisches Anwendungsbeispiel genauer eingehen, um zu zeigen, daß KI nicht nur in den Universitäten zuhause ist.

3.2 Gene und Chromosome

Im Sinne eines genetischen Algorithmuses (GA) stellt ein *Gen* die Beschreibung einer Eigenschaft eines Lebewesens dar. So könnte z.B. ein Gen angeben, welche Farbe ein Lebewesen hat. Ein Lebewesen kann beliebig viele Gene haben, die Anzahl ist in der Regel von der Komplexität des Problems abhängig.

Der gesamte Gencode eines Lebewesens wird als *Chromosom* bezeichnet, und kann jederzeit entschlüsselt werden. Die Bedeutung eines Gens ist durch seine Position im Gencode vorgegeben. Jedes Lebewesen hat seinen eigenen, individuellen Gencode, das es von anderen Lebewesen unterscheidet.

Anhand des Gencodes kann ermittelt werden, wie „fit“ ein Lebewesen ist. Diese Aufgabe übernimmt die sogenannte *Bewertungsfunktion*, welche durch Decodierung des Gencodes einen Wert proportional zur „fitnes“ oder „Überlebensfähigkeit“ des Lebewesens ermittelt.

[6] Bei dieser Strategie wird organisiert nach jeder Lösung gesucht und die beste selektiert.

3.3 Generationen

Eine Generation von Lebewesen sind mehrere Lebewesen, die in einer gemeinsamen Umwelt[7] leben. Jedes in dieser Generation lebende Lebewesen kann sich *fortpflanzen*, wodurch ein Lebewesen einer neuen Generation entsteht. Bevor jedoch auf die Fortpflanzungsmechanismen von genetischen Algorithmen eingegangen wird, möchte ich erwähnen, daß man der Einfachheit halber immer nur eine Generation leben läßt. Das heißt die Elterngeneration geht in die Nachfolgergeneration über, doch kostet dieser Vorgang die Eltern meist das Leben. Um die Population[8] konstant zu halten, wurden zusätzliche Fortpflanzungsmöglichkeiten eingeführt.

3.4 Fortpflanzung

Wie auch im richtigen Leben hat in der KI ein Lebewesen die Möglichkeit sich fortzupflanzen. Dabei wird der Gencode (oder Teile davon) an die nächste Generation, die Kinder, weitergegeben. Da jedoch keine Kopie der Elterngeneration enstehen soll, wird die Genübergabe gewissen Mechanismen unterworfen:

Mutation

Durch eine zufällige Veränderung einer kleinen Anzahl von Genen ensteht eine Mutation eines Elternlebewesens. Bei dieser zufälligen Veränderung ist nicht bekannt, welche Bedeutung ein Gen hat.[9] Da nur eine geringe Änderung am Gencode des Vorfahrens vorgenommen wird, ist das enstehende Lebewesen meist ähnlich „fit" wie der Vorfahre.

Durch Mutation kann eine gänzlich neue Eigenschaft entstehen, so kann z.B. aus einem blauen Lebewesen ein violettes werden.

Kombination

Dies ist die beim Menschen üblichen Form der Fortpflanzung. Durch die Kombination mehrerer Gencodes entsteht ein neuer. Es ist sinnvoll (nach dem Vorbild der Natur) die Anzahl der Eltern auf zwei zu fixieren, da die

[7] In diesem Zusammenhang ist auch der Begriff *Universum* üblich, womit ausgedrückt wird, daß alle Lebewesen den gleichen (physikalischen) Gesetzen unterliegen.

[8] Unter Population versteht man die Anzahl der gleichzeitig lebenden Lebewesen eines Universums

[9] Es scheint so, als wäre es effizienter, die Bedeutung der Gene zu beachten um besonders die negativen Effekte zu variieren. Dies führt aber dazu das sich der Algorithmus nicht mehr aus lokalen Maxima[14] herausbewegen kann.

Möglichkeit der Einbringung eines weiteren Gencodes in der nächsten Generation besteht.[10]

Bei der Kombination von Gencodes werden Teile der Elterngencodes unverändert übernommen, ohne dabei die Bedeutung der übernommenen Gene zu ändern.[11] Verschmelzungen von einzelnen Genen sind hierbei nicht üblich. Wenn die Vorfahren also die Farben Grün und Rot haben, kann das Kind nur grün oder rot sein, nicht jedoch blau.

Wieviele Nachkommen ein Elternpaar auf diese Weise erzeugt, kann prinzipiell verschieden sein, der Einfachheit halber wird jedoch meist ein, maximal jedoch zwei Nachkommen pro Elternpaar erzeugt.

Um zu verhindern, daß durch die Kombination von Gencodes ein ähnlicher oder gar gleicher Gencode ensteht, paaren sich bevorzugt Elternpaare mit möglichst unterschiedlichen Chromosomen. Der Effekt der *Inzest* kann durch besondere Wahl der Binären Codierung der Gene nachempfunden werden.[12]

Spontanes Sterben eines Lebewesens

Ein Lebewesen kann, bevor es seinen Gencode durch Mutation oder Kombination weiter gibt, sterben. Die Entscheidung über Leben und Tod fällt im Falle eines GAs die Bewertungsfunktion (siehe unten) . In der Natur kommt es jedoch auch vor, daß ein Lebewesen durch höhere Gewalt (z.B. Blitzschlag) zu Tode kommt. Heute ist es noch nicht klar, was für einen Einfluß das zufällige Töten eines Lebewesens auf die Gesamtlösung hat. Da der gesunde Menschenverstand keinen Grund zum Töten eines sehr gesunden Lebewesens bietet, wird auf diese höher Gewalt in der Simulation meist verzichtet.

Spontanes entstehen neuer Lebewesen

Zur Konstanthaltung der Population eines Volkes, können Lebewesen bei Bedarf auch spontan entstehen. Da diese Lebewesen keine Vorfahren haben, wird Ihr Gencode zufällig ermittelt.

Wie Ihnen sicher schon aufgefallen ist, werden die Fortpflanzungsformen der

[10] So kann ein Lebewesen nach zwei Generationen Fragmente von bis zu vier Gencodes enthalten.

[11] Wenn z.B. die erste Hälfte des Gencodes vom Vorfahren A übernommen wird, muß dieses Gencodefragment bei dem neuen Lebewesen auch an erster Stelle stehen.

[12] Auf diesen Effekt kann im Rahmen dieser Arbeit leider nicht genauer eingegangen werden. Zum genauen Verständnis dieses Effektes ist tiefgehendes Wissen über binäre Codierungsformen notwendig.

Mutation und Kombination in der Natur vereint. Bei genetischen Algorithmen versucht man jedoch die Fortpflanzungsmechanismen so einfach wie möglich zu halten, und nimmt dafür eine häufigere Fortpflanzung in Kauf.[13]

Die Anzahl der Fortpflanzungen bis zu einer befriedigenden Lösung wird auch Anzahl der *Iterationen* genannt, und kann zur Bewertung eines Lösungsweges herangezogen werden (siehe Kapitel 1).

Bei den oben angeführten Fortpflanzungsmechanismen spielt der *Zufall* eine sehr große Rolle. Hier kann also die Fuzzy Logik zum Einsatz kommen. Es ist bei diesen Fortpflanzungsmöglichkeiten ungleich wichtiger, daß ein Fehler bewußt (aber zufällig) erzeugt wird, als daß ein Wert fehlertolerant interpretiert wird.

Damit jedoch ein zufälliger Fehler entstehen kann, werden Zufallsgeneratoren benötigt, welche unter ähnlichen Ausgangsbedinungen sehr verschiedene Werte liefern. Diese Generatoren sollten also ein *chaotisches* Verhalten aufweisen, damit ein genetischer Algorithmus bei verschiedenen Durchgängen auch verschiedene Ergebnisse liefert.

In die Entwicklung von Zufallsgeneratoren wird ein nicht unerheblicher Aufwand investiert, da sie einen großen Einfluß auf das Verhalten eines genetischen Algorithmuses haben.

3.5 Bewertungsfunktion

Aufgabe der Bewertungsfunktion ist es, zu ermitteln, wie „gesund“ ein Lebewesen ist. Diese Funktion wird in der Natur „natürliche Selektion“ genannt. In der Regel liegt bei der Erstellung dieser Funktion die Schwierigkeit eines genetischen Algorithmusses. Durch diese Funktion wird die „Ideallösung“ beschrieben.

Die Funktion muß gewünschte Eigenschaften positiv und unerwünschte negativ bewerten. Dabei findet eine Gewichtung der Eigenschaften statt, um besonders positive Eigenschaft hervorzuheben.

Diese Funktion entscheidet auch darüber, ob ein Lebewesen überlebens- bzw. fortpflanzungsfähig ist oder nicht.

3.6 Bemerkungen

Wie oben gezeigt, verfällt man bei der Realisierung genetischer Algorithmen leicht in moralische Bedenken (spontanes Töten eines Lebewesens). Der Sinn eines GAs ist jedoch, die sehr erfolgreiche Natur nachzuahmen. Das spontane

[13] Durch den Verzicht auf zu komplexe Fortpflanzungsmechanismen kann der Entwicklungsaufwand stark gesenkt werden, ohne qualitative Einbußen in Kauf nehmen zu müssen.

Töten eines Lebewesens kann durchaus sinnvoll sein, wenn dadurch ein neues Lebewesen spontan entstehen kann, welches „frisches Blut“ in die Generation bringt. Das spontane Töten bzw. Entstehen von Lebewesen kann z.B. effektiv verhindern, daß sich ein Volk zu lange in einem *lokalen Maximum*[14] aufhält oder dort gar hängen bleibt.

3.7 Zusammenarbeit

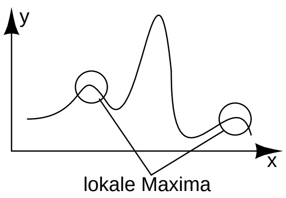
Wenn die oben beschriebenen Module realisiert wurden, ist es leicht daraus einen genetischen Algorithmus zu erstellen:

1. Zufälliges Erstellen einer Population.
2. Decodierung der Chromosomen.
3. Berechnung der „Gesundheit“ der Lebewesen.
4. Erstellen einer neuen Generation durch die oben beschriebenen Fortpflanzungsmethoden. Hierfür werden nur die gesündesten Lebewesen herangezogen, die „kranken“ Lebewesen sterben aus. Zur Aufrechterhaltung der Population können spontan Lebewesen entstehen.
5. Wiederholen der Punkte 2 bis 4, bis eine Abbruchbedingung erfüllt ist, d.h. bis eine befriedigende Lösung gefunden wurde.

3.8 Beispiel

Durch meine Arbeit beim Österreichischen Internet Provider „magnet - Internet At Work“[15] kann ich Ihnen im folgenden eine Beispiel aus der Praxis erläutern.

---

[14] 

Ein lokales Maximum liegt dann vor, wenn durch kleine Variationen keine Steigerung mehr gefunden werden kann.

[15] **http://www.magnet.at/** so viel Werbung sei magnet gegönnt.

Problem

Durch steigende Userzahlen konnte der Datenhighway der PTA[16] den quantitativen und qualitativen Ansprüchen von magnet nicht mehr genügen. Daher wurde entschieden, daß magnet über kurz oder lang eine eigene österreichweite Infrastruktur aufbauen muß. Diese Infrastruktur enthält sog. POPs[17], welche möglichst kosteneffizient über das Bundesgebiet verteilt werden sollten. Die Anzahl der POPs sowie die Standorte dieser waren gesucht.

Eine Generation als Lösung

Nachdem klar war, daß eine „Straight Forward" Lösung zuviel Zeit in Anspruch nehmen würde, wurde die Idee eines GAs aufgegriffen.

Ursprünglich war geplant, ein Lebewesen zu definieren, das als Gene die Standorte der einzelnen POPs enthält. Dieser Ansatz mußte jedoch bald verworfen werden, da die Anzahl der POPs im vorhinein nicht bekannt war, die Anzahl der Gene eines Lebewesens jedoch unveränderlich ist.

So lag der Schluß nahe, nicht ein Lebewesen als Lösung anzusehen, sondern vielmehr ein Lebewesen als POP zu definieren und eine Generation von Lebewesen als Lösung zu interpretieren.

Lebewesen

Ein POP entspricht daher einem Lebewesen mit nur einem Gen: dem Standort. Aus diesem Gen leiten sich weitere Eigenschaften des POPs unmittelbar ab. Diese Eigenschaften sind vorallem die Kosten (für Hardware, Standleitung und Wartung) und der Nutzen (die Anzahl der User im Einzugsgebiet des POPs).

Fortpflanzung

Auf einen POP als Lebewesen können folgende Fortpflanzungsmethoden angewandt werden:

- Mutation

  Verschieben des Standorts des POPs.

[16]Durch den sog. Highway-194 der PTA (=Post und Telekom Austria) konnte magnet schon vor Jahren die Österreich weite einwahl zum Onlinetarif anbieten, ohne eine Infrastruktur aufbauen zu müssen.

[17]POP = Point of Presence, Einwahlknoten. Die 50km Zone um einen Einwahlknotetn wird „Einzugsbereich" genannt, da von dort aus die Einwahl zum ermäßigten Onlinetarif möglich ist.

- Spontanes Töten eines Lebewesens
  Reduktion der Anzahl der POPs.
- Spontanes Entstehen eines Lebewesens
  Erhöhung der Anzahl der POPs.

Die Kombination ist nicht möglich, da jedes Lebewesen nur ein Gen hat.

Bewertungsfunktion

Da die Bewertungsfunktion bei diesem Problem enorm komplex ist, möchte ich an dieser Stelle nur andeuten, welche Faktoren eine Rolle spielen.

Durch den Standort eines Lebewesens ergibt sich unmittelbar die Länge der Standleitung, die Kosten der Standleitung sind jedoch nicht nur von der Länge abhängig, sondern auch von der Bandbreite, welche wiederum von der Anzahl der User im Einzugsgebiet des POPs abhängt. Wieviele User im Einzugsgebiet eines POPs sind, ist von den Wohnorten der User, aber auch vom nächstgelegenen POP abhängig.

Es ist deutlich, daß die Bewertungsfunktion die ganze Generation beurteilen muß und nicht nur ein einzelnes Lebewesen.

Weiters muß in der Bewertungsfunktion eine Strafe für nicht erreichte User eingebaut werden. Da magnet durch die Umstellung keine User(=Kunden!) verlieren will, mußte der Highway für User, die nicht im Einzugsbereich der magnet Infrastruktur sind, weiterhin bestehen. Für jede Generation mit nur einem nicht erreichten User fallen Fixkosten für den Highway-194, sowie weitere pro-User Kosten an.

Lösung

Die von diesem genetischen Algorithmus erbrachten Lösungen waren ausgesprochen gut, um nicht zu sagen beeindruckend.

Nach Angaben der PTA Bestand die kosteneffizienteste Lösung für eine flächendeckende Infrastruktur aus 24 POPs. Der genetische Algorithmus lieferte nach wenigen Tage eine günstigere Lösung mit 18 POPs.

Bei genauer Betrachtung diese Lösung war auffällig, daß der Algorithmus den Firmensitz in Wien einfach wegrationalisiert hat. Dies ist durch die Arbeitsweise der PTA zu begründen. Das Wiener Ortsnetz liegt zwar im 50km Radius vom Ortskern Eisenstadts, nicht jedoch umgekehrt. Das Problem konnte durch eine Bestrafung gelöst werden, falls in einer Generation kein Lebewesen in Wien ist. Diese Strategie war sogar sehr erfolgreich, so fanden sich POPs, die mitten im nirgendwo plaziert waren, aber dennoch viele User im Einzugsbereich hatten.

4 Expertensysteme

4.1 Einleitung

Expertensysteme sind sogenannte „knowledge base systems“. Das bedeutet, daß sie auf *Wissen* basieren. Einem Expertensystem können bestimmte Sachverhalte (Wissen) eingegeben werden. Aufgrund dieses Wissens kann es dann Fragen beantworten.

Der Name Expertensystem kommt daher, daß diese Software das Wissen eines Fachmanns (=Expertens) haben soll, und dadurch einen ähnlichen Entscheidungsfindungsprozeß ausführen kann.

4.2 Wissensrepräsentation

Die Hauptschwierigkeit eines Expertensystems liegt darin, das Wissen[18] in einer einheitlichen Form zu präsentieren. Eine einfache Form dieser Darstellung ist die sogenannte *Aussagenlogik*. Die Weiterentwicklung der Aussagenlogik ist die *Prädikatenlogik*, welche auf die Satzstruktur eingeht. Die Prädikatenlogik ist ungleich komplexer als die Aussagenlogik, ist jedoch die auf Expertensystemen eingesetzte Logik.

Im Rahmen dieser Arbeit kann ich nicht tiefer auf diese beiden Fachgebiete eingehen, will jedoch einen kurzen Einblick in die Aussagenlogik liefern.

Aussagenlogik

Die klassische Aussagenlogik arbeitet mit Boolschen Ausdrücken[19]. Die Anwendung der Fuzzy Logik ist jedoch äußerst sinnvoll und wird daher auch angewandt. Zum besseren Verständnis werde ich meine Erläuterungen ohne die Anwendung von Fuzzy Logik durchführen.

Die Aussagenlogik stellt Verbindungen zwischen einzelnen Aussagen her. Eine Aussage kann z.B. sein „es regnet“. Zur kürzeren Schreibweise werden solchen Aussagen symbolische Namen, wie man es von der Mathematik her kennt, zugewiesen.

[18] In der Informatik unterscheidet man zwischen Daten, Informationen und Wissen. Daten sind Abfolgen von Bytes, von denen die Bedeutung jedoch unbekannt ist. Information sind Daten, deren Bedeutung bekannt ist. Wissen ist Information über logische Zusammenhänge. Man beachte hierbei alltägliche linguistische Formen: „das weiß ich“ drückt aus, daß man einen Zusammenhang kennt.

[19] Werte mit nur zwei Zuständen: Wahr oder Falsch.

Name	**Bedeutung**
A	„es regnet“
B	„es fällt Wasser vom Himmel“
C	„die Sonne scheint“

Auf eine (oder mehrerer) solche Aussagen können nun bestimmte Operatoren angewandt werden (Auszug):

- Negation

  Der einfachste Operator ist die Negation ($\neg$). Sie bewirkt die Inversion des Wahrheitsgehaltes einer Aussage.

  So bedeutet die Aussage „$\neg$A“ soviel wie „es regnet nicht“.

- Konjunktion

  Der zweistellige Operator der Konjunktion ($\wedge$) verbindet zwei Aussagen mittels eines UND Verknüpfung. Das heißt die Gesamtaussage gilt nur dann als wahr wenn beide Teilaussagen wahr sind.

  Die Aussage „A $\wedge$ C“ heißt „es regnet und die Sonne scheint“ erscheint etwas merkwürdig. Diese Aussage kann zu keiner Situation stimmen.[20] Solche immer falschen Aussagen nennt man *Kontradiktion*.

- Disjunktion

  Die Disjunktion ($\vee$) verbindet zwei Aussagen mit einer ODER Verknüpfung. Die Gesamtaussage ist also dann wahr, wenn eine der beiden Teilaussagen Wahr ist.

  Das Beispiel „A $\vee$ B“ stellt genauso wie das vorherige Beispiel einen Sonderfall dar. Eine Aussage, die immer richtig ist wird *Tautologie* genannt.

- Implikation

  Dieser binäre Operator ($\rightarrow$) kann am einfachsten als *Folgerung* verstanden werden. Die Aussage „A $\rightarrow$ B“ kann als „aus A folgt B“ gesprochen werden, und bedeutet, daß wenn A wahr ist, auch B wahr sein muß. Das obige Beispiel stellt wieder eine Tautologie dar.

  Ein interessanteres Beispiel wäre jedoch „A $\rightarrow$ $\neg$ C“ was soviel bedeutet wie „Wenn es regnet, scheint die Sonne nicht“.

[20] Der Sonderfall, daß der Sonneneinfallswinkel gering genug ist, um unter den Wolken durchzuscheinen, wird hierbei nicht berücksichtigt.

Ich möchte an dieser Stelle nicht weiter auf die Wissensrepräsentation eingehen, da dies zum grundlegenden Verständnis von Expertensystemen nicht notwendig ist.

4.3 Die „knowledge base“

Das Erstellen der Wissensbasis ist das Hauptproblem von Expertensystemen. Durch die Prädikatenlogik müssen die einzelnen Aussagen nicht mehr händisch zerlegt werden, um eingegeben werden zu können. Dennoch ist bei der Wisseneingabe sehr sorgfältig umzugehen, da es sonst schnell zu falschen Schlüssen kommen kann.

Die technische Realisierung wird über ein Datenbank ähnliches System durchgeführt. Da jedoch keine Daten, sondern Wissen gespeichert werden soll, müssen die bekannten Datenbank-Systeme erweitert werden. Eine solche Wissensbank kann nachdem sie entworfen wurde, für verschiedene Expertensysteme verwendet werden, da durch die Aussagenlogik alles Wissen in kleinste Zusammenhänge zerlegt werden kann, welche bei jedem Fachgebiet anwendbar sind.

Eine solche allgemein verwendbare Wissensbanksoftware ist PROLOG. Diese Programmiersprache unterstützt die Aussagenlogik und erlaubt es, direkte Fragen zu formulieren.

Wenn eine Wissensbank eine gewisse Größe erreicht hat, kommt es leicht vor, daß das System Antworten auf Fragen findet, welche für einen menschlichen Experten verwirrend erscheinen.

4.4 Schließen

Die Aufgabe eines Expertensystems ist es, Schlüsse zu ziehen. Diese Schlüsse werden anhand von Wissen aus der oben vorgestellten Wissensbank gezogen. Man kann das Schließen(Lernen) in drei wichtige Grundtypen unterteilen:

1. das *deduktive Schließen* Das deduktive Schließen ist das Schließen von allgemeinen Sachverhalten auf spezielle Fälle.

   Aus den Aussagen „ein Schwein hat vier Beine“ und „Babe ist ein Schwein“ kann die Aussage „Babe hat vier Beine“ durch deduktives Schließen abgeleitet werden.

   Diese Form des Schließens ist die einfachste, aber auch die wichtigste Form des Schließens. Ein Expertensystem kann solche Schlüsse durchführen.

2. das *induktive Schließen* Beim induktiven Schließen wird ein allgemeinerer Sachverhalt aus beobachteten, spezielleren Sachverhalten abgeleitet.

   Dies ist das Lernen aus Beispielen. Aus den Aussagen „Babe ist ein Schwein und hat vier Beine„ und „Rudi ist ein Schwein und hat vier Beine“ kann induktiv auf „Schweine haben vier Beine“ geschlossen werden.[21]

   Diese Form des Schließens ist bei Expertensystemen teilweise vorhanden.

3. das *analoge Schließen*

   Dabei werden bei unbekannten Sachverhalten Schlüsse aus Analogien gezogen.

   Es könnte z.B. durch die Analogie bei Fachwörtern von genetischen Algorithmen auf die Vererbungslehre geschlossen werden.

   Diese Form des Schließens ist bei Expertensystemen heute nicht vorhanden. Zum einen wäre Wissen über mehrere Fachgebiete notwendig, zum anderen müßten sehr Fehlertolerante Verfahrensweisen verwendet werden, welche heute einfach zu langsam sind.

4.5 Fragen

Wenn ein Expertensystem im Zuge des Schließens auf eine zu hohe „Unsicherheit“ stößt, besteht die Möglichkeit, einem Menschen eine Frage zu stellen, um diese Unsicherheit zu eliminieren. Wenn z.B. die Information, daß ein Schwein vier Beine hat, induktiv ermittelt wurde, liegt bei diesem Wissen eine hohe Vagheit vor. Nach der Frage „wieviele Beine hat Rudi-Rüssel, wenn er ein Schwein ist“, könnte diese Vagheit durch die Frage „haben Schweine vier Beine?“ eliminiert werden.

Durch dieses Konzept kann bei sehr schwierigen Fragen dennoch sehr elementares Wissen abgefragt werden.

[21] Die Anzahl der Beispiele, die zu einem induktiven Schluß führen, wirken sich darauf aus, mit welcher Wahrscheinlichkeit der Schluß korrekt ist. Diese Unsicherheit kann durch Anwendung der Fuzzy Logik berücksichtigt werden. Das Ergebnis des Schlusses könnte dann lauten: „viele Schweine haben vier Beine“. An dieser Stelle will ich auch erwähnen, daß die Doppeldeutigkeit linguistischer Begriffe in keiner Weise von Expertensystemen unterschieden werden können. Die Aussagen „Thomas hat zwei Beine“ und „Thomas ist ein Schwein“ würden dazu führen, daß das Expertensystem die Wahrscheinlichkeit, daß ein Schwein zwei Beine hat, erhöhen. Wenn dann jedoch noch die Information, daß „Thomas ein Mensch ist“ eingegeben wird, liegt eine Kontradiktion vor, da Thomas nicht ein Schwein und ein Mensch gleichzeitig sein kann.Auch solche Situationen können durch Fuzzy Logik umgangen werden.

4.6 Zusammenfassung

Ein Expertensystem kann also durch eine Wissensbank und durch Zuhilfenahme der Prädikatenlogik Fragen eines Fachgebietes beantworten. Expertensysteme sind heute schon für viele Fachgebiete verfügbar und im Einsatz.

Als allgemein bekanntes Expertensystem (das jedoch noch nicht in Betrieb ist) möchte ich den Schiffscomputer der U.S.S.[22] Enterprise von Star Trek anführen. Dieses Beispiel ist zwar heute noch nicht realisiert, kann aber recht deutlich aufzeigen, wozu ein Expertensystem mit einer entsprechenden Wissensbasis im Stande ist.

Bevor Sie das nächste mal „so ein Blödsinn!“ behaupten wenn dem Schiffscomputer die Frage „Ist es möglich einen Sprung durch die Zeit durchzuführen?“ gestellt wird, sollten sie sich vor Augen halten, daß ein Expertensystem solche Fragestellungen prinzipiell verarbeiten kann. Natürlich ist es bis zur Realisierung solcher Systeme noch ein weiter Weg, aber als „unmöglich“ sollte dies nicht mehr betrachtet werden.

Heutige Anwendungen von Expertensystemen liegen z.B. im medizinischen Bereich, wo die EDV dem behandelnden Arzt bei einer Diagnose behilflich sein kann.

Literatur

[1] Thomas Tilli *Fuzzy-Logik,* Franzi‘s Verlag 1993, ISBN 3-7723-4323-6

[2] Michael M. Richter *Prinzipien der Künstlichen Intelligenz,* B.G. Teubner Stuttgart 1992, ISBN 3-519-12269-3

[3] Berger / Krieger / Mahr *Grundlagen der elektronischen Datenverarbeitung, Teil I,* Eigenverlag Berger/Krieger/Mahr Wien 1991, Bestellungen an: Dr. Krieger, Schwindg. 11/7, A-1040 Wien

[4] Marek Obitko *Introduction to genetic algorithms with Java applets* http://cs.felk.cvut.cz/˜xobitko/ga/

[5] Christian Gottschall *Logiktutorium* http://logik.phl.univie.ac.at/˜chris/logik-tutorium.html

[22] U.S.S.=United Star Ship.

Index