

FORSCHUNGSGESELLSCHAFT FÜR STRASSEN- UND VERKEHRSWESEN  
ARBEITSKREIS ENTSCHEIDUNGS- UND OPTIMIERUNGSMETHODEN

VERBAND DEUTSCHER VERKEHRSUNTERNEHMEN

## HEUREKA '96

OPTIMIERUNG IN VERKEHR UND TRANSPORT

VORTRAGSVERANSTALTUNG  
MIT VORSTELLUNG VON HARD- UND SOFTWARE

KARLSRUHE  
13./14. MÄRZ 1996

TAGUNGSBERICHT

ZUSAMMENGESTELLT VON HARTMUT KELLER  
ARBEITSKREIS ENTSCHEIDUNGS- UND OPTIMIERUNGSMETHODEN  
HERAUSGEGEBEN VON DER  
FORSCHUNGSGESELLSCHAFT FÜR STRASSEN- UND VERKEHRSWESEN, KÖLN  
1996

### Darstellung neuerer Optimierungsmethoden am Beispiel des Travelling Salesman Problems

Arbeitskreis Quantitative Entscheidungsmethoden der FGSV  
Dr.-Ing. Manfred Bolze (Albert Speer & Partner GmbH, Frankfurt)  
Dr.-Ing. Markus Friedrich (ptv system GmbH, Karlsruhe)  
Dipl.-Math. Uwe Gabriel (GVS GmbH, Hannover)  
Dr. Joachim Schmidt (ptv GmbH, Karlsruhe)

#### Kurzfassung

Der Beitrag stellt folgende vier neuere Optimierungsverfahren vor, die jeweils am Beispiel des allgemein bekannten Travelling Salesman Problem erläutert werden:

- Simulated Annealing,
- Genetische Algorithmen,
- Simulated Trading,
- Neuronale Netze.

Ziel ist es, diese Optimierungsverfahren, die sich für die Anwendung im Verkehrswesen eignen, für Verkehrsingenieure und Entscheidungsträger in Praxis, Forschung und Lehre zugänglich zu machen.

## 1 Einführung

Der im Jahr 1994 gegründete Arbeitskreis 3.18.6 „quantitative Entscheidungsmethoden“ der Forschungsgesellschaft für Straßen- und Verkehrswesen (FGSV) hat sich zum Ziel gesetzt, neue Entscheidungsmethoden, die sich für die Anwendung im Verkehrswesen eignen, für Verkehrsingenieure und Entscheidungsträger in Praxis, Forschung und Lehre zugänglich zu machen. Quantitative Entscheidungs- und Optimierungsmethoden, die seit dem Erscheinen der „Hinweise für die Anwendung von Entscheidungs- und Optimierungsmethoden im Verkehrswesen, Ausgabe 1982“ (FGSV, 1982) neu entwickelt wurden, sollen systematisch beschrieben werden, und ihre Anwendungsmöglichkeiten im Verkehrswesen sollen dargestellt werden. Mittelfristig ist geplant, die „Hinweise“ aus dem Jahr 1982 fortzuschreiben.

Die Arbeit beschränkte sich bisher vorrangig auf die folgenden neuen, geschlossenen Optimierungsverfahren:

- „Simulated Annealing“, zu übersetzen mit „Simuliertes Erstarren“, ahmt Verfahren zur Herstellung perfekter Kristalle nach, die sehr langsam durch kontrollierte Wärmezufuhr aus der Schmelze abgekühlt werden. Die zugeführte kinetische Energie (Wärme) erlaubt es den in Mulden (lokales Optimum) feststehenden Atomen, die sie umgebenden energetischen Barrieren zu überwinden und ihren optimalen Platz im Kristallgitter (globales Optimum) zu finden.
- Genetische Algorithmen bilden im Sinne Darwins den Selektionsprozeß der Natur nach (survival of the fittest). Mögliche Lösungen werden in einer Informationskette abgebildet, die einem Chromosom entspricht. Jedes Chromosom wird bewertet, und mehrere Chromosomen bilden eine Population. Nach vorzugebenden Regeln wird aus den besten Chromosomen dieser Population eine neue Population gebildet. Dieser Vorgang kann bei häufiger Wiederholung zu einem Optimum führen.
- „Simulated Trading“, zu übersetzen mit „Simulierter Handel“, legt ein Modell zugrunde, das mit den Regelungsmechanismen der freien Marktwirtschaft verglichen werden kann. Das Gesamtsystem wird in Teilsysteme zerlegt, die durch den geregelten Handel mit anderen Teilsystemen ihren eigenen Wert optimieren.
- Neuronale Netze sind informationsverarbeitende Systeme, die aus einer großen Anzahl einfacher Einheiten (Zellen, Neuronen) bestehen. Für die Lösung eines bestimmten Problems wird ein Verbund dieser Einheiten konzipiert, der sich Informationen durch gegenseitige Aktivierung über gerichtete Verbindungen zusetzt. Das neuronale Netz ist in der Lage, auf ein bestimmtes Eingabemuster durch eine Veränderung der Verbindungsgewichte zu reagieren. Es kann aus Trainingsbeispielen lernen und dann neue Aufgaben selbständig lösen.

Diese vier neuen Verfahren können nicht ohne weiteres in die bisher bekannten Systematiken zur Klassifizierung von Optimierungsverfahren eingeordnet werden (z.B. Klassifizierung gemäß FGSV, 1982). Sie sind auch nicht zweckmäßig nach nur einem Kriterium zu klassifizieren.

Vorrangig können Verfahren ohne und mit mathematischer Zielfunktion zur Beschreibung des Optimierungsziels unterschieden werden. Verfahren ohne Zielfunktion können durch die Anwendung von Regeln zu einem Optimum führen, denen kein mathematisches Modell zugrunde liegt (deskriptiv). Verfahren mit Zielfunktion beinhalten eine mathematische Beschreibung des Optimierungsziels (normativ). Einige dieser Verfahren ermitteln die Wirkungen bestimmter Lösungen direkt in der Zielfunktion. Andere benutzen ein separates Wirkungsmodell als Abbild der Realität. Dieses Wirkungsmodell ermittelt die Auswirkungen einer bestimmten Lösung nach festgelegten mathematischen Zusammenhängen, und die Ergebnisse werden dann von der Zielfunktion zur Bewertung und zur Optimierung genutzt.

Eine besondere Gruppe der Optimierungsmethoden bilden die konnektionistischen Verfahren. In diesen wird das zu optimierende Gesamtsystem in miteinander kommunizierende Teilsysteme zerlegt, für die jeweils eigenständig nach vorgegebenen Regeln das individuelle Optimum gesucht wird. Die Teilsysteme beeinflussen sich dann gegenseitig durch die jeweils übermittelten Daten.

Abbildung 1 zeigt die wesentlichen Eigenschaften der hier vorgestellten Optimierungsverfahren.














Verfahren	Zielfunktion	Wirkungsmodell	regelbasiert	konnektionistisch
Simulated Annealing				
Genetische Algorithmen				
Simulated Trading				
Neuronale Netze				
		 teilweise vorhanden trifft teilweise zu		 vorhanden, trifft zu

Abb. 1: Eigenschaften der hier vorgestellten Optimierungsverfahren

Der weitere Beitrag soll die vier Verfahren jeweils an einem Beispiel näher vorstellen. Um die Unterschiede deutlich zu machen, wurde für alle Verfahren das gleiche, allgemein bekannte Travelling Salesman Problem (TSP) gewählt, bei dem ein Handelsreisender vorgegebene Orte zu besuchen und dabei die Streckenlänge seiner Route zu minimieren hat.

## 2 Simulated Annealing

*Simulated Annealing* oder "Simuliertes Erstarren" ahmt Verfahren zur Herstellung perfekter Kristalle nach, die sehr langsam durch kontrollierte Wärmezufuhr aus der Schmelze abgekühlt werden (vgl. Otto, 1994). Die zugeführte kinetische Energie (Wärme) erlaubt es den in Mulden (lokales Optimum) feststehende Atomen, diese energetischen Barrieren zu überwinden und ihren optimalen Platz (globales Optimum) im Kristallgitter zu finden. An die Stellen der Energien treten beim TSP Entfernungen. Simulated Annealing akzeptiert nicht nur energetisch günstigste, d.h. längenminimale Lösungen, sondern auch ungünstigere Konfigurationen mit einer von der Temperatur abhängigen Wahrscheinlichkeit. Ausgehend von einer Anfangslösung und einer Anfangstemperatur werden neue Lösungen erzeugt und bewertet. Neue Lösungen werden akzeptiert, wenn sie besser sind als die vorhandene Lösung. In Abhängigkeit von der Temperatur, die sukzessive sinkt, akzeptiert Simulated Annealing aber auch schlechtere Lösungen, d.h. Lösungen mit einer längeren Reiseroute. Dadurch können lokale Optimalstellen überwunden werden.

### Verfahrensablauf

Die Erläuterung des Simulated Annealing Verfahrens erfolgt anhand des in **Abbildung 2** dargestellten Beispiels, das 10 Städte umfaßt.

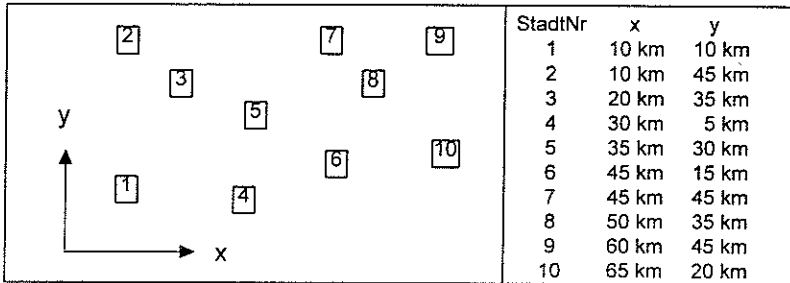


Abb. 2: Beispiel zur Erläuterung des Simulated Annealing Verfahrens (Zahl der Städte: NCity = 10)

Die Route des Handlungsreisenden wird in einem Feld (Array) dargestellt, das die Feldlänge NCity (Zahl der Städte) hat. Jedes Element des Feldes enthält dabei die Nummer einer Stadt. Die Reihenfolge der Feldelemente beschreibt die Reihenfolge, in der der Handlungsreisende die Städte anfährt. Zur Ermittlung einer Reihenfolge, die eine möglichst kurze Reiseroute erfordert, sind beim Simulated Annealing die folgenden Schritte erforderlich:

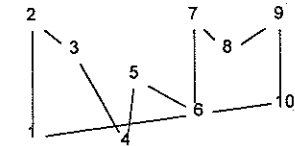
### 1.1 Vorbereiten der Startwerte

- Wähle eine Anfangslösung L1.
- Wähle eine Anfangstemperatur Temp > 0. Die Anfangstemperatur hängt von der Ausdehnung der Städte ab.
- Setze Zahl der Veränderungen V1 = 0. V1 gibt an, wieviel neue Lösungen erzeugt werden.
- Setze Zahl der erfolgreichen Veränderungen V2 = 0. V2 gibt an, wieviel der neuen Lösungen akzeptiert werden.

L1= 1 2 3 4 5 6 7 8 9 10

$$\text{Temp} = \sqrt{(x_{\text{MAX}} - x_{\text{MIN}})^2 + (y_{\text{MAX}} - y_{\text{MIN}})^2}$$

Temp = 68



### 1.2 Ermittle die Wirkungen der aktuellen Lösung L1

Die Wirkung einer Lösung wird durch die Kenngröße *Länge der Route* [km] beschrieben. Die Routenlänge Lge1 der aktuellen Lösung L1 wird hier vereinfacht über die Luftlinienentfernung der Städte bestimmt.

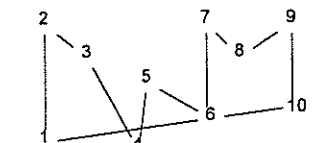
$$\text{Lge1} = \sum_{i=1}^{\text{NCity}} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$$

Lge1 = 261 km

### 2.1 Wähle neue Lösung L2, die eine kleine Änderung der Lösung L1 ist.

Um eine kleine Änderung an der aktuellen Lösung vorzunehmen, wird die Route an zwei zufällig bestimmten Schnittstellen zerschnitten (Lin, 1965). Die zwei Schnittstellen cut1 und cut2 werden so gewählt, daß gilt:

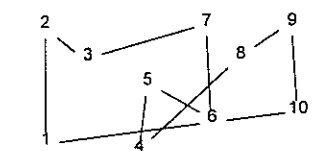
- cut1 ≠ cut2
- 1 ≤ cut1, cut2 ≤ NCity
- wenn cut1 < cut2, dann muß cut2 > cut1+1 sein
- wenn cut2 < cut1, dann muß cut1 > cut2+1 sein



L1= 1 2 3 4 5 6 7 8 9 10

cut1 cut2

L2= 1 2 3 7 6 5 4 8 9 10



Das herausgetrennte Stück der Reiseroute wird nun in umgekehrter Reihenfolge wieder eingefügt.

**2.2 Ermittle Längendifferenz ΔLge der Lösung L1 und der neuen Lösung L2**

- Länge Lge1 der aktuellen Lösung L1     Lge1 = 261,0 km
- Länge Lge2 der neuen Lösung L2        Lge2 = 281,2 km
- Längendifferenz ΔLge = Lge2 - Lge1     ΔLge = + 20,2 km

**2.3 Neue Lösung L2 akzeptieren?**

- Die neue Lösung L2 wird akzeptiert, wenn ΔLge < 0 ist, d.h. wenn die Routenlänge Lge2 der neue Lösung kürzer ist als die aktuelle Routenlänge Lge1.
- Wenn die Routenlänge Lge2 länger ist (ΔLge ≥ 0), dann wird die neue Lösung L2 mit einer Wahrscheinlichkeit P akzeptiert, die von der aktuellen Temperatur Temp abhängt.

$$P = \exp(-20,2/68) = 0,743$$

$$V1 = V1+1$$

$$Z = \text{Zufallszahl zwischen 0 und 1}$$

WENN ΔLge < 0

$$L1 = L2$$

$$V2 = V2+1$$

SONST WENN Z < P

$$L1 = L2$$

$$V2 = V2+1$$

$$P = \exp\left(\frac{-\Delta Lge}{Temp}\right)$$

**3 Wiederhole Schritte 2.1 bis 2.3 mehrmals, verringere dann die Temperatur**

- Verringere die Temperatur, wenn
- Zahl der Veränderungen > 100 x NCity oder wenn
  - Zahl der akzeptierten Veränderungen > 10 x NCity.

WENN V1 > 100 x NCity  
 ODER WENN V2 > 10 x NCity  
 WENN V2 = 0  
 ENDE Simulated Annealing System erstarrt  
 L1 ist beste Lösung

SONST Starte neuen Durchlauf

$$V1 = 0$$

$$V2 = 0$$

$$Temp = Temp \times 0.9$$

Mit dieser niedrigeren Temperatur wird ein neuer Durchlauf (Schritte 2.1 bis 2.3) gestartet. Dabei ist die Wahrscheinlichkeit, daß schlechtere Lösungen akzeptiert werden aufgrund der gesunkenen Temperatur kleiner.

Werden in einem Durchlauf keine neuen Lösungen L2 akzeptiert, wird das System als „erstarrt“ bezeichnet. Die Lösung L1 stellt dann die beste Lösung dar.

**3 Genetische Algorithmen**

Genetische Algorithmen (vgl. Goldberg, 1989) bilden den Selektionsprozeß der Natur nach, bei dem sich die besten Arten weiterentwickeln und schlecht angepaßte Arten aussterben („survival of the fittest“). Das den genetischen Algorithmen zugrunde liegende Modell unterscheidet dabei zwischen einem Gen, einem Chromosom und einer Population:

- Ein *Gen* entspricht i.d.R. einer natürlichen oder einer binären Zahl. Es ist Teil eines Chromosoms und enthält ein Element der Lösung.
- Ein *Chromosom C* setzt sich aus mehreren Genen zusammen und stellt eine mögliche Lösung dar.
- Eine *Population P* umfaßt eine Menge von Chromosomen, d.h. jede Population enthält mehrere mögliche Lösungen.

Startpunkt des genetischen Optimierungsverfahrens ist eine zufällig erzeugte Anfangspopulation. Sie stellt die Keimzelle der nachfolgenden Generationen von Populationen dar. Um eine neue Nachfolgerpopulation aus einer vorhanden Vorgängerpopulation zu erzeugen, werden die einzelnen Lösungen (Chromosomen) der Vorgängerpopulation mit einer sog. Fitneßfunktion bewertet. Während gute Chromosomen einen großen Fitneßwert erhalten, werden schlechte Chromosomen mit einem kleinen Fitneßwert belegt. In Abhängigkeit von der Fitneß werden nun einzelne Chromosomen aus der Vorgängerpopulation zufällig ausgewählt (Selektion) und in die neue Nachfolgerpopulation eingefügt (Reproduktion). Durch die Kreuzung zweier Chromosomen, d.h. durch den Austausch einzelner Gene, ergeben sich dann in der Nachfolgerpopulation Chromosomen, die sich von den Chromosomen der Vorgängerpopulation unterscheiden und neue Lösungen darstellen.

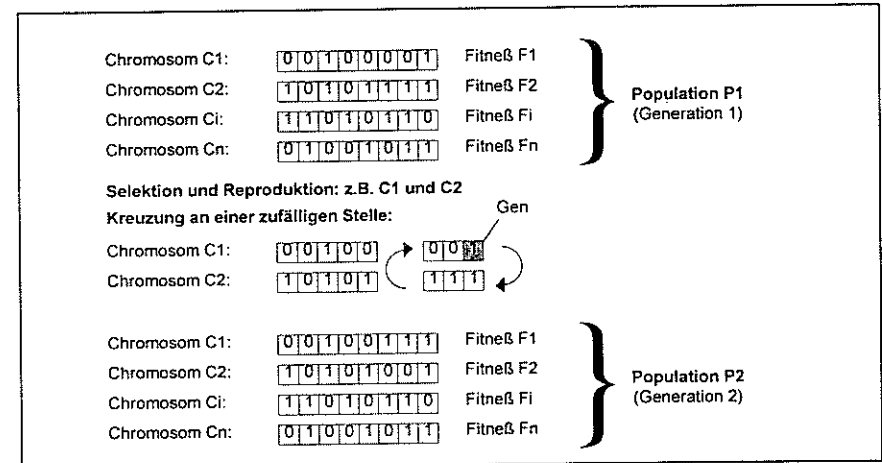


Abb. 3: Erzeugung einer neuen Population

### Verfahrensablauf

Die Wirkungsweise genetischer Algorithmen wird wieder an dem in **Abbildung 2** dargestellten Beispiel gezeigt. Die Route des Handlungsreisenden wird jetzt in einem Chromosom abgebildet, das aus NCity Genen besteht. Jedes Gen entspricht dabei der Nummer einer Stadt. Die Reihenfolge der Gene im Chromosom beschreibt die Reihenfolge, in der der Handlungsreisende die Städte anfährt. Dabei muß gewährleistet sein, daß jede Stadt genau einmal in dem Chromosom enthalten ist. Zur Ermittlung eines Chromosoms, das eine möglichst kurze Reiseroute beschreibt, sind die folgenden Schritte erforderlich:

#### 1 Erzeugen einer Anfangspopulation P (Generation g=1)

- Festlegung einer Populationsgröße n. In der Praxis sollte die Populationsgröße etwa der Zahl der Städte entsprechen,  $n \approx N_{City}$ . z.B.  $n = 4$ :  

C1 (g=1):	1	2	3	4	5	6	7	8	9	10
C2 (g=1):	10	5	1	9	7	3	2	6	4	8
C3 (g=1):	2	9	7	4	5	8	10	1	3	6
C4 (g=1):	8	9	5	3	1	2	7	10	6	4
- Erzeugung von n Chromosomen. Jedes Chromosom enthält jede Stadtnummer genau einmal. Die Reihenfolge ist zufällig.

#### 2 Ermittlung der Fitneß

- Die Fitneß eines Chromosoms  $i$  ergibt sich aus der Kenngröße *Länge der Route* [km], die für jedes Chromosom berechnet wird. Um bei der Berechnung der Fitneß eines Chromosoms die Qualität der anderen Chromosomen zu berücksichtigen, wird als Fitneßfunktion die Kirchhoff'sche Formel mit einem Potentialfaktor  $\alpha$  gewählt. z.B. für  $\alpha = -4$ :  

C1:	Lge1 = 261 km	Fit1 = 35
C2:	Lge2 = 302 km	Fit2 = 19
C3:	Lge3 = 331 km	Fit3 = 13
C4:	Lge4 = 263 km	Fit4 = 33

$$\text{Fitness } \text{Fit}_i = \frac{Lge_i^\alpha}{\sum_{j=1}^n Lge_j^\alpha}$$

#### 3 Erzeugen einer neuen Population P (Generation G+1)

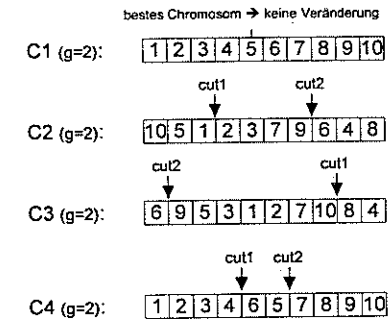
##### 3.1 Selektion und Reproduktion

- Aus der Vorgängerpopulation werden n Chromosomen ausgewählt und in die neue Population kopiert. Die Auswahl der Chromosome erfolgt zufällig, wobei Chromosome mit einem großen Fitneßwert mit einer größeren Wahrscheinlichkeit ausgewählt werden. Auswahl von 4 Chromosomen aus Generation 1: z.B. C1, C2, C4, C1  

C1 (g=2):	1	2	3	4	5	6	7	8	9	10
C2 (g=2):	10	5	1	9	7	3	2	6	4	8
C3 (g=2):	8	9	5	3	1	2	7	10	6	4
C4 (g=2):	1	2	3	4	5	6	7	8	9	10

### 3.2 Kreuzung

- Bei der Kreuzung werden üblicherweise die Gene zweier Chromosomen der Vorgängerpopulation an einer zufälligen Stelle miteinander vertauscht (vgl. **Abbildung 3**).
- Nachdem beim Travelling Salesman Problem jede Stadt pro Chromosom genau einmal enthalten sein soll, kann eine derartige zufällige Kreuzung zu unzulässigen Lösungen führen. Um dies zu vermeiden wird die Reihenfolge der Gene nicht durch die Kreuzung zweier Chromosomen verändert. Statt dessen wird wie beim Simulated Annealing aus jedem Chromosom ein zufälliges Stück herausgeschnitten und in umgekehrter Reihenfolge wieder eingefügt.
- Bei der Erzeugung einer neuen Population sollte immer das beste Chromosom der Vorgängerpopulation unverändert als ein Chromosom der Nachfolgerpopulation übernommen werden.



### 3.3 Ermittlung der Fitneß

- Für die neue Population wird, wie oben erläutert, wieder die Fitneß jedes Chromosoms ermittelt.

### 4 Ende bzw. Abbruch

- Die Erzeugung neuer Populationen wird solange fortgesetzt
  - bis eine vorgegebene Zahl von Generationen erreicht ist oder
  - bis sich die Länge des besten Chromosoms lange nicht ändert.

#### 4 Simulated Trading

Für die Darstellung des Simulated Trading Verfahrens wird eine Erweiterung des bisher behandelten Travelling Salesman Problems benötigt. Es sollen anstelle einer Besuchstour, welche alle Orte (Kunden) besucht, Besuchstouren für M Handlungsreisende geplant werden (M-Travelling Salesman Problem oder M-TSP). Gesucht ist eine Aufteilung der Kunden auf Touren (Handlungsreisende), so daß jeder Kunde genau einmal besucht wird und die insgesamt, innerhalb aller Besuchstouren zurückgelegten Strecken minimal werden. Um die Problembeschreibung einfach zu halten und um die Vergleichbarkeit mit den vorangehenden Ausführungen sicher zu stellen, werden keine Restriktionen für die einzelnen Besuchstouren, wie z.B. die Beschränkung der maximalen Tourdauer oder der Anzahl der besuchten Kunden eingeführt. Deshalb ist die optimale Lösung des M-TSP niemals besser als die optimale Lösung des TSPs (vgl. Weber, 1985).

Beim Simulated Trading handelt es sich um ein iteratives Verbesserungsverfahren, das ausgehend von einer zufällig oder heuristisch erzeugten Startlösung durch die Vertauschung von Kunden zwischen den Touren eine Verbesserung der Ausgangslösung zu erzielen sucht.

Die Vertauschungen von Kunden werden als Verkaufs- und Kaufaktionen innerhalb eines simulierten Börsenhandels modelliert. Angewandt auf das M-TSP entsprechen die Besuchstouren den Marktteilnehmern und die Besuchsorte den Gütern (vgl. auch Malich, 1994). Als Erweiterung gegenüber dem realen Börsenhandel muß innerhalb der Simulated Trading Heuristik berücksichtigt werden, daß neben der Gewinnmaximierung der einzelnen Marktteilnehmer auch die Maximierung des durch alle Marktteilnehmer zu erzielenden Gesamtgewinnes gesucht wird.

Jede Tour wird innerhalb des Verfahren durch einen Tourmanager  $T_j$  repräsentiert. Dieser entscheidet autonom, ob er Besuchsorte an der Börse kauft, verkauft oder ob er wartet. Im Falle des Verkaufs wird der Besuchsort  $i$  vom Tourmanager  $T_1$  zum Verkaufspreis

$$P_i = \text{cost}(T_1) - \text{cost}(T_1 - \{i\})$$

angeboten, d.h. der Verkaufspreis entspricht der Kostenersparnis des Tourmanagers  $T_1$ , falls Kunde  $i$  nicht innerhalb von Tour  $T_1$  besucht wird. Ein Kaufangebot wird von Tourmanager  $T_2$  abgegeben, wenn der Kunde  $i$  zu geringeren Kosten als  $P$  in Tour  $T_2$  eingefügt werden kann.

Zur Abwicklung des Börsenhandels wird der Stockmanager eingeführt. Eine Aufgaben des Stockmanagers ist es, die Verkaufs- und Kaufangebote der Tourmanager über eine festgelegte Anzahl von Iterationen (Level) zu sammeln und eine Zuordnung dieser Aufträge zu finden, welche den Gesamtgewinn maximiert und den vorgegebenen Randbedingungen entspricht (hier: jeder Kunde wird nur durch genau einer Tour besucht).

Die Modellierung dieser Kauf- und Verkaufsoperationen erfolgt über den Trading Graphen  $G(V, E)$ . Ein Knoten in  $G$  entspricht entweder einer Kauf- oder einer Verkaufsoperation eines Tourmanagers. Eine gerichtete Kante stellt die Verknüpfung zwischen einem Verkaufsangebot  $V$  und einem Kaufangebot  $K$  dar (vgl. **Abbildung 4**). Die Kantenbewertung ergibt sich aus der Differenz von Kauf- und Verkaufspreis der verbundenen Knoten, d.h. dem bei einer Verkaufsaktion erzielbaren Gewinn (umrandete Ziffern in Abbildung 4).

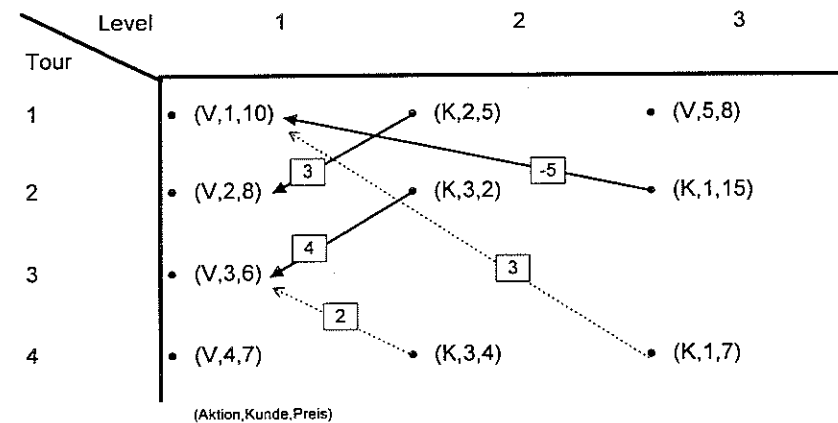


Abb. 4: Trading Graph mit Trading Matching (dick gezeichnete Kanten)

#### Verfahrensablauf

Die Simulated Trading Heuristik beinhaltet prinzipiell die folgenden Schritte:

##### 1 Wahl der Aktion (Kauf, Verkauf oder Warten)

Die Wahl der Aktionen erfolgt nicht deterministisch. Damit soll vermieden werden, daß nach einer gewissen Anzahl von Iterationen keine Verkaufsoperationen mehr durchgeführt werden können (z.B. weil für die angebotenen Aufträge keine Kaufangebote vorliegen). Eine solche Situation bedeutet, daß die Heuristik in einem lokalen Minimum gefangen ist.

Die Wahrscheinlichkeiten für die zufällige Auswahl der Aktionen gehört zu den Verfahrensparametern, mit denen die allgemein verwendbare Heuristik für unterschiedlichste kombinatorische Optimierungsprobleme angepaßt werden kann. Die Wahrscheinlichkeit für einen Verkauf erhöht sich sinnvollerweise mit der erzielbaren Kostenersparnis. Die Wahrscheinlichkeit für einen Kauf eines Besuchsortes bestimmt sich durch das Verhältnis aus Einzelgewinn (Kantenbewertung der Kante, die Kauf und Verkaufsknoten verbindet) und der Summe der Gewinne aller käuflichen Aufträge (Summe aller Kantenbewertungen mit positivem Wert).

## 2 Aufbau des Trading Graphen

Für jede Verkaufs- und Kaufoperation wird ein Knoten im Trading Graph hinzugefügt. Die Anzahl der Iterationen (Level) für den ein Trading Graph aufgebaut werden soll, wird im voraus festgelegt, d.h. die Anzahl der Level, die der Trading Graph besitzt. Für jedes Level wird die durch das Verkaufs- oder Kaufangebot veränderte Tour gespeichert. Diese veränderte Tour ist Basis für die weiteren Angebote späterer Level.

## 3 Ermittlung der Verkaufsoperationen

Wurde der Trading Graph für die festgelegte Anzahl von Level aufgebaut, so muß der Stockmanager die Zuordnung der Verkaufs- und Kaufknoten des Trading Graphen (Kanten) ermitteln, die zu einer Verbesserung der Ausgangslösung führen. Diese Kantenmenge heißt Trading Matching M und muß folgende Bedingungen erfüllen:

- Jeder Verkaufsknoten (Kaufknoten) in M besitzt genau einen zugehörigen Kaufknoten (Verkaufsknoten) in M.
- Mit jedem Knoten im Level l der Tour T befinden sich auch alle Knoten der Level  $1, \dots, l-1$  in M. Damit ist sichergestellt, daß ein Tourmanager nur solche Besuchsorte verkauft, die er auch besitzt (d.h. die im zu diesem Level gehörenden Tourenplan vorhanden sind).
- Die Summe der Gewichte der Kanten in M ist positiv, d.h. die durch M repräsentierten Vertauschungen stellen tatsächlich eine Verbesserung der bisherigen Lösung dar.

Die Bestimmung des Trading Matching entspricht dem Problem der Bestimmung eines maximalen Summenmatching auf einem gerichteten Graphen. Bei Beschränkung der Größe des Trading Graphen (durch Beschränkung der Level) kann daß Problem jedoch durch vollständige Enumeration gelöst werden (vgl. Bachem et al. ,1993).

## 4 Aktualisierung der Tourenpläne

Die vom Trading Matching vorgegebenen Verkaufs- und Kaufoperationen werden durch die Tourmanager realisiert. Dabei müssen u.U. Operationen, die zu dem auf dem letzten Level gespeicherten Tourplan geführt haben, rückgängig gemacht werden. Wird der Besuchsort i an einen Tourmanager  $T_1$  verkauft, so müssen alle Tourmanager, die ebenfalls Kaufangebote für den Kunden i abgegeben haben, alle in späteren Level durchgeführten Operationen rückgängig machen (siehe, gestrichelt gezeichnete Kanten in Abbildung 4).

## Zusammenfassung

Abschließend wird eine zusammenfassende Darstellung eines vollständigen Verfahrensschrittes gegeben:

Gegeben: Zulässige Besuchstouren  $T_1, \dots, T_t$

Für Level 1, ..., MaxLevel

Wiederhole

Für eine beliebige noch nicht betrachtete Tour

Auswahl der Verkaufs- und Kaufoperationen

Aufbau des Trading Graphen

Bis alle Touren betrachtet

Auswahl der Aktionen (Kauf, Verkauf, Warten)

Aktualisierung der Tourenpläne

Der oben beschriebene Verfahrensschritt kann solange wiederholt werden, bis keine Verbesserung erzielbar ist, oder eine festzulegende Anzahl von Verfahrensdurchläufen erreicht wird. Dabei ist nicht ausgeschlossen, daß durch das Verfahren lediglich ein lokales Optimum bestimmt wurde.

Um ein "Festhängen" des Verfahrens auf einem einmal erreichten lokalen Optimum zu vermeiden, kann das in Kapitel 2 beschriebene Prinzip des Simulated Annealing zur Ablaufsteuerung angewendet werden. Hierbei sind zu Beginn der Durchläufe auch Verkaufsoperationen zulässig, die eine Verschlechterung der Ausgangslösung darstellen. Eine weitere Möglichkeit ist es, erreichte Lösungen durch eine "Tabu-Liste" zunächst für folgende Durchläufe zu sperren (Tabu Search).

## 5 Neuronale Netze

Wesentlicher Bestandteil eines künstlichen neuronalen Netzwerks sind stark idealisierte Neuronen. Sie bestehen in Anlehnung an das biologische Vorbild aus drei Komponenten:

- einem Zellkörper,
- Dendriten, welche die Eingabe des Netzes für jede Zelle aufsummieren und
- einem Axon, welches die Ausgabe einer Zelle nach außen weiterleitet, sich verzweigt und mit den Dendriten nachfolgender Neuronen über Synapsen in Kontakt tritt.

Die Stärke der Synapsen wird meist durch einen numerischen Wert, das Verbindungsgewicht dargestellt.

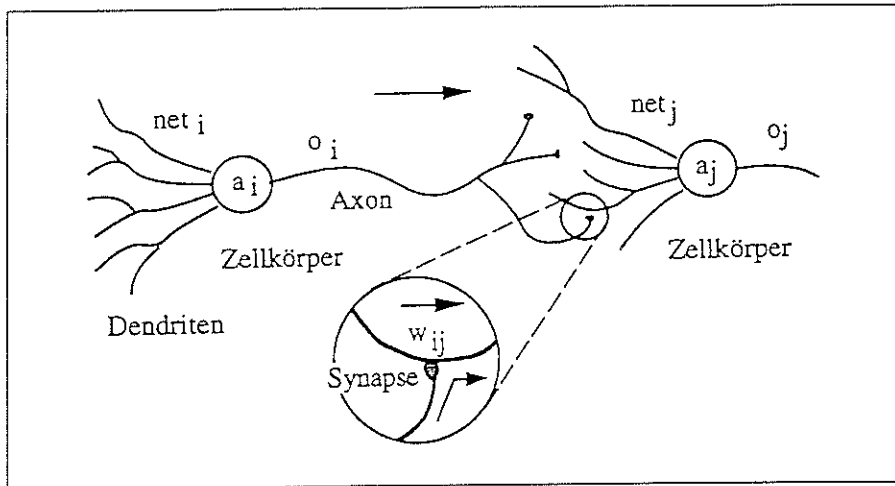


Abb. 5: Zellen als stark idealisierte Neuronen (Zell, 1994)

Für die Realisierung künstlicher neuronaler Netze benötigt man die folgenden Komponenten:

1. **Zellen (Neuronen).** Diese Zellen besitzen die Bestandteile:

- **Aktivierungszustand  $a_j(t)$ :** Er gibt den Grad der Aktivierung der Zelle an.
- **Aktivierungsfunktion  $f_{act}$ :** Sie gibt an, wie sich ein neuer Aktivierungszustand  $a_j(t+1)$  des Neurons  $j$  aus der alten Aktivierung  $a_j(t)$  und der Netzeingabe  $net_j(t)$  ergibt, meist nach der allgemeinen Formel

$$a_j(t+1) = f_{act}(a_j(t), net_j(t)),$$

wobei  $f_{act}$  die Aktivierungsfunktion ist, die aus den angegebenen Parametern die neue Aktivierung berechnet.

- **Ausgabefunktion  $f_{out}$ :** Die Ausgabe der Zelle  $j$  wird durch eine sogenannte Ausgabefunktion aus der Aktivierung der Zelle bestimmt:

$$o_j = f_{out}(a_j).$$

2. **Verbindungsnetzwerk** der Zellen. Ein neuronales Netz kann als gerichteter, gewichteter Graph angesehen werden, wobei die Kanten die gewichteten Verbindungen zwischen den Neuronen darstellen. Das Gewicht der Verbindung von Zelle  $i$  nach Zelle  $j$  wird hier durch  $w_{ij}$  bezeichnet. Die Matrix der Verbindungen aller Zellen (*Gewichtsmatrix*) wird dann mit  $W$  bezeichnet.
3. **Propagierungsfunktion.** Sie gibt an, wie sich die Netzeingabe eines Neurons aus den Ausgaben der anderen Neuronen und den Verbindungsgewichten berechnet. Die Netzeingabe  $net_j(t)$  von Zelle  $j$  berechnet sich nach

$$net_j(t) = \sum_i o_i(t) \times w_{ij}$$

aus der Summe der Ausgaben  $o_i(t)$  der Vorgängerzellen multipliziert mit dem jeweiligen Gewicht  $w_{ij}$  der Verbindung von Zelle  $i$  nach Zelle  $j$ .

4. **Lernregel.** Die Lernregel ist ein Algorithmus, gemäß dem das neuronale Netz lernt, für eine vorgegebene Eingabe eine gewünschte Ausgabe zu produzieren. Am häufigsten wird hierbei die dynamische Modifikation der Verbindungsgewichte  $w_{ij}$  zugrunde gelegt.

In künstlichen neuronalen Netzwerken wird zumeist eine hierarchische Struktur von Neuronen aufgebaut. In der Regel gibt es eine Eingabeschicht, eine gewisse Zahl sogenannter verdeckter Schichten sowie eine Ausgabeschicht. Einfache neuronale Netzwerke bestehen unter Umständen lediglich aus einer Schicht, die gleichzeitig alle Funktionen erfüllt.

### Verfahrensablauf

Die Lösung des Problems erfolgt durch Verwendung der selbstorganisierenden Karten von Kohonen (vgl. Pietsch, Teubner, 1992). Der Aufbau erfolgt schrittweise:

### Einbettung eines Netzes in einen Raum

Das neuronale Netz muß in den Raum eingebettet werden. Dazu ordnet man jedem Neuron  $i$  einen Ortsvektor  $x$  im Raum zu. Die einzelnen Neuronen kann man dann mit ihrem Ortsvektor  $x$  indizieren. Um dem neuronalen Netz eine räumliche Funktionalität zu geben, muß man noch eine Beziehung zwischen der Metrik des Raumes (gewöhnlich der euklidischen) und den Verbindungen der Neuronen (d.h. den Gewichten) herstellen.



### Aufbau und Reproduktion

Das Netz enthält eine einzige vollständig verbundene Schicht von  $N_A$  Neuronen. Jedes Neuron erhält den vollständigen Eingangsvektor  $E_j, (j=1...N_E)$  und gibt seinen Ausgang an den Netzausgang weiter. Da jedem Neuron eindeutig ein Netzausgang zugeordnet ist, können auch die Netzausgänge mit  $x$  indiziert werden.

Aus praktischen Gründen wird zwischen externen und internen Gewichten ( $w_{xj}^{ext}, w_{xy}^{int}$ ) unterschieden. Der effektive Eingang  $\epsilon_x$  des Neurons  $x$  ist daher durch

$$\epsilon_x = w_{xj}^{ext} E_j + w_{xy}^{int} A_y$$

gegeben. Als Aktivierungsfunktion wird die Identität verwendet, die Ausgangsfunktion  $a$  ist beliebig. Die Ausgänge ergeben sich daher durch

$$A_x = a(\epsilon_x).$$

### Festlegen der internen Gewichte

Die internen Gewichte müssen die Metrik des Raumes widerspiegeln.  $d(x,y)$  sei hier die euklidische Metrik. Die internen Gewichte werden so festgelegt, daß sich benachbarte Neuronen erregen, weiter voneinander entfernte dagegen gegenseitig hemmen. Das läßt sich durch die Wahl

$$w = w(d(x,y))$$

erreichen. Dabei ist die Funktion  $w$  nur auf der rechten Halbachse definiert und nimmt für kleine  $d$  positive, für große  $d$  dagegen negative Werte an.

### Lernen der externen Gewichte

Die externen Gewichte selbstorganisierender Karten werden unüberwacht, d.h. ohne äußere Beeinflussung durch Vorgabe der richtigen Lösung oder Bewertung des bisherigen Lernergebnisses, gelernt. Dafür ist eine Vielzahl von Lernregeln in Gebrauch, die meist der Hebb'schen Lernregel ähnlich sind (vgl., Hebb, 1949):

"Wenn Zelle  $j$  eine Eingabe von Zelle  $i$  erhält und beide gleichzeitig stark aktiviert sind, dann erhöhe das Gewicht  $w_{ij}$  (die Stärke der Verbindung von  $i$  nach  $j$ )."

Die hier vorgestellte Regel verwendet das Erregungszentrum  $x'$ ; das ist jenes Neuron, welches den größten Ausgangswert aufweist. Sie lautet:

$$\Delta w_{xj}^{ext} = (E_j - w_{xj}^{ext}) w_{xx'}^{int}$$

Der externe Gewichtsvektor eines Neurons, das in der Nähe des Erregungszentrums liegt, wird also in die Richtung des Eingangsvektors gedreht.

### Zusammenfassung

Durch diese Wahl von internen Gewichten und Lernregel wird die nachbarschafts-erhaltende Eigenschaft des neuronalen Netzwerkes begünstigt. Das so gebildete neuronale Netz wird in einem unüberwachten Lernvorgang iteriert. Dabei stellt sich ein Gleichgewichtszustand ein, bei dem jedes Neuron  $i$  lediglich von einem anderen Neuron  $j$  aktiviert wird (alle anderen Gewichte  $w_{ki}$  streben gegen Null). Darüber hinaus strebt die Summe aller Ausgangswerte gegen das Minimum. Das Travelling Salesman Problem ist gelöst.

Eine beispielhafte Darstellung des Lernverfahrens bei selbstorganisierenden Karten zeigt die folgende Abbildung 6.

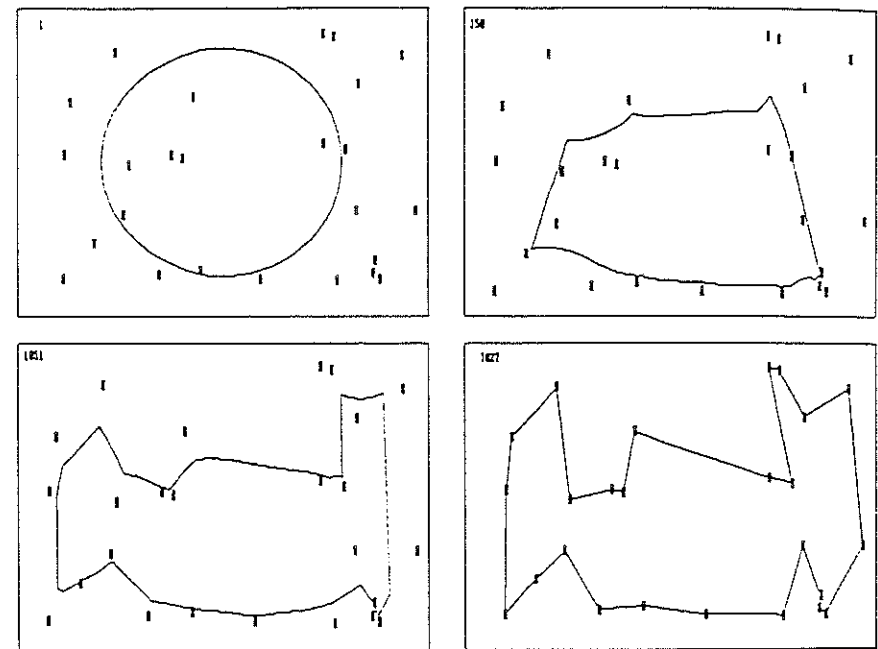


Abb. 6: Lernen des optimalen Reiseweges mit Hilfe selbstorganisierender Karten. Die Abbildung zeigt verschiedene Lernstadien, die Zahlen geben die Zahl der Iterationsschritte an (Kinnebrock, 1994)

## Literaturverzeichnis

- Bachem, A.; Hochstättler, W.; Malich, M. (1993):**  
*The Simulated Trading Heuristik for Solving Vehicle Routing Problems.*  
Technical Report 93.139, Mathematisches Institut der Universität zu Köln.
- FGSV, (1982):**  
*Hinweise für die Anwendung von Entscheidungs und Optimierungsmethoden im Verkehrswesen.* Forschungsgesellschaft für Straßen- und Verkehrswesen, Köln.
- Goldberg, D. E. (1989):**  
*Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.
- Hebb, D. (1949):**  
*Organisation of behaviour.* Wiley New York.
- Kinnebrock, W. (1994):**  
*Neuronale Netze - Grundlagen, Anwendungen, Beispiele.* Oldenbourg München Wien.
- Lin, S. (1965):**  
Bell System Technical Journal, Nr 44.
- Malich, M. (1994):**  
*Simulated Trading: Ein paralleles Verfahren zur Lösung von kombinatorischen Optimierungsproblemen.* Dissertation, Universität zu Köln.
- Otto, T. (1994):**  
*Reiselust - Travelling Salesman, eine neue Strategie für eine alte Aufgabe.* c't, Heft 1.
- Pietsch, W; Teubner, A. (1992):**  
*Konnektionistische Tourenplanung für die dezentrale Vertriebsaussendienststeuerung.* In: Operations Research Proceedings 1991. Springer Verlag Berlin Heidelberg.
- Weber, R. (1985):**  
*Vehicle Routing- und Scheduling Probleme unter besonderer Berücksichtigung von Dial-a-Ride Problemen.* Verlag V. Florentz, München.
- Zell, A. (1994):**  
*Simulation neuronaler Netze.* Addison-Wesley.

## DRUM - Ein Modell zur dynamischen Routensuche und Umlegung

Dr.-Ing. Dirk Serwill  
Oberingenieur am Institut für Stadtbauwesen  
RWTH Aachen  
Mies-van-der-Rohe-Str. 1, 52056 Aachen

- Kurzfassung -

*Am Institut für Stadtbauwesen der RWTH Aachen wurde ein innovatives Modellkonzept zur dynamischen Routensuche und Umlegung (DRUM) entwickelt, das in der Lage ist, zeitlich-räumliche Routen als Minimum der Summe von zeitvarianten Netzwideständen zu bestimmen. Dabei werden Alternativrouten pro Zeitschritt, die sich aufgrund unterschiedlicher Netzkenntnis und unterschiedlichen Schätzvermögens von Verkehrsteilnehmern verhaltenshomogener Personengruppen einstellen, berücksichtigt. Der modulare Aufbau des Modells ermöglicht den Einsatz sowohl auf der strategischen Planungsebene, bei der Systemanalyse organisatorisch-betrieblicher Maßnahmen des Verkehrsmanagements, als auch auf der operationalen Ebene im Zusammenhang mit der Integration aktueller Verkehrsdaten.*

### 1. Einführung

Die Simulation der Verkehrswegewahl mit Hilfe eines Routenwahlalgorithmus und einer anschließenden Umlegung ist der zentrale Prozeß bei der Ermittlung von Verkehrsbelastungen in einem Verkehrsnetz. Dem Verkehrsingenieur steht hierzu seit Jahren eine Reihe von Verfahren zur Verfügung, die sich alle das Prinzip zunutze machen, daß ein Fahrer die Route zum Ziel wählt, die seinen Aufwand zur Distanzüberwindung - zumeist ausgedrückt in Reisezeit - minimiert. Anders ausgedrückt: Routen werden nach dem Nutzenmaximierungsprinzip ausgewählt. Die Reisezeit jeder einzelnen Fahrt setzt sich aus den Fahrzeiten auf den einzelnen Teilstrecken der Route und den Wartezeiten an den dazwischenliegenden Knoten zusammen. Diese Zeiten wiederum hängen direkt von den dort vorherrschenden Verkehrsbelastungen ab.

Deshalb ist im motorisierten Individualverkehr eine ständige Wechselwirkung zwischen den existierenden Verkehrsbedingungen und der Wegewahlentscheidung der Autofahrer gegeben. Die Aggregation vieler Wegewahlentscheidungen bestimmt das Verkehrsbild. Die Verkehrsbelastungen auf der Straße beeinflussen wiederum die Routenwahl, da ein Verkehrsteilnehmer bemüht ist, den Aufwand für die Distanzüberwindung zu minimieren.