

# Evolutionäre Algorithmen

## Anwendungsbeispiele

**Prof. Dr. Rudolf Kruse**     **Pascal Held**

`{kruse,pheld}@iws.cs.uni-magdeburg.de`

Otto-von-Guericke-Universität Magdeburg

Fakultät für Informatik

Institut für Wissens- und Sprachverarbeitung

# Übersicht

## 1. Flugroutenplanung: ROGENA

Problemstellung

Lösungsidee

Lösungsansatz mit evolutionären Algorithmen

## 2. Erlernen von Fuzzy-Reglern

# Priekt: Rogena (DLR, 1995)

**Flugzeuge** bewegen sich im Luftraum normalerweise

- auf Standardrouten zwischen Flughäfen,
- mit vorgeschriebenem Mindestabstand auf gleicher Route,
- abhängig von ihrer Flugrichtung auf unterschiedlichen Höhen

**Vorteile** dieser Lösung:

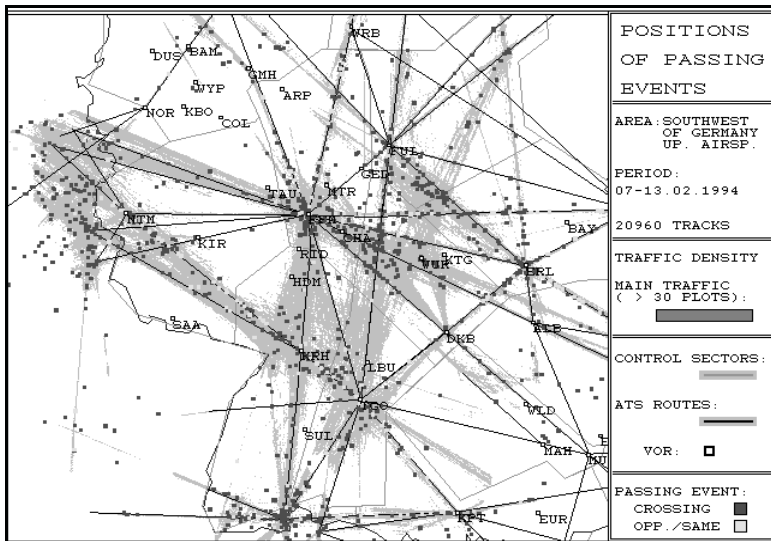
- einfache Regeln/Vorschriften für Flugverkehr,
- leichte Kontrolle der Flugrouten durch Fluglotsen,
- große Sicherheit im Flugverkehr (.w. Kollisionsvermeidung)

**Nachteile** dieser Lösung:

- nur relativ kleiner Teil des Luftraums wird genutzt,
- Flugzeugdichte in Nähe von Flughäfen ist relativ eng begrenzt

**Lösung** ist steigendem Flugverkehr nicht mehr angemessen

# Luftraum in der Umgebung von Frankfurt



# Lösungsidee: Fluglotsenunterstützung

**Luftraum** zwischen Standardrouten muss genutzt werden für zukünftige Luftverkehrssteigerungen

**Problem** dieses Ansatzes:

- sich freier bewegende Flugzeuge sind schwerer zu koordinieren  
⇒ Arbeitsbelastung für Fluglotsen steigt

**Aufgabenstellung:** Entwicklung eines Unterstützungswerkzeugs, das Konstruktion sicherer und effizienter Routen zwischen Eintritts- und Austrittspunkt in dem vom Lotsen kontrollierten Teil des Luftraumes übernimmt

**vorgegebene Rahmenbedingungen:**

- Eintrittszeitpunkt, Ein- und Austrittsort im kontrollierten Bereich,
- Flugeigenschaften der kontrollierten Flugzeuge,
- Bewegung weiterer Flugzeuge (Kollisionsvermeidung),
- Sperrgebiete (Bebauung, militärische Nutzung, schlechtes Wetter)

# Lösungsansatz mit evolutionären Algorithmen

- Suchraum für mögliche Flugrouten ist sehr groß
- analytische Konstruktionsmethode ist schwer zu finden
- **Suche** mit EA: **ROGENA** (free ROuting with GENetic Algorithms) [Gerdes, 1994, Gerdes, 1995]
- notwendigen Daten und Berechnungsformeln für Beschreibung der Flugeigenschaften und des Flugverhaltens von Flugzeugen: BADA-Datenbank der EUROCONTROL
- betrachte Ausschnitt von  $200 \times 200$  nautischen Meilen (NM) mit Höhe von 0 bis 10000 Fuß (ft) aus Luftraum
- Mindestabstände zwischen Flugzeugen: 5 NM, beim Endanflug 2.5–6 NM (abhängig von Gewicht der beiden Flugzeuge)
- EA beginnt Arbeit erst bei Flugroutenkonflikten (Unterschreitung der Sicherheitsabstände)

# Kodierung der Lösungskandidaten

Flugroute = Sequenz von Linienstücken

Chromosomen mit variabler Länge/Anzahl von Genen

jedes Gen: zu überfliegenden Punkt;  
zusätzlich mit Überfluggeschwindigkeit:

	$x$	$y$	$z$	$v$
Gen 1	86.2	15.8	1.65	258
Gen 2	88.9	24.7	1.31	252
$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$
Gen $k$	105.0	98.0	0.00	120

Nebenbedingungen, z.B.

- monoton fallende Flugroute (da Landeanflüge modelliert werden),
- maximale Beschleunigung/Verzögerung zwischen Punkten,
- minimaler Winkel zwischen Linienstücken (Kurvenradius) etc.

# Fitnessfunktion

Berücksichtigung folgender Eigenschaften einer Flugroute:

- einzuhaltender Sicherheitsabstand zu anderen Flugzeugen,
- kein Durchfliegen von gesperrtem Luftraum,
- Länge der Flugroute bis zur Landebahn,
- Pünktlichkeit des Fluges (planmäßige Ankunft),
- möglichst geringe Abweichungen von optimaler Sinkrate (berechnet nach BADA-Datenbank),
- keine zu spitzen Winkel zwischen Linienstücken, um Abweichungen von tatsächlicher Flugbahn klein zu halten

Diese **Eigenschaften** gehen mit Gewichtungsfaktoren in zu minimierende Fitnessfunktion ein

**Benutzer** kann Gewichtungsfaktoren für einzelnen Eigenschaften über Schieberegler beeinflussen



# Ablauf des evolutionären Algorithmus

**Ziel:** Erzeugen einer sicheren und effizienten Flugroute für neu in den kontrollierten Luftraum eintretendes Flugzeug

**Basis:** modifizierte Form eines EA (ein Chromosom wird entweder Crossover oder Mutation unterworfen)

**Populationsgröße:** 60 Individuen

Initialisierung der **Anfangspopulation:** durch wiederholte zufällige Veränderung der Standardflugroute

**Selektionsverfahren:**

i.W. Glücksradauswahl, wobei allerdings Chromosomen, deren Fitness über Schwellenwert liegt, nicht in Berechnung der Selektionswahrscheinlichkeit eingehen (entspricht Forderung einer Mindestgüte)

Schwellenwert wird mit  $t$  gesenkt in Abhängigkeit vom Fitnessdurchschnitt der Population (Kombination aus Glücksradauswahl und Sintflutalgorithmus)

# Anwendung genetischer Operatoren

20 Chromosomen werden unverändert übernommen, darunter die 5 besten Chromosomen (Elitismus)

20 Chromosomen werden **2-Punkt-Crossover** unterworfen

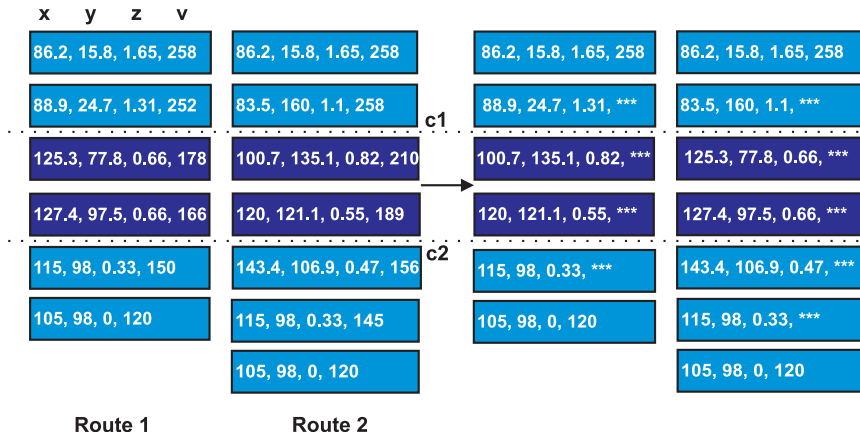
20 Chromosomen werden speziellem **Mutationsoperator** unterworfen:

- entweder völlig zufällige Koordinatenänderung (globale Suche)
- oder mittlere Veränderung in Punkt „in der Nähe“
- oder kleine Veränderung in Punkt „in der Nähe“ (Zufallsaufstieg)
- zusätzlich: Veränderung der Genanzahl (mit geringer W'keit)  
Lösch- oder Einfügeort im Chromosom ist zufällig, neuer Punkt zufällig in Nähe seiner Nachbarn initialisiert

## Reparaturmechanismen:

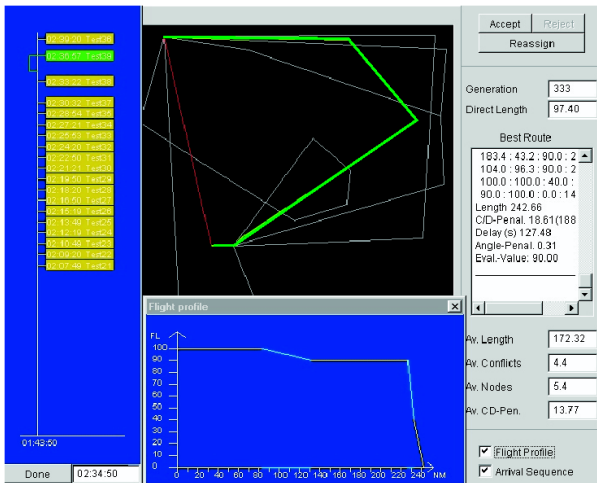
Abfolge von Sink- und Steigvorgängen ist unökonomisch, ebenso Abfolge von Beschleunigungs- und Bremsvorgängen

## 2-Punkt-Crossover



Grund: Austausch von Teilstücken in Routen

# Benutzeroberfläche



The interface displays a list of test cases on the left, a map of the flight path in the center, a flight profile graph at the bottom, and a summary of the best route on the right.

**Test Cases List:**

- 02:30:30 Test28
- 02:30:47 Test38
- 02:33:22 Test28
- 02:30:32 Test27
- 02:29:44 Test26
- 02:27:21 Test24
- 02:24:43 Test23
- 02:24:20 Test22
- 02:22:50 Test21
- 02:21:21 Test20
- 02:14:50 Test24
- 02:13:20 Test28
- 02:10:50 Test27
- 02:10:30 Test26
- 02:10:30 Test25
- 02:10:19 Test24
- 02:10:09 Test23
- 02:09:20 Test22
- 02:07:09 Test21

**Flight Path Map:** Shows a green highlighted path connecting several waypoints on a map.

**Flight Profile Graph:** Shows altitude (FL) in feet versus distance (NM). The altitude starts at 100 feet, remains constant until approximately 100 NM, then drops to 90 feet, remains constant until approximately 220 NM, and finally drops to 0 feet at the end of the route.

**Best Route Summary:**

- 183.4 : 43.2 : 90.0 : 2
- 104.0 : 96.3 : 90.0 : 2
- 100.0 : 100.0 : 40.0 :
- 90.0 : 100.0 : 0.0 : 14
- Length 242.66
- CD-Penal. 18.61(188)
- Delay (s) 127.48
- Angle-Penal. 0.31
- Eval-Value: 90.00

**Summary Statistics:**

- Av. Length: 172.32
- Av. Conflicts: 4.4
- Av. Nodes: 5.4
- Av. CD-Pen.: 13.77

**Options:**

- Flight Profile
- Arrival Sequence

# Testläufe und Ergebnisse

## Benutzeroberfläche:

- Originalroute in rot (direkte Verbindung von Eintrittsort und Zielpunkt)
- beste aktuelle Route in grün
- alternative Routen in grau
- Höhendigramm der Route in getrenntem Fenster
- links: Zeitleiter mit Landezeiten aller Flugzeuge

mehrere **Simulationsläufe** mit realen Daten (beschreiben Eintrittsort/-zeitpunkt und tatsächliche Flugbahn)

von **ROGENA** erzeugten Flugrouten: deutlich kürzer als die tatsächlichen, ohne erzeugte Konflikte mit anderen Flugzeugen

wirkungsvolle **Unterstützung** eines Fluglotsen bei Koordinierung

**Verkürzung der Routen** ermöglicht höheren Flugzeugdurchsatz

# Übersicht

## 1. Flugroutenplanung: ROGENA

## 2. Erlernen von Fuzzy-Reglern

Fuzzy-Logik

Fuzzy-Mengenlehre

Linguistische Variablen

Fuzzy-Regelung

Erzeugen/Optimieren von Fuzzy-Reglern mit EA

# Kurzeinführung in die Fuzzy-Theorie

**Klassische Logik:** nur Wahrheitswerte *wahr* und *falsch*

**Klassische Mengenlehre:** entweder *ist Element* oder *nicht*

Zweiwertigkeit dieser Theorien: oft unangemessen

**Beispiel: Sorites-Paradoxon** (griech. *sorites*: Haufen)

*wahr*: „Eine Milliarde Sandkörner sind ein Sandhaufen.“

*wahr*: „Wenn man von einem Sandhaufen ein Sandkorn entfernt,  
bleibt ein Sandhaufen übrig.“

*wahr*: 999 999 999 „Sandkörner sind ein Sandhaufen.“

mehrfache Wiederholung des gleichen Schlusses:

*falsch*: „1 Sandkorn ist ein Sandhaufen.“

Frage: Bei welcher Anzahl Sandkörner ist Schluss nicht wahrheitsbewahrend?

# Kurzeinführung in die Fuzzy-Theorie

Offenbar: keine genau bestimmte Anzahl Sandkörner, bei der Schluss auf nächstkleinere Anzahl falsch ist

Problem: Begriffe der natürlichen Sprache (z.B. „Sandhaufen“, „kahlköpfig“, „warm“, „schnell“, „hoher Druck“, „leicht“ etc.) sind **vage**

Vage Begriffe sind *unexakt*, aber nicht *unbrauchbar*

- auch für vage Begriffe gibt es Situationen/Objekte auf die sie *sicher anwendbar* sind, und solche, auf die sie *sicher nicht anwendbar* sind
- dazwischen: **Penumbra** (lat. für *Halbschatten*) von Situationen, in denen es unklar ist, ob die Begriffe anwendbar sind, oder in denen sie nur mit Einschränkungen anwendbar sind („kleiner Sandhaufen“).
- Fuzzy-Theorie: mathematische Modellierung der Penumbra



# Fuzzy-Logik

**Erweiterung der klassischen Logik** um Zwischenwerte zwischen *wahr* und *falsch*

**Wahrheitswert:** jeder Wert aus  $[0, 1]$ , wobei  $0 \hat{=} falsch$  und  $1 \hat{=} wahr$

**Erweiterung der logischen Operatoren**

Klassische Logik		Fuzzy-Logik	
Operation	Notation	Operation	Notation
Negation	$\neg a$	Fuzzy-Negation	$1 - a$
Konjunktion	$a \wedge b$	<i>t</i> -Norm	$\min(a, b)$
Disjunktion	$a \vee b$	<i>t</i> -Konorm	$\max(a, b)$

**Grundprinzipien** der Erweiterung:

- für Extremwerte 0 und 1 sollen sich Operationen genauso verhalten wie ihre klassischen Vorbilder (Rand-/Eckbedingungen)
- für Zwischenwerte soll Verhalten monoton sein
- Gesetze der klassischen Logik sollen (fast alle) erhalten werden

# Fuzzy-Mengenlehre

klassische Mengenlehre basiert auf Begriff „*ist Element von*“ ( $\in$ )  
alternativ: Zugehörigkeit zu Menge mit *Indikatorfunktion*  
beschreibbar: sei  $X$  eine Menge, dann heißt

$$I_M : X \rightarrow \{0, 1\}, \quad I_M(x) = \begin{cases} 1, & \text{falls } x \in M, \\ 0, & \text{sonst,} \end{cases}$$

**Indikatorfunktion** der Menge  $M$  bzgl. Grundmenge  $X$

Fuzzy-Mengenlehre: Indikatorfunktion wird durch  
*Zugehörigkeitsfunktion* bestimmt:

$$\mu_M : X \rightarrow [0, 1], \quad \mu_M(x) \hat{=} \text{Zugehörigkeitsgrad von } x \text{ zu } M,$$

**Zugehörigkeitsfunktion** (membership function) der  
**Fuzzy-Menge**  $M$  bzgl. der *Grundmenge*  $X$

Fuzzy-Menge: definiert über ihre Zugehörigkeitsfunktion

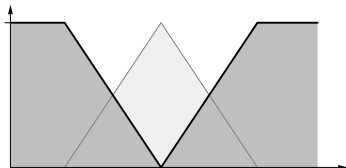
# Operationen

## Grundprinzip dieser Erweiterung:

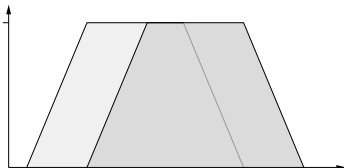
Elementweise Anwendung der logischen Operatoren für  
(Fuzzy-)Mengen  $A$  und  $B$  über Grundmenge  $X$

<b>Komplement</b>	klassisch	$\bar{A} = \{x \in X \mid x \notin A\}$
	fuzzy	$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$
<b>Schnitt</b>	klassisch	$A \cap B = \{x \in X \mid x \in A \wedge x \in B\}$
	fuzzy	$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$
<b>Vereinigung</b>	klassisch	$A \cup B = \{x \in X \mid x \in A \vee x \in B\}$
	fuzzy	$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$

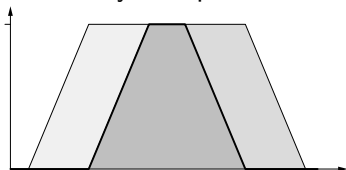
# Fuzzy-Mengenoperationen: Beispiele



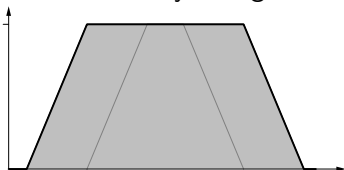
Fuzzy-Komplement



zwei Fuzzy-Mengen



Fuzzy-Schnitt



Fuzzy-Vereinigung

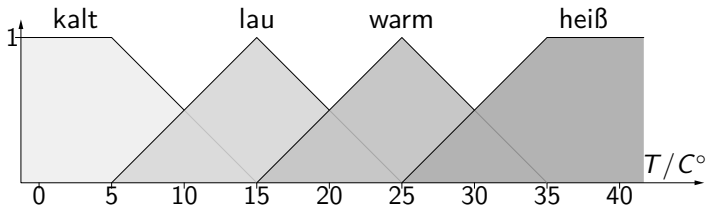
# Fuzzy-Partitionen und Linguistische Variablen

Im Wertebereich durch sprachliche (linguistische) Ausdrücke beschreiben zu können, wird er mithilfe von Fuzzy-Mengen fuzzy-partitioniert

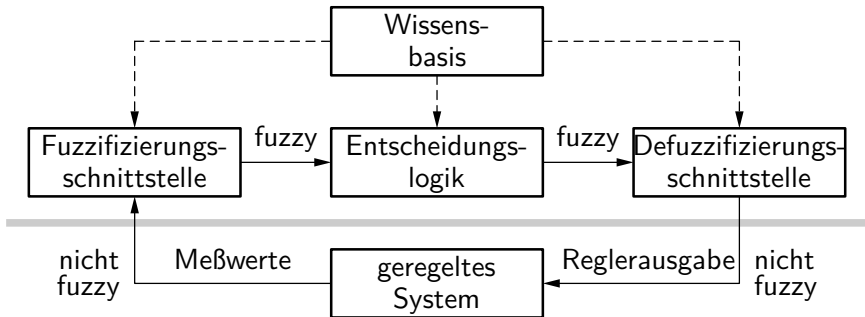
- jeder Fuzzy-Menge der Partitionierung ist ein linguistischer Term zugeordnet
- übliche Bedingung: an jedem Punkt müssen sich Zugehörigkeitsgrade aller Fuzzy-Mengen zu 1 addieren

**Beispiel:** Fuzzy-Partitionierung für Temperaturen

Linguistische Variable mit den Werten *kalt*, *lau*, *warm* und *heiß*.

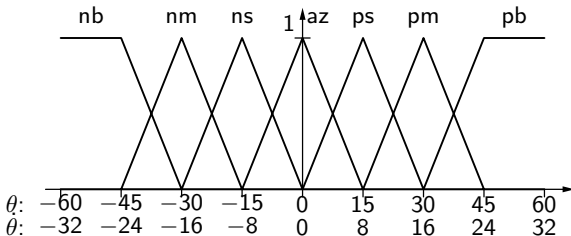
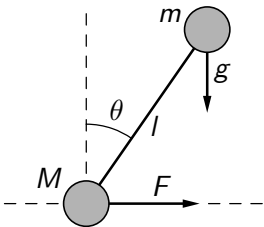


# Architektur eines Fuzzy-Reglers



- Wissensbasis enthält Fuzzy-Regeln für Steuerung und Fuzzy-Partitionen der Wertebereiche der Variablen
- Fuzzy-Regel: **if**  $X_1$  **is**  $A_{i_1}^{(1)}$  **and** ... **and**  $X_n$  **is**  $A_{i_n}^{(n)}$  **then**  $Y$  **is**  $B$ .  
 $X_1, \dots, X_n$  sind Messgrößen und  $Y$  ist Stellgröße  
 $A_{i_k}^{(k)}$ ,  $B$ : linguistische Terme (mit Fuzzy-Mengen assoziiert)

# Beispiel eines Fuzzy-Regler: Stabbalance

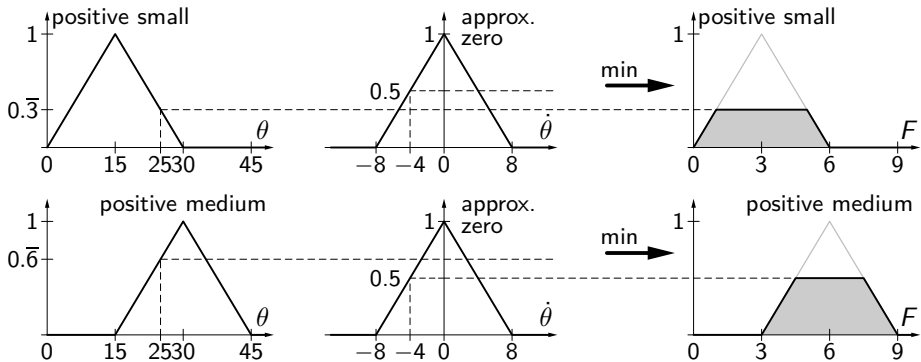


## Abkürzungen

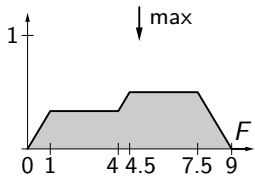
- pb – positive big
- pm – positive medium
- ps – positive small
- az – approximately zero
- ns – negative small
- nm – negative medium
- nb – negative big

$\dot{\theta} \setminus \theta$	nb	nm	ns	az	ps	pm	pb
pb			ps	pb			
pm				pm			
ps	nm		az	ps			
az	nb	nm	ns	az	ps	pm	pb
ns				ns	az		pm
nm				nm			
nb				nb	ns		

# Fuzzy-Regelung nach Mamdani-Assilian

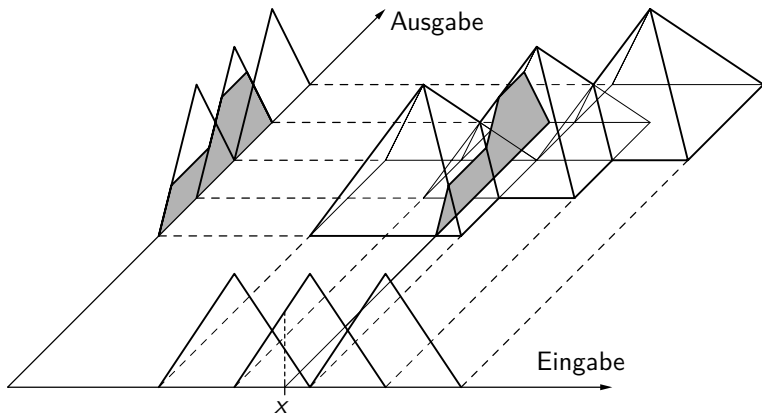


Regelauswertung in Mamdani-Assilian-Regler. Eingabepupel (25, -4) fuhrt zur rechts gezeigten unscharfen Ausgabe. Aus dieser Fuzzy-Menge wird entsprechender Ausgabewert durch Defuzzifizierung bestimmt, z.B. durch Mean-of-Maxima-Methode (MOM) oder die Schwerpunktmethod (COG).





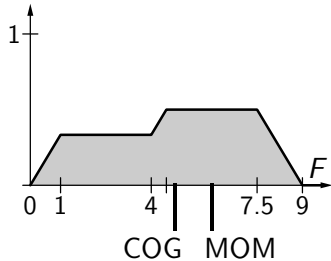
# Fuzzy-Regelung nach Mamdani–Assilian



Fuzzy-Regelsystem einer Mess- und einer Stellgröße und drei Fuzzy-Regeln: jede Pyramide = eine Fuzzy-Regel, Eingabewert  $x$  führt zur grau gezeichneten unscharfen Ausgabe

# Defuzzifizierung

Auswertung der Fuzzy-Regeln liefert **Ausgabe-Fuzzy-Menge**, die in einen **scharfen Stellwert** umgewandelt werden muss. Vorgang heißt **Defuzzifizierung**.



wichtigste Defuzzifizierungsmethoden:

- **Schwerpunktmethode** (Center of Gravity, COG)  
Schwerpunkt der Fläche unter Ausgabe-Fuzzy-Menge
- **Flächenmittelpunktmethode** (Center of Area, COA)  
Punkt, der Fläche unter Ausgabe-Fuzzy-Menge in gleich große Teile teilt
- **Maxima-Mittelwert-Methode** (Mean of Maxima, MOM)  
arithmetisches Mittel der Stellen mit maximalem Zugehörigkeitsgrad

# Erzeugen/Optimieren von Fuzzy-Reglern mit EA

Bei Mamdani-Assilian-Regler kann optimiert werden:

- **Regelbasis** (welche Regeln, welche Ausgaben)
- **Fuzzy-Mengen/Fuzzy-Partitionen**  
(Form, Lage, Ausdehnung, ggf. Anzahl der Fuzzy-Mengen)
- ***t*-Norm** bzw. ***t*-Konorm** für Regelauswertung (selten)
- Parameter der **Defuzzifizierungsmethode** (falls vorhanden; selten)
- welche **Eingangsgrößen** in Regeln verwendet werden  
(Merkmalsauswahl, engl. *feature selection*)

hier: nur Optimierung von Regelbasis und Fuzzy-Mengen bei fester Wahl der Eingabe-/Messgrößen

Regelauswertung: über Minimum und Maximum, Defuzzifizierung: Schwerpunktmethode

## Mögliche Vorgehensweisen

1. *optimiere Regelbasis und Fuzzy-Partitionen gleichzeitig*  
Nachteil: gleichzeitiges Optimieren extrem vieler Parameter
2. *optimiere erst Fuzzy-Partitionen bei vorgegebener Regelbasis, dann Regelbasis mit besten Fuzzy-Partitionen*  
Nachteil: Expertenwissen benötigt zum Aufstellen der Regelbasis (Wahl einer zufälligen Regelbasis ist wenig erfolgversprechend)
3. *optimiere erst Regelbasis für vorgegebene Fuzzy-Mengen, dann Fuzzy-Partitionen mit bester Regelbasis*  
Fuzzy-Mengen können z.B. äquidistant (gleichmäßig) verteilt werden. In diesem Fall: Anwender muss Zahl der Fuzzy-Mengen je Messgröße und für Stellgröße vorgeben.

hier: nur dritte Möglichkeit

# Fitnessfunktion

Ein guter Regler sollte verschiedene Kriterien erfüllen:

- aus jeder möglichen Situation sollte der Sollzustand erreicht werden
- Sollzustand sollte möglichst schnell erreicht werden
- Sollzustand sollte mit geringem (Energie-)Aufwand erreicht werden

Der Regler wird mehrfach testweise auf zu regelndes System angewandt

(hier: Simulation des Stabbalance-Problems in Rechner, mehrere, zufällig gewählte Anfangssituationen)

je nach Regelerfolg/Regelgüte erhält Regler Punkte (Anzahl Situationen, Dauer erfolgreicher Regelung, Energieaufwand)

- **beachte:**

in dieser Anwendung ist Bewertung der Individuen die mit Abstand aufwendigste Operation. Jedes Individuum muss über gewisse Mindestzahl von Zeitschritten zur Regelung eingesetzt werden.

## Bewertung des Regelerfolgs

- Istwert weicht zu stark vom Sollwert ab  $\Rightarrow$  Abbruch (Fehlschlag)  
(z.B. Stabbalance-Problem: Istwert muss in  $[-90^\circ, 90^\circ]$  bleiben)
- nach bestimmten Dauer sollte Istwert in Nähe des Sollwertes liegen und dort verbleiben (Toleranzbereich).  
Falls nicht, wird ebenfalls abgebrochen (Fehlschlag)
- Toleranzbereich wird mit  $t$  verringert (langsameres Hinführen auf gewünschtes Ziel)
  - in ersten Generationen: hinreichend, wenn Stab nicht umfällt
  - später: Stab muss in einem immer engeren Winkelbereich um senkrechte Lage gehalten werden
- Beträge der Stellwerte werden aufsummiert und als Strafterm benutzt  
(in Gleichgewichtslage hat schnelles Umschalten zwischen großen Kräften gleiche Wirkung wie Regelung mit geringen Kräften. Hohe Kräfte sollen vermieden werden.)

# Erzeugen/Optimieren der Regelbasis: Kodierung

- nur vollständige Regelbasen (zu jeder Kombination von Eingabe-Fuzzy-Mengen gibt es eine Regel)
- für jede Kombination von Eingabe-Fuzzy-Mengen muss lediglich der linguistische Term der Stellgröße festgelegt werden (i.w. Ausfüllen einer Tabelle)

Beispiel: Regelbasis für Stabbalance-Regler:

$\dot{\theta} \setminus \theta$	nb	nm	ns	az	ps	pm	pb
pb	az	ps	ps	pb	pb	pb	pb
pm	ns	az	ps	pm	pm	pb	pb
ps	nm	ns	az	ps	pm	pm	pb
az	nb	nm	ns	az	ps	pm	pb
ns	nb	nm	nm	ns	az	ps	pm
nm	nb	nb	nm	nm	ns	az	ps
nb	nb	nb	nb	nb	ns	ns	az

schematisch

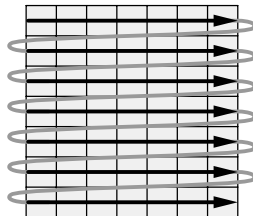

# Kodierung der Regelbasis

**Linearisierung** (Umwandlung in Vektor)

Tabelle wird in willkürlicher, aber festen Reihenfolge durchlaufen und Tabelleneinträge werden in Vektor aufgelistet

Beispiel: zeilenweise Auflistung

Problem: Nachbarschaftsbeziehungen  
zwischen den Zeilen gehen verloren  
(benachbarte Einträge sollten ähnliche  
linguistische Terme enthalten; z.B. sollte  
dies der Crossover berücksichtigen)



**Tabelle** (direkte Verwendung des Schemas)  
zwei- oder mehrdimensionale Chromosomen  
(spezielle genetische Operatoren benötigt)



## Variation (Standardmutation)

**Regel/Tabelleneintrag** wird zufällig gewählt

**Linguistischer Term** der Ausgabe wird zufällig verändert

ggf. **mehrere Regeln/Tabellenfelder** gleichzeitig veränderlich

ggf. vorteilhaft: **schränke Mutation einer Regelbasis so ein**,  
dass Tabelleneintrag nur auf linguistische Terme geändert werden  
kann, die dem vorhandenen Eintrag benachbart oder hinreichend  
nahe sind

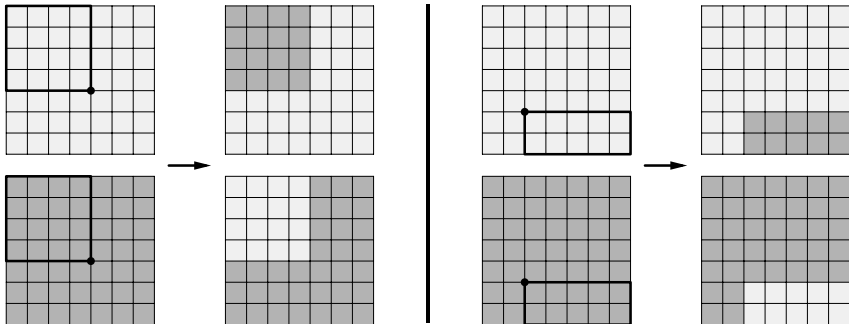
Beispiele: „positive small“ kann nur auf „approximately zero“  
oder „positive medium“ geändert werden

„negative big“ kann nur auf „negative medium“  
oder ggf. „negative small“ geändert werden

(verhindert zu schnelles „Zerfließen“ gesammelter Information;  
Regelbasen werden nur „vorsichtig“ verändert)

## Rekombination (Ein-Punkt-Crossover)

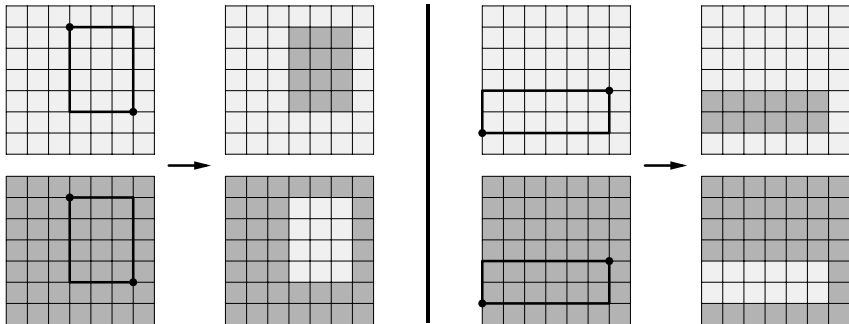
Wähle zufällig inneren Gitterpunkt der Tabelle und Ecke  
Tausche so definierte Teiltabellen zwischen zwei Eltern aus



**Beachte:** um bevorzugt benachbarte Regeln zusammen zu vererben, *solte* Crossover ortsabhängige Verzerrung zeigen!

## Rekombination (Zwei-Punkt-Crossover)

Wähle zufällig zwei Gitterpunkte der Tabelle (auch Randpunkte)  
Tausche so definierte Teiltabellen zwischen zwei Eltern aus



Zwei-Punkt-Crossover ist besser geeignet als Ein-Punkt-Crossover,  
da Teillösungen flexibler ausgetauscht werden können

# Optimierung der Fuzzy-Mengen

**geg.:** Optimierte Regelbasis mit fest gewählten und nicht veränderten äquidistanten (gleichmäßig verteilten) Fuzzy-Mengen

**ges:** Weitere Verbesserung des Reglerverhaltens durch Anpassen der Fuzzy-Mengen bei fester Regelbasis („Fine-Tuning“)

**Kodierung der Fuzzy-Mengen:** (erste Möglichkeit)

- wähle Form der Fuzzy-Mengen  
(z.B. Dreieck, Trapez, Gaußglocke, Parabel, Spline etc.)
- liste definierenden Parameter der Fuzzy-Mengen auf  
(z.B. Dreieck: linker Rand, Mitte, rechter Rand)

z.B. Stabbalance-Regler mit Dreieck-Fuzzy-Mengen (Ausschnitt):

...	nm			ns			az			ps			...
...	-45	-30	-15	-30	-15	0	-15	0	15	0	15	30	...

## Nachteile dieser Kodierung

**Kodierung** ist sehr „starr“ bzgl. der Form der Fuzzy-Mengen  
z.B. vorher festgelegt, ob Dreiecke oder Trapeze  
**genetische Operatoren** können Ordnung der Parameter  
zerstören

(bei Dreiecken muss z.B. gelten: links  $\leq$  Mitte  $\leq$  rechts)  
mögliche „Überholvorgänge“ zwischen Fuzzy-Mengen: durch  
Mutation und Crossover kann sinnvolle Reihenfolge der  
Fuzzy-Mengen zerstört werden (z.B. sollte gelten: ns rechts von  
ps)

**Bedingung** „Summe der Zugehörigkeitsgrade = 1“ wird u.U.  
verletzt (kann durch einmalige Darstellung identischer Parameter  
behandelt werden)

...	-45	-15	15	-15	15	30	15	30	45	30	45	60	...
...	-45	-30	-20	-30	-20	-10	-20	-10	0	-10	10	30	...

# Kodierung der Fuzzy-Partitionen

An Stelle von gleichmäßig über Wertebereich verteilten Stützstellen werden Zugehörigkeitsgrade der verschiedenen Fuzzy-Mengen angegeben

$$\underbrace{\begin{pmatrix} \mu_1(x_1) \\ \vdots \\ \mu_n(x_1) \end{pmatrix}}_{\text{Gen 1}} \cdots \underbrace{\begin{pmatrix} \mu_1(x_i) \\ \vdots \\ \mu_n(x_i) \end{pmatrix}}_{\text{Gen } i} \cdots \underbrace{\begin{pmatrix} \mu_1(x_m) \\ \vdots \\ \mu_n(x_m) \end{pmatrix}}_{\text{Gen } m}$$

Kodierung mit  $m \times n$  Zahlen aus  $[0, 1]$

pb	0	0	0	0	0	0	0	0	0	0	0	0	0	.5	1	1	1
pm	0	0	0	0	0	0	0	0	0	0	0	.5	1	.5	0	0	0
ps	0	0	0	0	0	0	0	0	0	.5	1	.5	0	0	0	0	0
az	0	0	0	0	0	0	0	.5	1	.5	0	0	0	0	0	0	0
ns	0	0	0	0	0	.5	1	.5	0	0	0	0	0	0	0	0	0
nm	0	0	0	.5	1	.5	0	0	0	0	0	0	0	0	0	0	0
nb	1	1	1	.5	0	0	0	0	0	0	0	0	0	0	0	0	0
	-60	-45	-30		-15		0		15		30		45		60		

# Genetische Operatoren

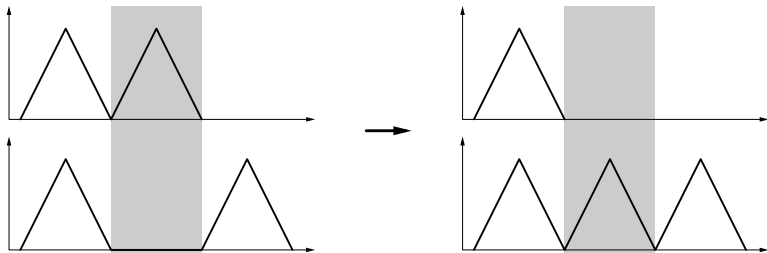
**Mutation:** analog zur 1-Bit-Mutation:

ein zufällig ausgewählter Eintrag wird zufällig geändert

sinnvoll: Größe der Änderung begrenzen, z.B. durch Festlegen eines Intervalls oder durch Normalverteilung

**Crossover:** Ein-Punkt- oder Zwei-Punkt-Crossover

beachte: Crossover kann Fuzzy-Mengen ausgelöscht werden



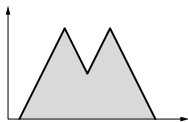
## Reparatur der Fuzzy-Mengen

Durch Mutation/Crossover: Zugehörigkeitsfunktionen der Fuzzy-Mengen u.U. nicht mehr *unimodal* (d.h.,

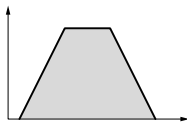
Zugehörigkeitsfunktion mit nur 1 lokales Maximum)

Multimodale Fuzzy-Mengen: viel schwerer zu interpretieren als unimodale

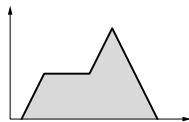
Fuzzy-Mengen werden ggf. repariert (unimodal gemacht)



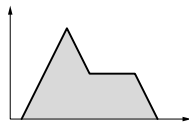
bimodale  
Fuzzy-Menge



Verbinden  
der Maxima



Abschneiden des  
linken Maximums



Abschneiden des  
rechten Maximums

Verbreitere/beschneide Fuzzy-Mengen, damit gesamter Wertebereich mit Fuzzy-Mengen überdeckt, aber nicht zu viele Fuzzy-Mengen gleichen Bereich abdecken



# Erzeugen/Optimieren von Fuzzy-Reglern mit GA

Erzeugen/Optimieren in zwei Schritten:

- Optimieren der Regelbasis bei festen Fuzzy-Partitionen
- Optimieren der Fuzzy-Partitionen mit erzeugter Regelbasis

In Experimenten: EA findet funktionsfähige Fuzzy-Regler fürs Stabbalance-Problem



Reglererzeugung per EA: sehr aufwendig (Rechenzeit)

Vorteil: kaum Hintergrundwissen benötigt

Weitere Anforderungen für Fitnessfunktion:

- **Kompaktheit:** geringe Anzahl von Regeln und Fuzzy-Mengen
- **Vollständigkeit:** Abdeckung relevanter Bereiche im Eingaberaum
- **Konsistenz:** keine Regeln mit sehr ähnlichem Antezedens und verschiedenem Konsequens
- **Interpretierbarkeit:** beschränkte Überlappung von Fuzzy-Mengen

# Literatur zur Lehrveranstaltung I

-  Byrne, J. (1995).  
*User Manual for the Base of Aircraft Data (BADA).*  
User Manual Revision 2.3, Issue 1, EUROCONTROL Experimental Centre.
-  Gerdes, I. (1994).  
Application of genetic algorithms to the problem of free routing for aircraft.  
In *Proceedings of the IEEE World Congress on Computational Intelligence: First IEEE Conference on Evolutionary Computation*, volume 2, page 536–541, Orlando, FL, USA.

## Literatur zur Lehrveranstaltung II



Gerdes, I. (1995).

Application of genetic algorithms for solving problems related to free routing for aircraft.

In Biethan, J. and Nissen, V., editors, *Evolutionary Algorithms in Management Applications*, page 328–340. Springer, Berlin, Germany.