Artificial Intelligence

Evolutionary Algorithms

Lesson 11:
Learning Fuzzy Control

Vincenzo Piuri

Università degli Studi di Milano

Generating/Optimizing Fuzzy Controls with Evolutionary Algorithms

- Mamdani-Assilian control can be optimized by changing
  - Rule base (which rules, which output)
  - Fuzzy sets / fuzzy partitions (shape, location, size, number of fuzzy sets)
  - t-norm or t-conorm for rule interpretation (rarely)
  - Parameters of the defuzzification method (if applicable; rarely)
  - Which input quantities to use for rules (feature selection)

Possible Approaches

1) Optimize rule base and fuzzy partitioning at the same time
   - Disadvantage: parallel optimization of many parameters
2) Optimize fuzzy partitioning with fixed rule base first, afterwards optimize rule base with best fuzzy partitioning
   - Disadvantage: expertise needed for creating rule base (choosing a random rule base is not promising)
3) Optimize rule base for fixed fuzzy sets first, afterwards fuzzy partitioning with fixed rule base
   - In this case: user needs to provide a fixed number of fuzzy sets per measured quantity and for control quantity

Fitness Function

- A good control system should satisfy the following criteria
  - Find the goal state for every potential situation
  - Reach the goal state as fast as possible
  - Find the goal state with minimum effort
- The control is used multiple times to control a test system. According to success of the rules (rule goodness) the control gets credits (number of situations, time till goal state, energy consumption)
- Note
  - Assessment of the individuals is the main task
  - Every individual needs to be simulated while controlling for a minimum number of time steps

Evaluating an Individual's Control Success

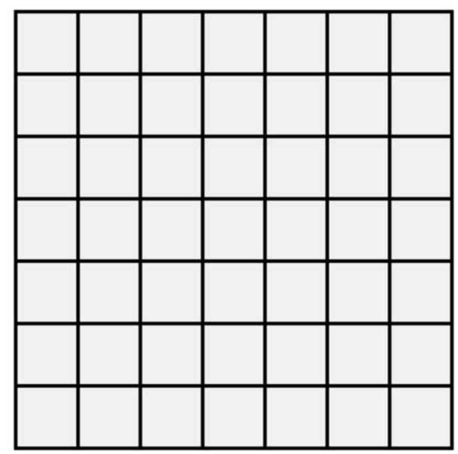
- Example: pole balancing controller
- Actual value differs from the goal state a lot: abort (failure)
  - E.g. pole balancing problem: actual value needs to be within [−90#, 90#]
- Actual value should get (and remain) close to the desire value after a certain time (range of tolerance)
  - Otherwise abort, too (failure)
- Range of tolerance is reduced with t (slowly leading to the goal state)
  - Within the first generations: it is sufficient to not knock down the pole
  - Later on: pole needs to stay vertically within a small angle
- Absolute values of the control quantity differences are summed up and used within a penalty term
  - In balanced state there is no difference in quick switching between high forces and controlling with small forces
  - High forces should be avoided

Generating/Optimizing the Rule Base: Coding

- Only complete rule bases (there must be a rule for every combination of input fuzzy sets)
- For every combination of input fuzzy sets only the term of the control quantity has to be determined (table filling)
- Example: rule base for pole balancing controller

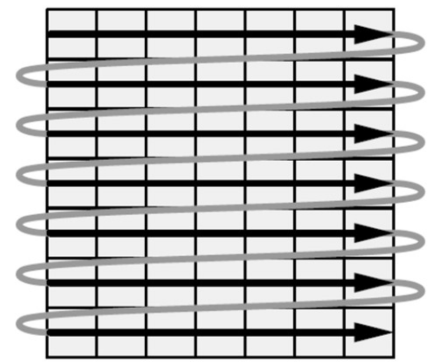
$\dot{\theta} \backslash \theta$	nb	nm	ns	az	ps	pm	pb
pb	**az**	**ps**	**ps**	**pb**	**pb**	**pb**	**pb**
pm	**ns**	**az**	**ps**	**pm**	**pm**	**pb**	**pb**
ps	**nm**	**ns**	**az**	**ps**	**pm**	**pm**	**pb**
az	**nb**	**nm**	**ns**	**az**	**ps**	**pm**	**pb**
ns	**nb**	**nm**	**nm**	**ns**	**az**	**ps**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

schematically



Coding of the Rule Base

- Linearization (conversion to vector)
  - Table is processed in arbitrary, but fixed order, and table entries are listed in a vector
  - Example: listing line by line
  - Problem: neighborhood relations between the rows are lost
    - Neighboring entries should have similar linguistical terms; this should be considered e.g. for crossover



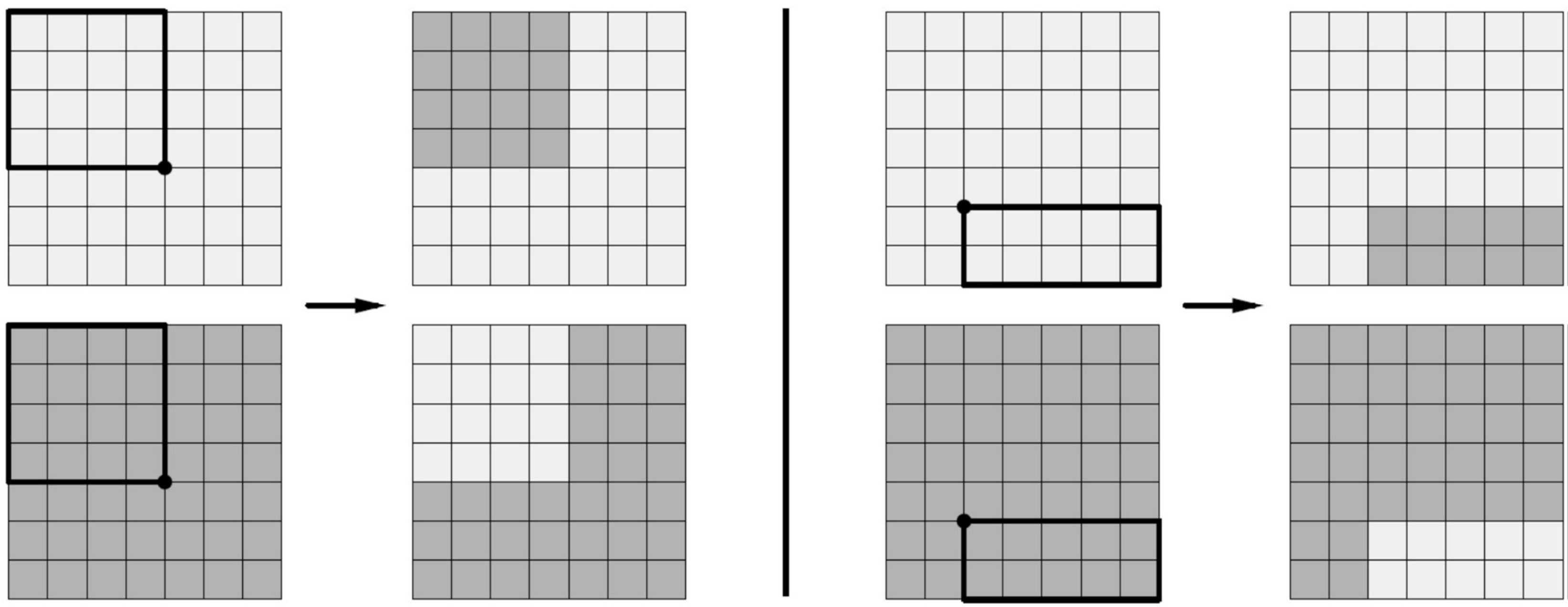
- Table (direct application of the scheme)
  - Two- or multi-dimensional chromosomes
  - Special genetical operators are needed

Variation (Standard Mutation)

- Rule/table entry is chosen randomly
- Linguistical term of the output is changed randomly
- If necessary multiple rules/table entries are changed simultaneously
- Limit mutation of a rule base so that table entries can only be changed to linguistic terms that are neighboring (or sufficiently close to) the current entry
- Example:
  - "positive small" can only be changed to "approximately zero" or "positive medium"
  - "negative big" can only be changed to "negative medium" or "negative small"
- Prevents fast scattering of learned information
  - Rule base is changed "with caution"

Recombination (One Point Crossover)

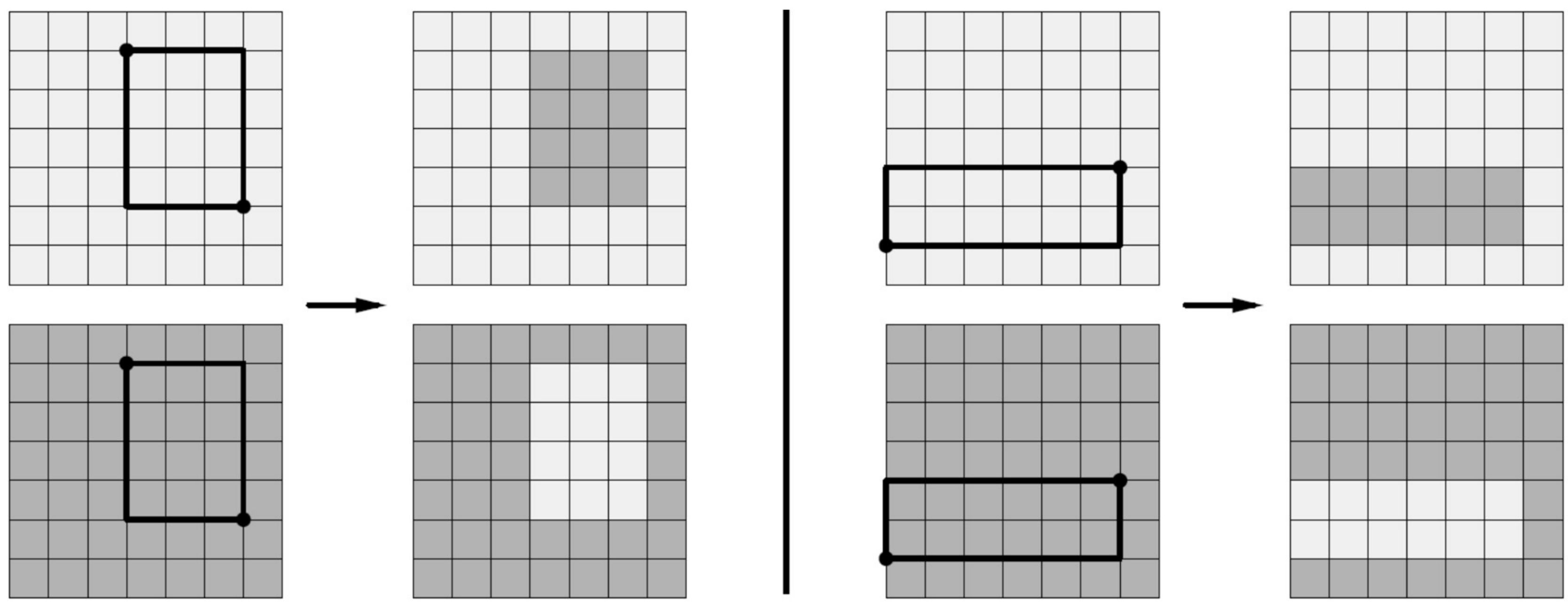
- Choose one inner grid point and a table corner randomly exchange sub-tables defined by these two points between parents



- Note: to prefer joint transfer ratio of neighboring rules, crossover should have positional bias

Recombination (Two Point Crossover)

- Choose two grid points randomly within the table (points lying on the border, too)
- Exchange the sub-table defined by these two points between parents



- Two point crossover is more appropriate, because subsolutions can be exchanged more flexibly

Optimizing the Fuzzy Sets

- Given: optimized rule base with fixed equidistantly distributed fuzzy sets
- Looking for: further improvements of the control unit by changing the fuzzy sets, with constant rule base (“Fine-Tuning”)
- Coding the fuzzy sets: (first option)
  - Choose shape of the fuzzy sets (e.g. triangle, trapezoidal, gaussian, parabola, spline, etc.)
  - List defining parameters of the fuzzy sets (e.g. triangle: left margin, center, right margin)

- E.g. pole balancing control with triangle fuzzy sets (excerpt)

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

Disadvantages of this Coding

- Coding is very “inelastic” regarding the shape of the fuzzy sets
  - E.g. previously fixed, whether triangles or trapezoids are used
- Genetical operators can destroy ordering of the parameters
  - E.g. for triangles “left * center * right‟ must hold
- Fuzzy sets “overtaking” each other possibly
  - Meaningful ordering amongst the fuzzy sets might be ruined by mutation or crossover
  - E.g. “ns right of ps” should be true
- Condition “sum of membership degrees = 1” might be violated
  - Can be corrected by unique representation of identical parameters

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

Coding of the Fuzzy Partitions

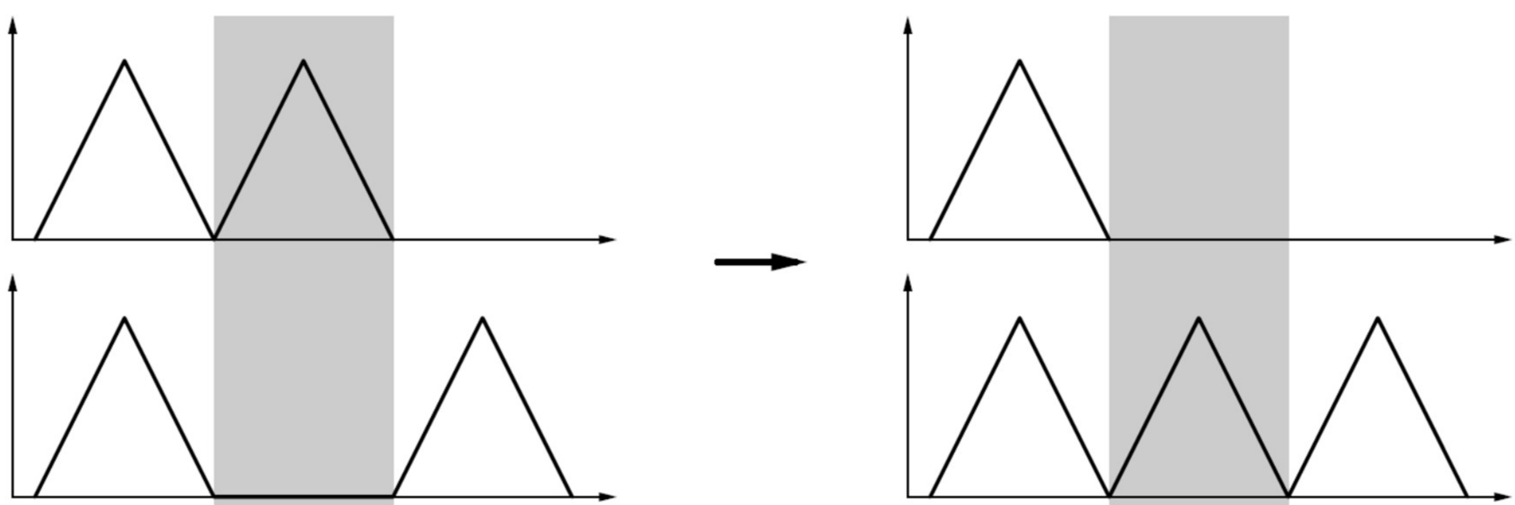
- Instead of equally distributed sampling points amongst the domain, the membership degrees of different fuzzy sets are given

$$\underbrace{\overbrace{\begin{pmatrix} \mu_1(x_1) \\ \vdots \\ \mu_n(x_1) \end{pmatrix}}^{\text{Gen } 1} \cdots \overbrace{\begin{pmatrix} \mu_1(x_i) \\ \vdots \\ \mu_n(x_i) \end{pmatrix}}^{\text{Gen } i} \cdots \overbrace{\begin{pmatrix} \mu_1(x_m) \\ \vdots \\ \mu_n(x_m) \end{pmatrix}}^{\text{Gen } m}}_{\text{coding with } m \times n \text{ numbers from } [0,1]}$$

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

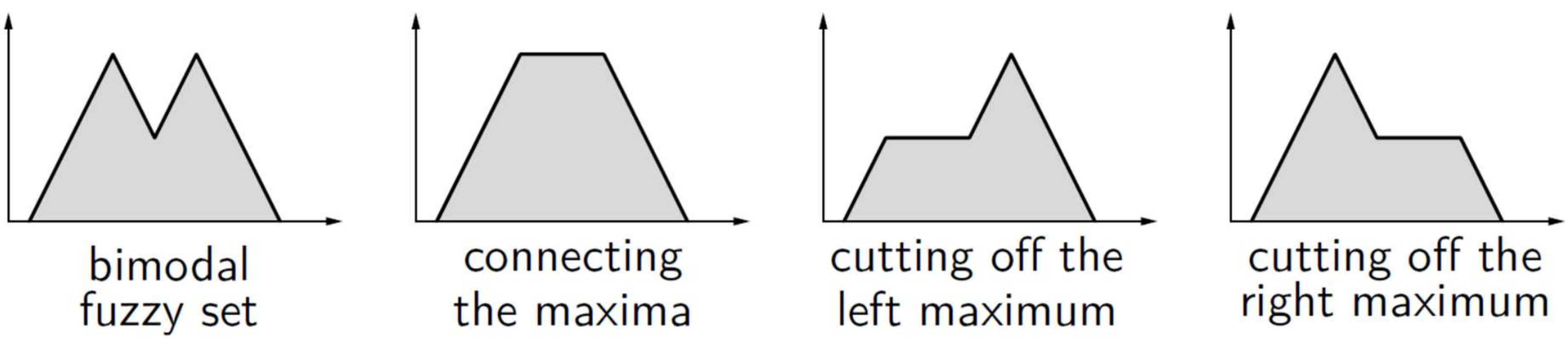
Genetical Operators

- Mutation: analogue to 1 bit mutation
  - A randomly chosen entry is randomly manipulated
  - Useful: limit the range of change, e.g. by defining an interval or a normal distribution
- Crossover: 1-point or 2-point crossover
  - Note: crossover can extinguish fuzzy sets



Repairing Fuzzy Sets

- By mutation/crossover
  - Membership function of fuzzy sets might not be unimodal anymore (a unimodal membership function has 1 local maximum only)
- Multimodal fuzzy sets
  - Much more difficult to interpret, than unimodal fuzzy sets
- If feasible, fuzzy sets are repaired
  - To make them unimodal



- Extend / cut fuzzy sets in a way that they cover the total domain, but no point is covered by too many fuzzy sets

