OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

INF

FAKULTÄT FÜR
INFORMATIK

Evolving Fuzzy Systems

Prof. Dr. Rudolf Kruse

. . . are based on *biological theory of evolution*.

Fundamental principle:

- Beneficial properties created by random variation are chosen by natural selection.
- Differential reproduction: Individuals with beneficial properties have better chances to reproduce themselves.

The theory of evolution describes the diversity and complexity of species.

It allows to unify all disciplines of biology.

Fundamental Terms and their Meanings I

<table>
<tr><th>Term</th><th>Biology</th><th>Computer Science</th></tr>
<tr><td>individual</td><td>living creature</td><td>candidate solution</td></tr>
<tr><td rowspan="3">chromosome</td><td>DNA histone protein string</td><td>character string</td></tr>
<tr><td colspan="2">defines “blueprint” and (partly)
properties of individual in coded form</td></tr>
<tr><td>usually several chromosomes
per individual</td><td>usually one chromosome
per individual</td></tr>
<tr><td rowspan="2">gene</td><td>part of chromosome</td><td>one character</td></tr>
<tr><td colspan="2">fundamental unit of inheritance that (partly)
defines property of an individual</td></tr>
<tr><td rowspan="2">allele
(allomorph)</td><td>characteristic of a gene</td><td>value of a character</td></tr>
<tr><td colspan="2">there is only one characteristic of a gene per chromosome</td></tr>
<tr><td rowspan="2">locus</td><td>position of a gene</td><td>position of a character</td></tr>
<tr><td colspan="2">exactly just one gene on every position in chromosome</td></tr>
</table>

Fundamental Terms and their Meanings II

Term	**Biology**	**Computer Science**
phenotype	outer appearance of a living creature	implementation of a candidate solution
genotype	genetic constitution of a living creature	encoding of a candidate solution
population	all living creatures	family/multiset of chromosomes
generation	population at given time	population at given time
reproduction	creation of offspring from one or two (if, then usually two) creatures	creation of (child) chromosomes from one or more (parent) chromosomes
fitness	capability/conformity of a living creature	capability/goodness of a candidate solution
	determines chances of both survival and reproduction	

Elements of an Evolutionary Algorithm I

An EA consists of

a **codebook** for the candidate solutions

- This codebook is problem-specific (*i.e.* no general rules exist).

a method to generate an **initial population**

- Usually, it generates random character strings.
- Depending on the encoding, sophisticated approaches can be necessary.

a **weighting function** (fitness function) for individuals

- It acts as an environment and indicates the goodness of the individuals.

a **technique of sampling** based on the fitness function

- It defines both which individuals are used to generate offspring and which ones reach the next generation without any change.

Elements of an Evolutionary Algorithm II

Furthermore, an EA consists of

genetic operators that change candidate solutions

- *mutation* (random modification of particular genes)
- *crossover* (recombination of chromosomes; actually “crossing over” due to process in meiosis (period of cell division) where chromosomes are divided and then joined crossed-over again)

values for several **parameters**

- *e.g.* population size, mutation probability, etc.

termination criterion, *e.g.*

- fixed number of generations has been processed
- for fixed number of generations, no improvement occurred
- predefined minimal goodness of solution has been reached

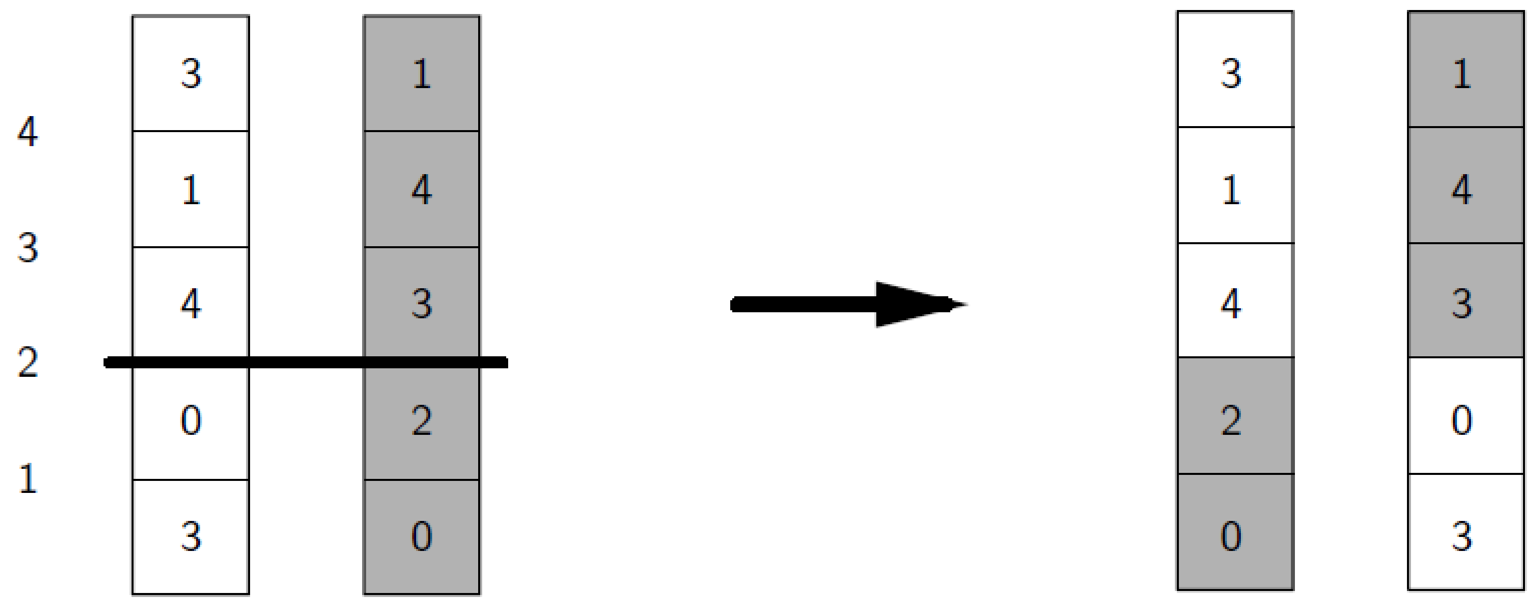
Generic Basic EA

Algorithm 1 EA Schema

$t \leftarrow 0$
pop(t) $\leftarrow$ create population of size μ
evaluate pop(t)
while termination criterion not fulfilled
{
  $pop_1 \leftarrow$ select parents for λ offspring from pop(t)
  $pop_2 \leftarrow$ generate offspring by recombination from pop_1
  $pop_3 \leftarrow$ variate individuals in pop_2
  evaluate pop_3
  $t \leftarrow t + 1$
  pop(t) $\leftarrow$ select μ individuals from $pop_3 \cup pop(t-1)$
}
return best individual from pop(t)

Exchange of one part of chromosome (or subset of selected genes) between two individuals, *e.g.* so-called **one-point crossover**:

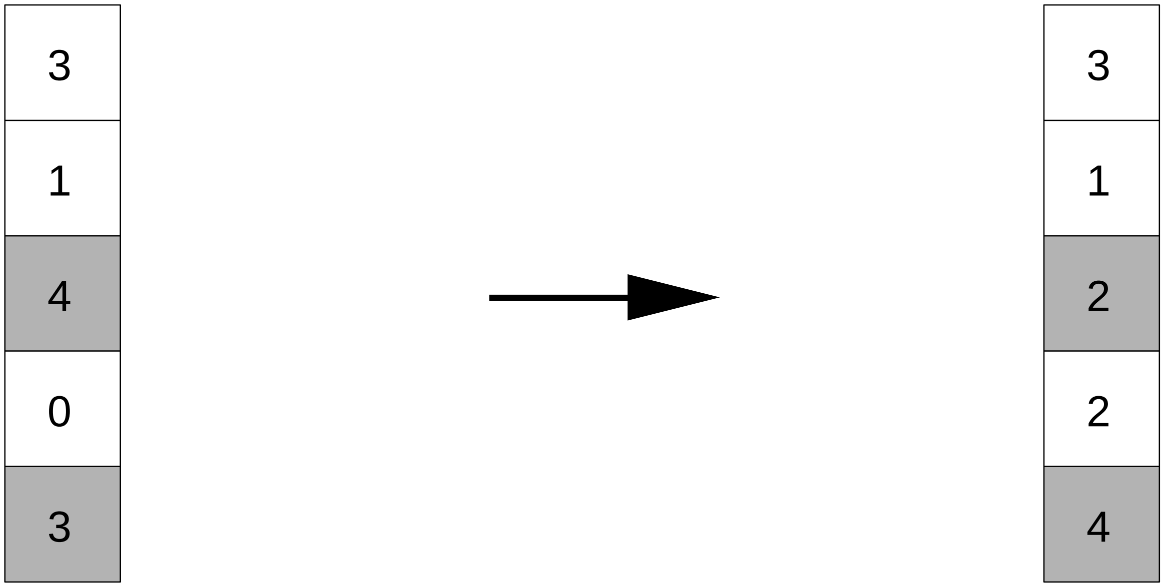
1. Choose a random point of division between two genes.
2. Exchange the gene sequences of one side of the division point.



Genetic Operators: Mutation

Randomly chosen genes are randomly replaced (alleles change).

The number of replaced genes maybe random, too (should be small).



Most mutations are damaging/harmful (they worsen the fitness).

Initially non-existent alleles can (only) be generated by mutation.

Optimization of Fuzzy Controllers with GAs

Mamdani controller can be induced/optimized as follows:

- rule base (which rules, which outputs)
- fuzzy sets/fuzzy partitions (shape, location, width, number of fuzzy sets
- *t* -norm or *t* -conorm for rule evaluation (rarely)
- parameters of defuzzification method (if present, rarely)
- inputs used in the rules (feature selection)

Here we talk about the optimization of the rule base and the fuzzy sets with a fixed choice of input values.

Rule evaluation: minimum and maximum

Defuzzification: center of gravity (COG)

Optimization: Three Possible Approaches

1. Rule base and fuzzy partitions are optimized simultaneously:
    - Disadvantage: Many parameters must be optimized at same time.
2. First, the fuzzy partitions are optimized *w.r.t.* given rule base, then rule base is optimized with best fuzzy partitions:
    - Disadvantage: Expert knowledge needed to create rule base (starting with random rule base is not promising).
3. First, rule base is optimized *w.r.t.* given fuzzy sets, then fuzzy partitions are optimized with best rule base:
    - Fuzzy sets may be distributed, *e.g.* equidistantly.
    - Here, the user must specify the number of fuzzy sets for input and output.
    - We only consider this approach here.

Optimization: Fitness Function

A good controller should exhibit several properties:

- The target value should be reached from any (initial) situation.
- The target value should be reached quickly.
- The target value should be reached with minimal effort (energy).

The controller is applied multiple times to target system:

- Here, simulation of inverted pendulum problem.
- Several randomly chosen starting points.
- A score is assigned to the controller according to its success (number of situations, duration of successful control, energy costs).

Here, the evaluation of individuals is by far the most expensive operation:

- Every individual must be put in control for at least certain number of time steps in order to yield reliable fitness score.

Assessing the Controller Success

If the deviation of the actual value to the target is big, it’s a failure, *e.g.* inverted pendulum: It must stay within $[-90^\circ, 90^\circ]$.

After some time, the actual value should be close to the target value and remain within its proximity (range of tolerance). Otherwise the process is aborted also (failure).

The range of tolerance is shrinked during generations (towards target).

- During the first generations, it suffices if the pendulum doesn’t topple over.
- Later the pendulum must stand upright within a shrinking angle interval.

The abs. values of adjusting values are added up as penalty value.

In balance, a fast switch betw. large/small forces effects the controller.

Thus large forces need to be avoided.

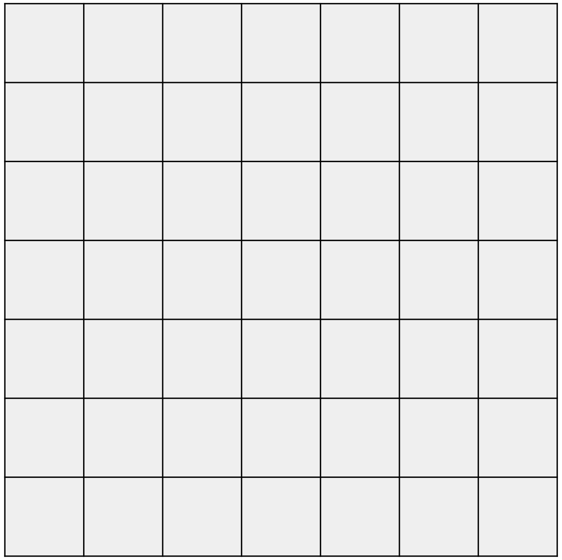
We only consider complete rule bases, that is, one rule for every combination of input fuzzy sets.

For every combination of input fuzzy sets, we need to determine a linguistic term of the control variable (by filling a table).

Example rule base for inverted pendulum 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

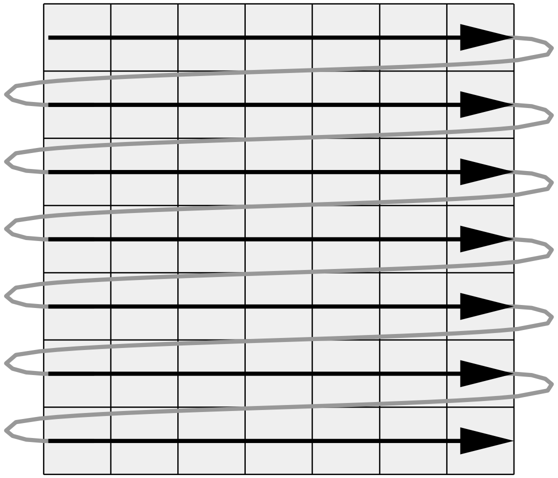
schematic



Representing the Rule Base as Chromosome

Linearization (transformation into a vector):

The table is traversed in an arbitrary but fixed order.



Entries are listed in a vector, *e.g.*

- listing row by row.
- So, neighboring relations between cells are lost.

Adjacent entries should contain similar linguistic terms (this is important, *e.g.* during crossover).

Table (direct usage of scheme)

for two- or higher dimensional chromosomes.

Thus special genetic operators are needed.

Genetic Operators for Rule Base: Mutation

1. The rule/table entry is chosen randomly.
2. The linguistic term of output is altered randomly.
3. Multiple table entries may be altered.

It might be beneficial to restrict the mutation of the rule base:

An entry is only changed to a linguistic term similar to the current one, *e.g.*

- “positive small” $\mapsto$ “approximately zero” or “positive medium”,
- “negative big” $\mapsto$ “negative medium” or “negative small”.

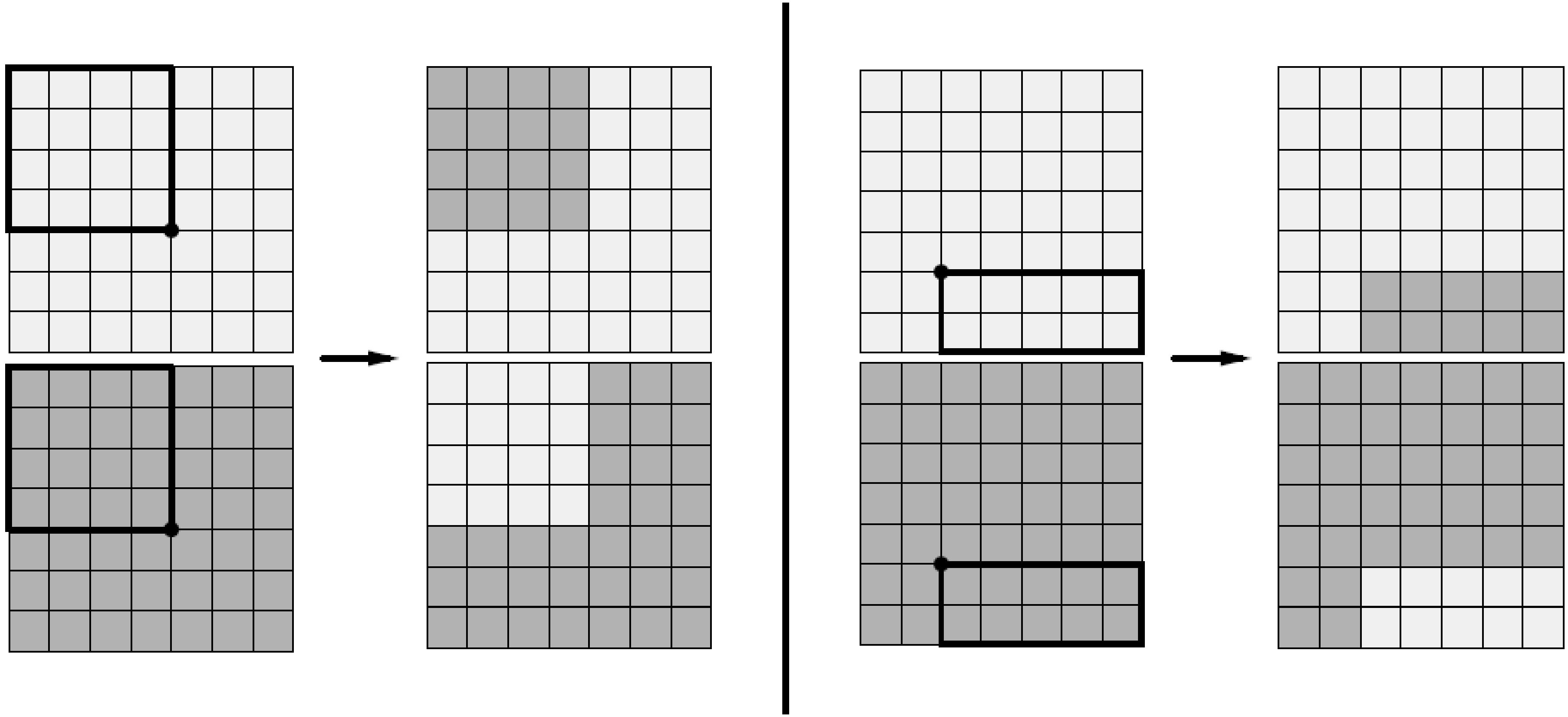
This prevents a too fast “depletion” of the collected information.

Also, the rule bases are only modified “carefully”.

Genetic Operators: One-Point Crossover

Choose both interior grid point and corner randomly.

This defines a subtable that will be exchanged between two parents.



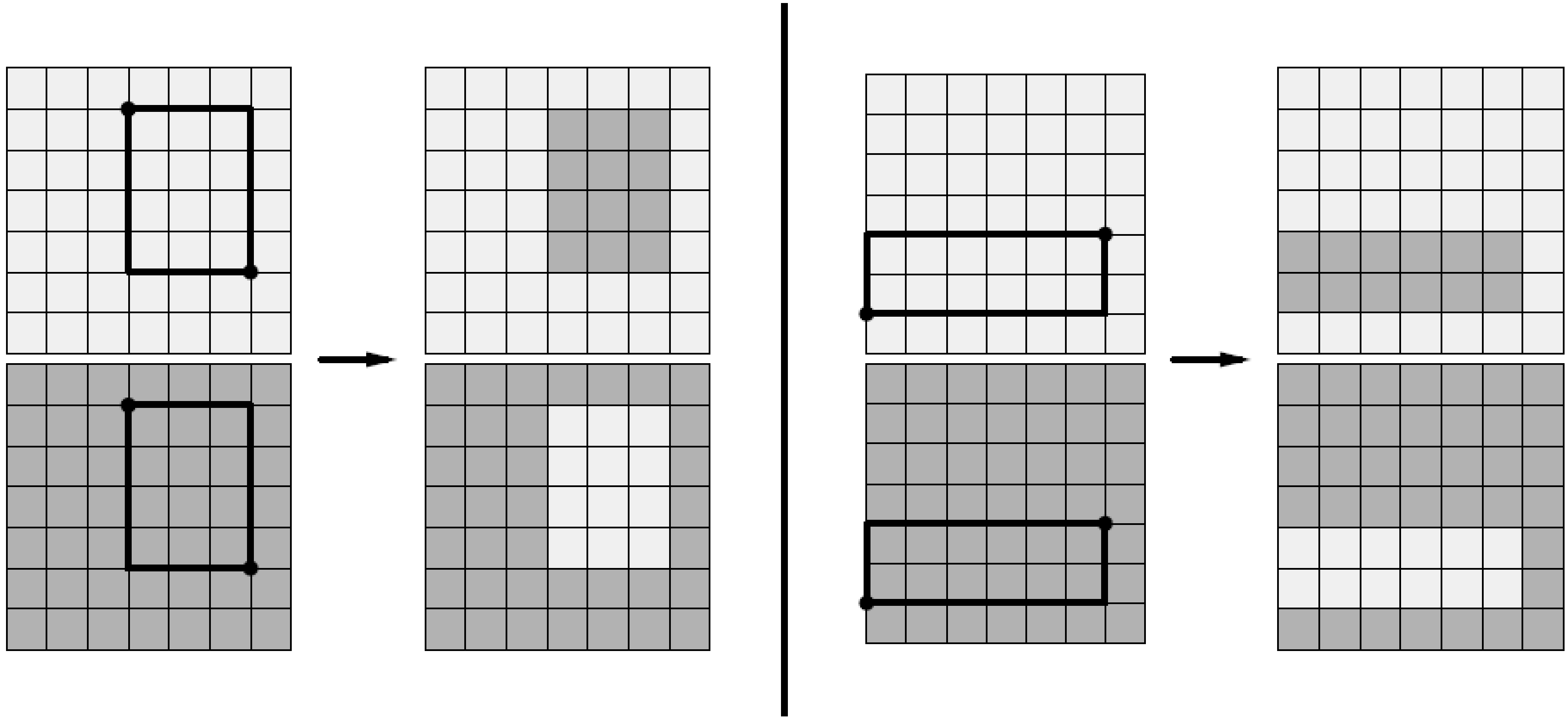
The crossover should exhibit a location-dependent bias.

So, it prefers to inherit adjacent rules.

Genetic Operators: Two-Point Crossover

Choose two grid points randomly (border points allowed).

This defines a subtable that will be exchanged between two parents.



Partial solutions can be exchanged in a more flexible way.

The two-point crossover performs better than one-point crossover.

Optimizing Fuzzy Sets

Given: optimized rules with fixed and unchanged equidistant fuzzy sets.

Goal: further improvement of rule behavior by adjusting fuzzy sets with fixed rule base (“fine tuning”).

Encoding fuzzy sets (first possibility):

1. Choose the shape of the fuzzy sets (*e.g.* triangle, trapezoid, Gaussian, parabola, spline, etc.).
2. List the defining parameters of the fuzzy sets (*e.g.* triangle: left border, center, right border).

Example: controller with triangular fuzzy sets (excerpt)

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

Optimizing Fuzzy Sets: Disadvantages

In advance one must specify that, *e.g.* triangles or trapezoids are used.

So, the encoding is quite "rigid" *w.r.t.* the shape of the fuzzy sets.

Genetic operators can violate the order of parameters, *e.g.*

- considering triangles $\mu_{l,m,r}$, $l \leq m \leq r$ must hold.

A possible "overtaking" between fuzzy sets, *i.e.*

- the meaningful order of the fuzzy sets may be destroyed by mutation/crossover, *e.g.* it should hold true that *ns* lies left of *ps*.

The condition $\forall x : \sum_i \mu_i(x) = 1$ might be violated:

- This can be treated by representing identical parameters only once.

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

Encoding Fuzzy Partitions

Distinct membership degrees of fuzzy sets are specified for a predefined and equidistant set of sampling points:

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

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

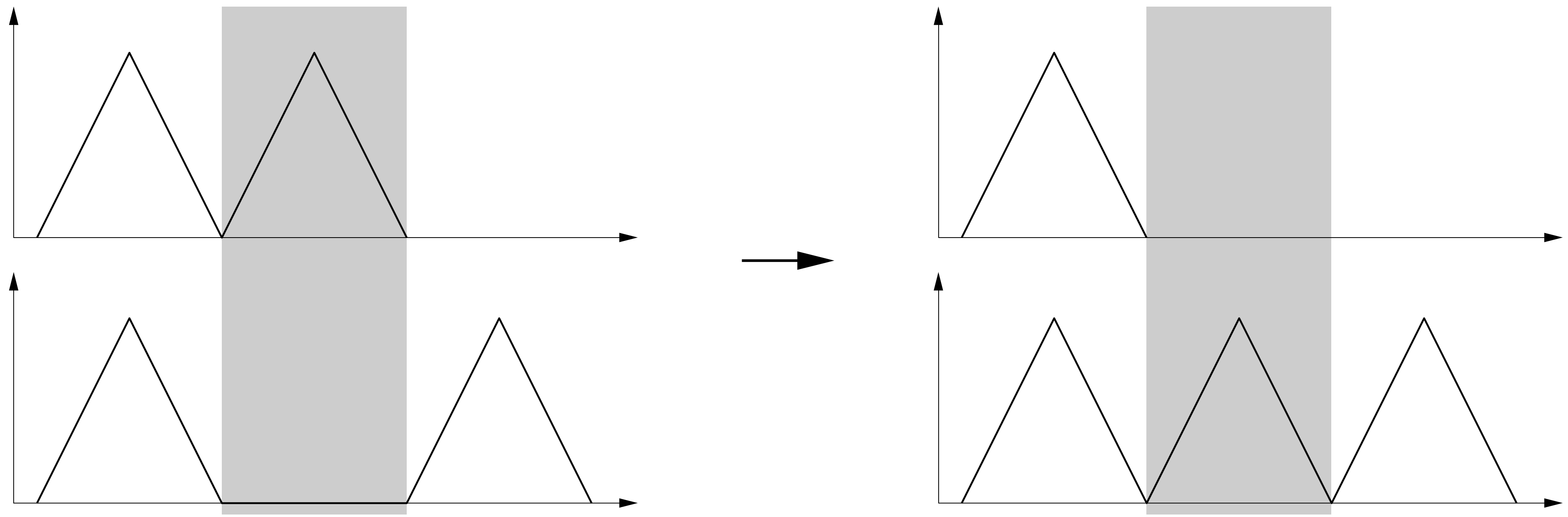
Genetic Operators

Mutation (analogous to standard mutation):

- A randomly chosen entry is altered randomly.
- It is reasonable to restrict the magnitude of alteration, *e.g.* by specifying an interval or by a normal distribution.

Crossover (basic one-/two-point crossover):

Note that fuzzy sets may be erased by crossover.

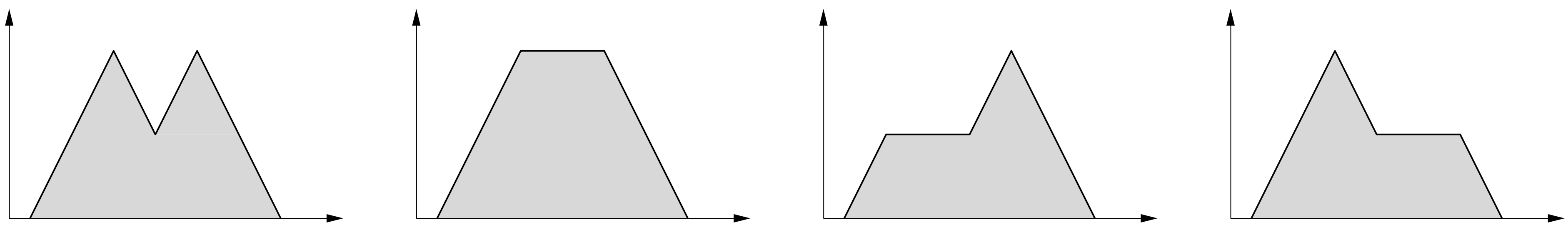


Repairing Fuzzy Sets

Mutation/crossover may violate the *unimodality* of the fuzzy set,

- *i.e.* the membership function may have ≥ 1 local maximum.
- Multimodal fuzzy sets are harder to interpret than unimodal ones.

Therefore, fuzzy sets are repaired (made unimodal).



bimodal fuzzy set | join the maxima | cut off left maximum | cut of right max.

Fuzzy sets may be widened or cut off such that the entire domain is covered, but without having too many fuzzy sets covering the same areas.

Summary

Optimization in two steps:

1. Optimize the rule base with fixed fuzzy partitions.
2. Optimize the fuzzy partitions with an induced rule base.

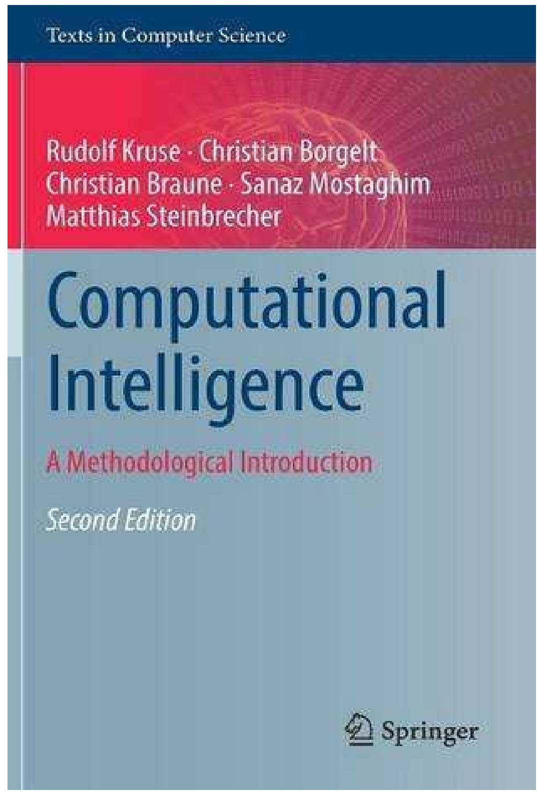
It was possible to generate a working rule set for the inverted pendulum problem with evolutionary algorithms.

The approach of inducing a rule base is quite expensive in time.

However, it does not need any background knowledge.

Additional requirements might be considered for fitness:

- *compactness* (small number of rules and fuzzy sets)
- *completeness* (coverage of relevant areas of input space)
- *consistency* (no rules with similar antecedents and different consequents)
- *interpretability* (limited overlapping of fuzzy sets)


Texts in Computer Science
Rudolf Kruse · Christian Borgelt
Christian Braune · Sanaz Mostaghim
Matthias Steinbrecher
Computational Intelligence
A Methodological Introduction
Second Edition
Springer