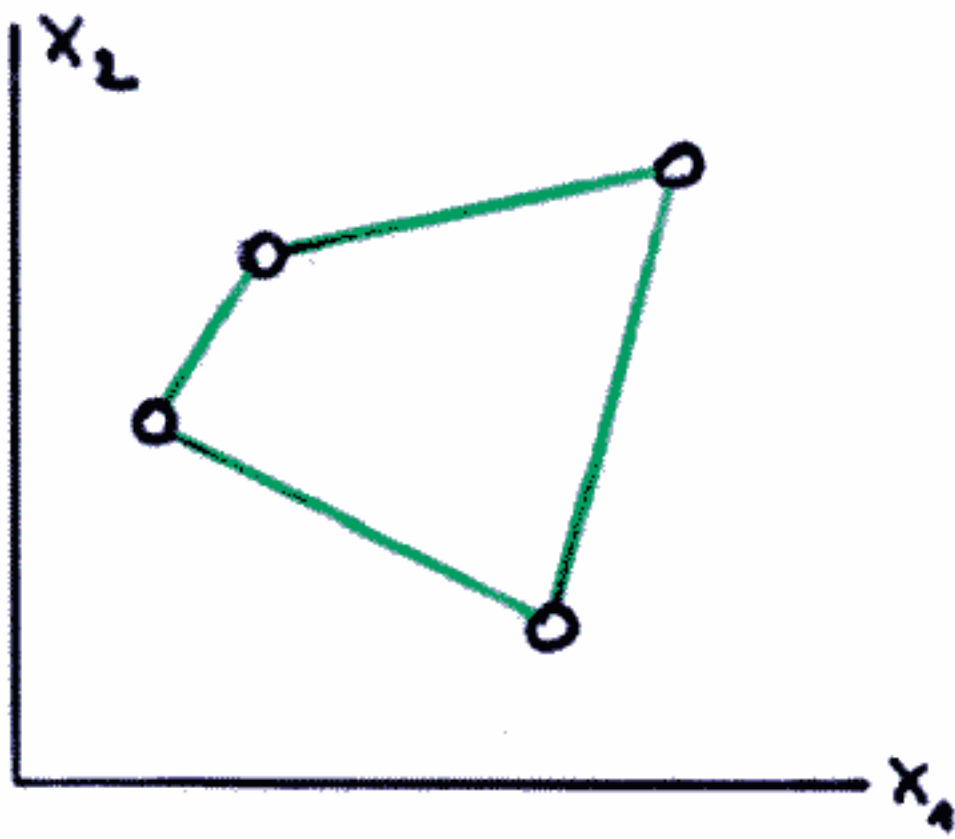
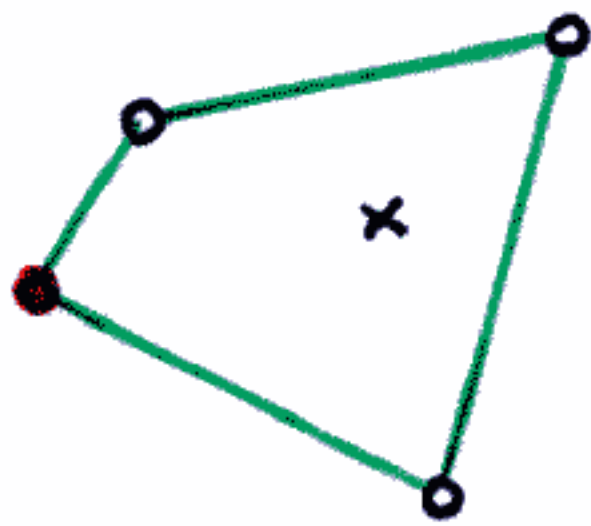


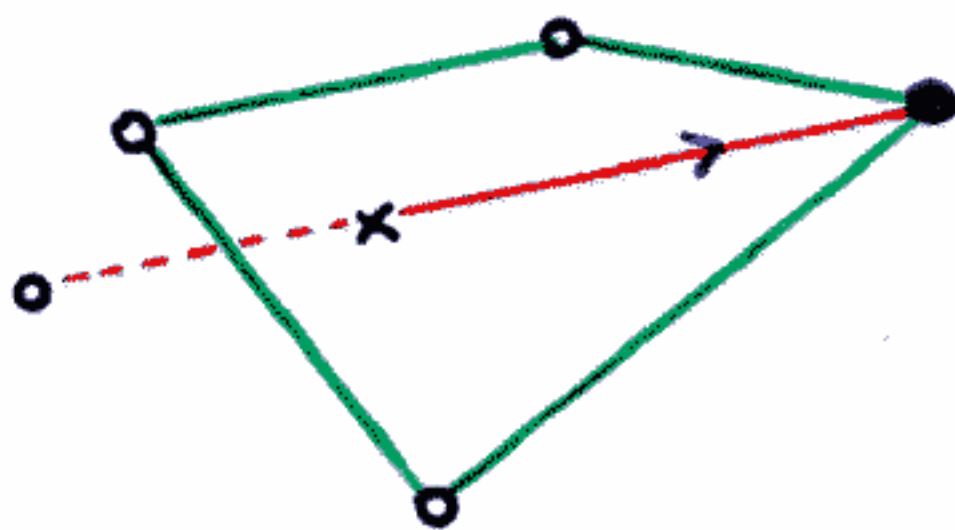
Complex - Strategie von M.J. Box (1965)



Ausgangspolyeder mit
 $N \geq m+1$ Ecken
 hier: $m=2$; $N=4$

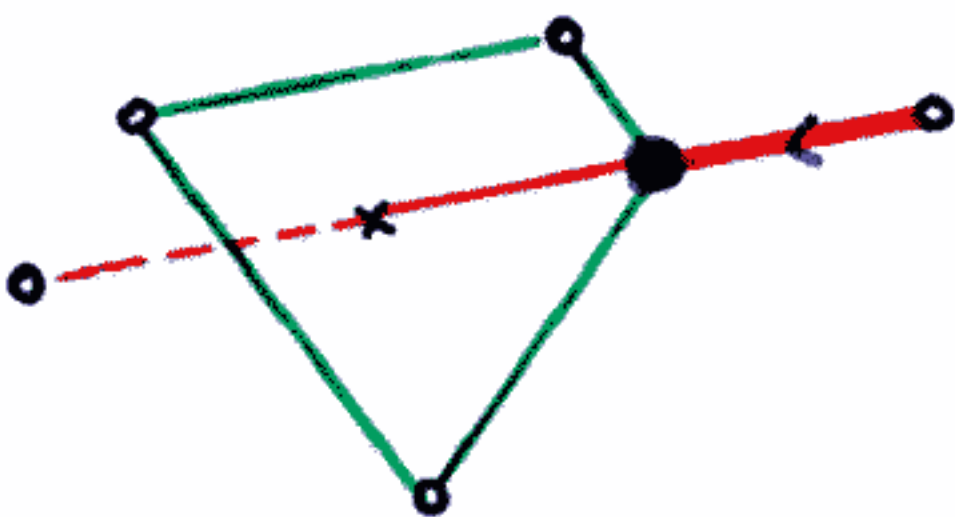


Bestimmung der schlechtesten
 Ecke und des Schwerpunkts
 der übrigen Ecken \bullet x



Spiegelung von \bullet an x
 mit Reflektionsfaktor > 1

bei Erfolg: neue Ecke ersetzt
 schlechteste



bei Mißerfolg:
 Rücksetzung auf die Hälfte
 zwischen neuer Ecke und
 Schwerpunkt

Abschaltregel: 5 aufeinander folgende erfolglose
 Rücksetzungen

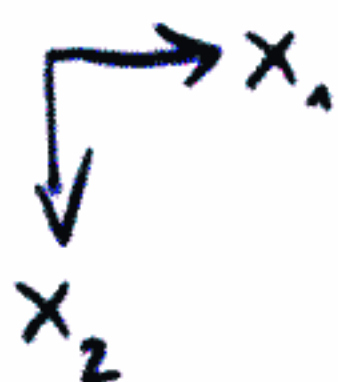
Nebenbedingungen: wenn verletzt: Rücksetzung

Suche nach zuläss. Anfangslösung über Ersatz-
 Zielfunktion möglich

EVOP

evolutionary operation

G.E.P. Box



x

x



x



Δ Start

x 1. Versuchsreihe

⊗ bester Versuch

□ 2. Versuchsreihe

full factorial design

$$2^n + 1$$

$$2^n - 1$$

 Start
 Versuche/Runde
 später

Fractional factorial design

$$2^m - 1$$

$$m < n$$

" später

gute Idee: nicht alle mögl. ausprobieren, sondern
sequentiell an Opt. herantasten

($2^m - 1$ Tests können gleichz. durchgeführt werden)

REVOP random EVOP (Satterthwaite)

nur $n+1$ zufällige Proben / Versuchsreihe

? Evolution

1 Elter, 2^m Nachkommen
Elter kann überleben

$$(1 + 2^m) - E.S.$$

Weitere Varianten: Simplex-Strat. Nelder + Mead

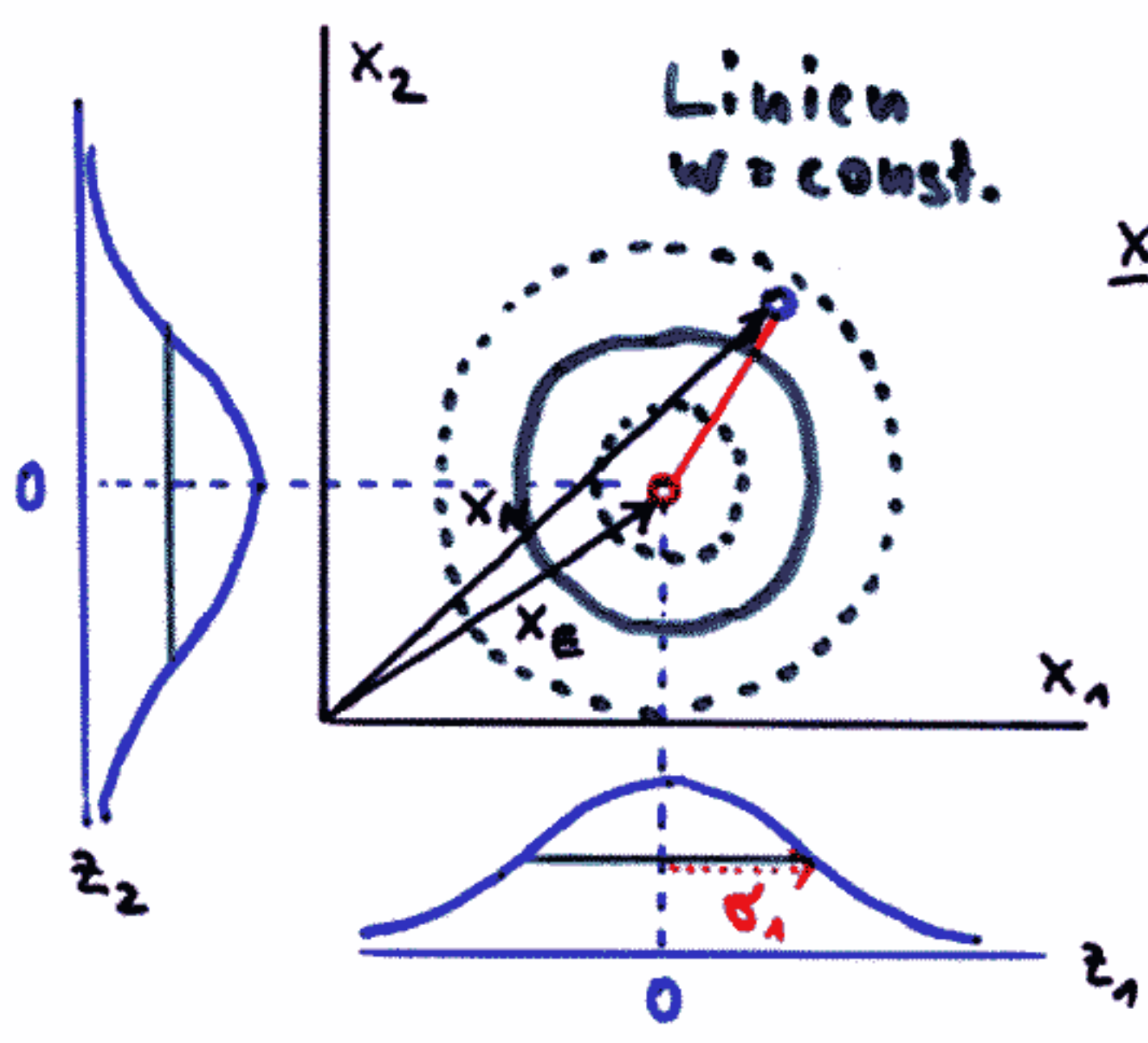
Complex-Strat. M.J. Box

(1+1) - Evolutionsstrategie (Rechenberg, 1970)

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} + s^{(k)} \cdot \underline{v}^{(k)}$$

Richtung : zufällig

„mittlere“ Schrittweite : anhand Erfolgsrate gesteuert



$$\underline{x}_N^{(k)} = \underline{x}_E^{(k)} + \underline{z}^{(k)}$$

Mutation
Nachkomme Elter Zufallsänderung

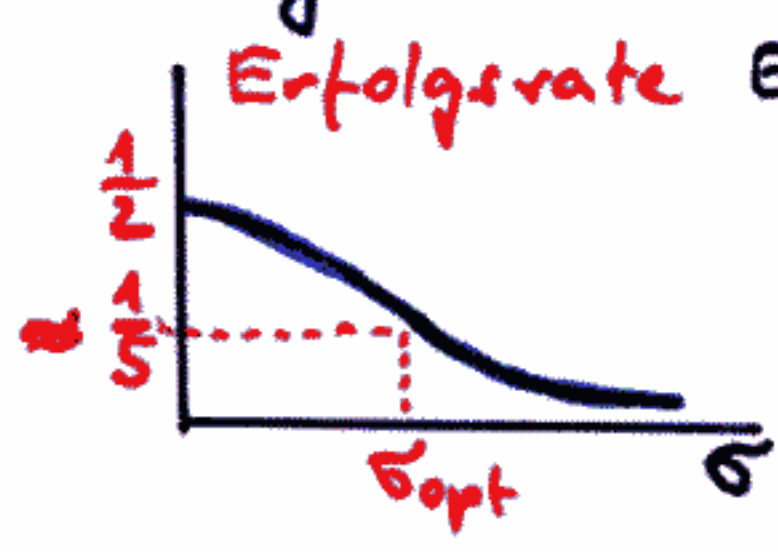
$$w(z_i) = \frac{1}{\sqrt{2\pi} \cdot \sigma_i} \exp\left(-\frac{z_i^2}{2\sigma_i^2}\right) \quad \text{Normalverteilung}$$

Selektion:

$$\underline{x}_E^{(k+1)} = \begin{cases} \underline{x}_N^{(k)} & \text{bei Verbesserung} \\ \underline{x}_E^{(k)} & \text{sonst} \end{cases}$$

(Zielf. wert besser und all Neb. erfüllt)

Schrittweitensteuerung:



$\sigma \uparrow$ wenn $E. > \frac{1}{5}$
 $\sigma \downarrow$ wenn $E. < \frac{1}{5}$

(1+1) : 1 Elter, 1 Nachkomme
 Auslese : Elter oder Nachkomme

Evolutionäre Algorithmen EAs

Evolutionary Computation EC

- Sammelbegriff für verschiedene Ansätze, natürliche (biologische, Lebens-) Entwicklungsprozesse algorithmisch nachzuahmen
- Grundideen aus den 60er Jahren
- seit etwa 1985 : exponentiell wachsend, wobei erfolgreiche Anwendungen maßgebend
- jetzt auch zunehmende Akzeptanz in Computer Science / Informatik
- zusammen mit anderen nicht-KI Methoden subsymbolisch

NN (künstliche neuronale Netzwerke)

FL Fuzzy-Logik

AL Artificial Life

AA autonome Agenten

CA zelluläre Automaten

noch vorläufige Sammelbegriffe:

soft computing

Computational Intelligence

intelligente hybride Algorithmen

Bioinformatik / Neuroinformatik

naturanaloge parallele Problemlösungs-Strategien

....

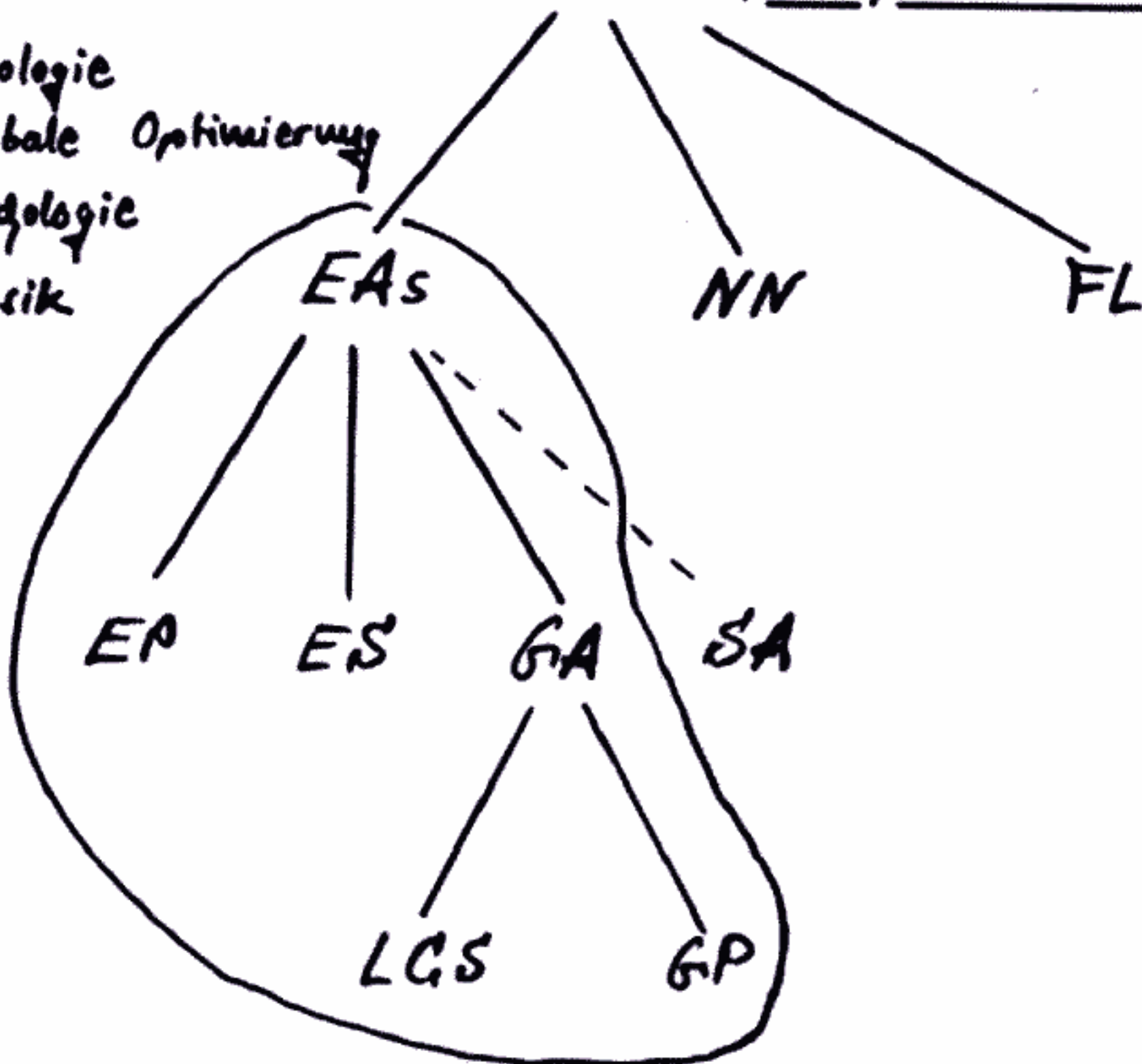
Darwin-Strategien
1859: The origin of species

Informatik

KI

CI: „Computational Intelligence“

- Biologie
- Globale Optimierung
- Psychologie
- Physik



Konferenzen:

ICGA	85, 87, 89, 91, 93
PPSN	90, 92, 94
FOGA	90, 92, 94
EP	92, 93, 94
IEEE-EC	94

Journals:

Evolutionary Computation
 Complex Systems (ab '94)
 Bio Systems

EAs

- sehr stark vereinfachte Nachahmung (unterhalb Einzeller)
- Begriff 'Evolution' offenbar emotionsbeladen
 ↘ falsche Modellvorstellung / Modellierung
 z.B. es gibt keine Evolution
 Evolution = Monte Carlo Strategie
 Evolution = stetiger, monotoner Anstieg der Fitness
- Hintergrundgedanken
 z.B. zeigen, daß Ev. nicht funktionieren kann
 besseres Verständnis für natürliche Prozesse (vgl. Gehirn / NN)
- nach wie vor 'philosophische' Unterschiede der 3 existierenden 'Schulen'

EP Evolutionary Programming

GA Genetic Algorithms GAs

GP Genetic Programming

ES Evolutionsstrategien ESs

Σ EAs Evolutionary Algorithms
 EC Evolutionary Computation

Evolutionäre Algorithmen (EA)

heute im wesentlichen 3 Gruppen

- Evolutionary Programming (EP)

D. Fogel

1966 : L. Fogel, Artificial Intelligence through Simulated Evolution

- Genetic Algorithms (GAs)

J. Holland, ... De Jong, Grefenstette ...

1975 : J. Holland, Adaptation in Natural and Artificial Systems

- Evolutionsstrategien (ESs)

I. Rechenberg, ... Schwefel

1971/73 : Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution

(Vorläufer, Seitenzweige)

seit 1985 : ICGA

-Konferenzen (alle 2 Jahre)

seit 1990 : PPSN

auch SA, IN, ...

April 1993 : Evolutionary Computation Int'l Journal (MIT Press)

Sommer 1994 : Weltkongress WCCI



adaptiver Automat

CS, GP
kombin. Opt.

Experimentelle Optimierung

wichtige Beobachtungen:

Individuen (zu einem Zeitpunkt) verschieden
phänotypische Diversität

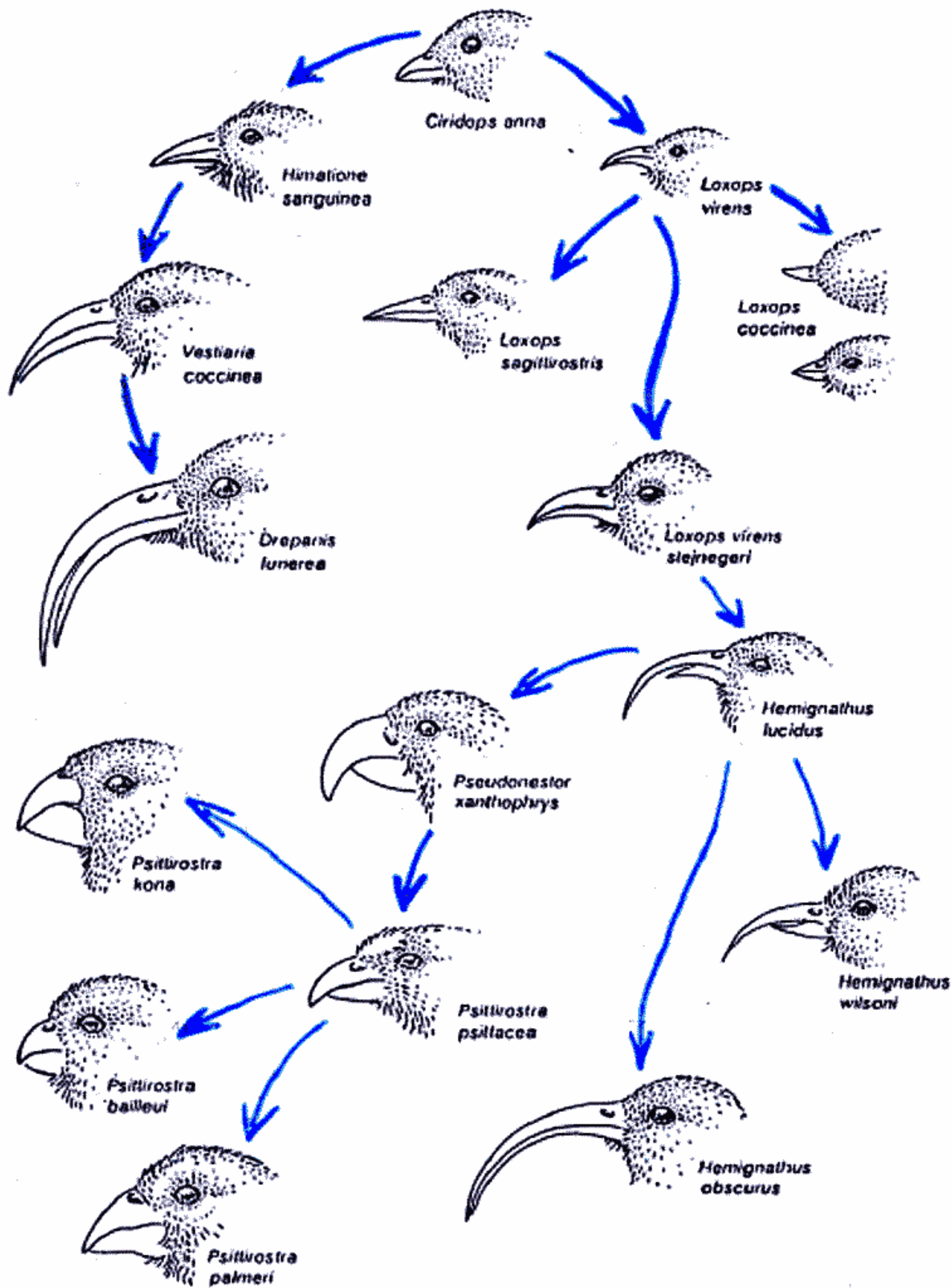
Individuen haben endliche Lebensdauer

Individuen haben Nachkommen (nicht alle),
die etwas verschieden von Vorgängern sind.
phänotypische Variabilität

über längere Zeiträume (Generationen)
bzw. unter verschied. Umweltbedingungen
b.z.B. auf Inseln mit verschied. Lebensbedingungen):
erhebliche Veränderungen / Anpassungen

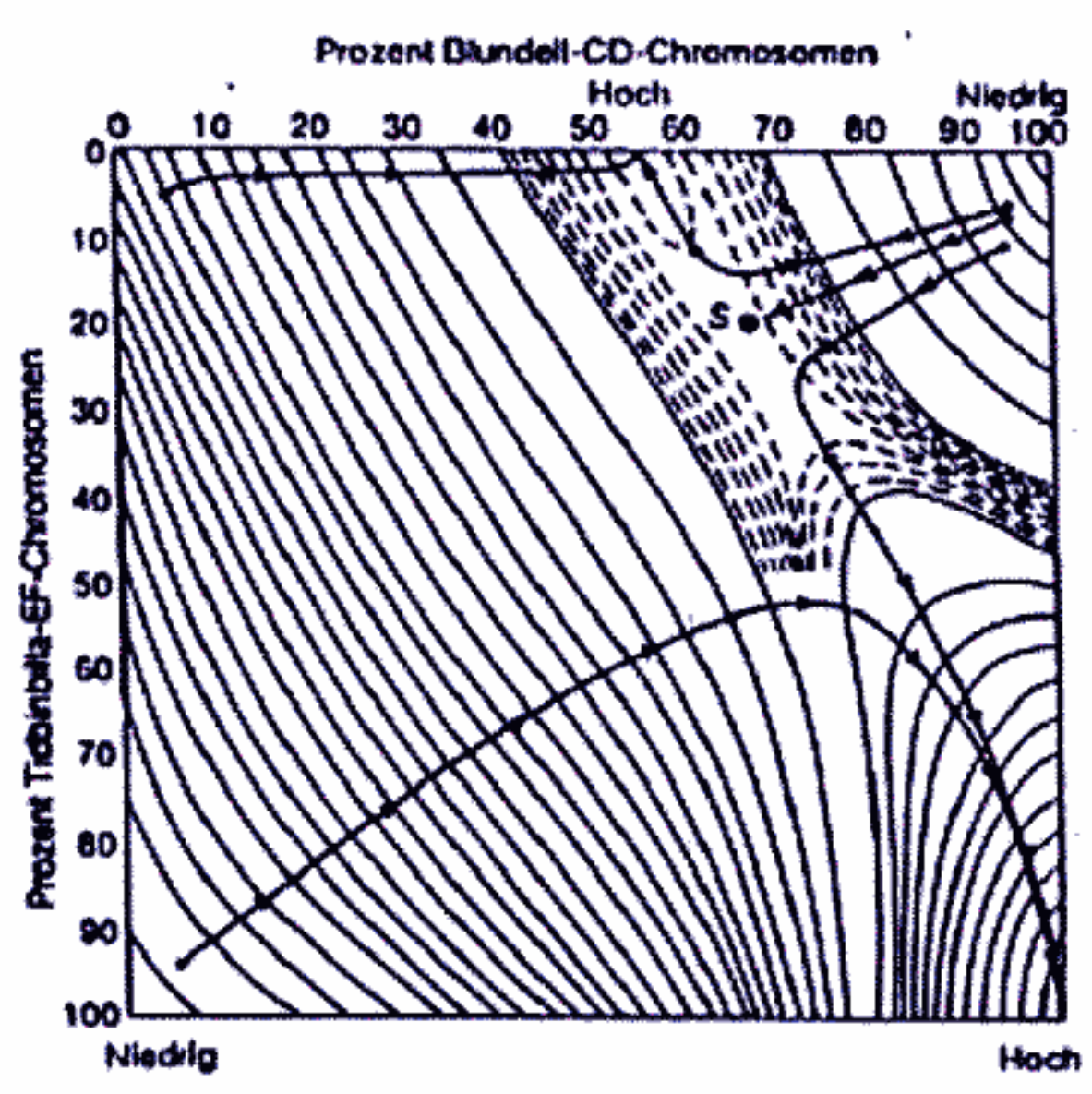
Lit.: Mayr, 1988
Futuyama, 1990

Hawaiische Kleidervögel



Diversifizierung von Arten,
die von gemeinsamen Vorfahren abstammen,
in verschiedene ökologische Nischen

Adaptive Landschaft (Wright, ~1932)



Fitness in Abh. von biolog. Eigensch.

S Sattelpunkt

Abbildung 1:
 Adaptive Landschaft einer Population der australischen Heuschrecke *Moraba scurra*, die für die beiden Chromosomen *EF* und *CD* polymorph ist. Ausgehend von den Genotypfrequenzen im Freiland wurde für jeden Genotyp die Lebensfähigkeit berechnet. Aus diesen Werten wurde wiederum die theoretische Fitness ϕ der Population für alle möglichen chromosomalen Zusammensetzungen berechnet. Kombinationen mit gleichem ϕ sind durch Höhenlinien miteinander verbunden. Die gestrichelten Linien geben feinere Unterscheidungen von ϕ an als die durchgezogenen Linien. Es gibt zwei Gipfel (Hoch) und einen Sattelpunkt (S). Die Trajektorien stellen theoretische Änderungen der genetischen Zusammensetzung dar, denen eine Population ausgehend von fünf Anfangszuständen folgen würde. (Nach Lewontin & White 1960)

Fitness: Überlebens- und Reproduktionsfähigkeit eines Individuums in seiner Umwelt

schematische Darstellung

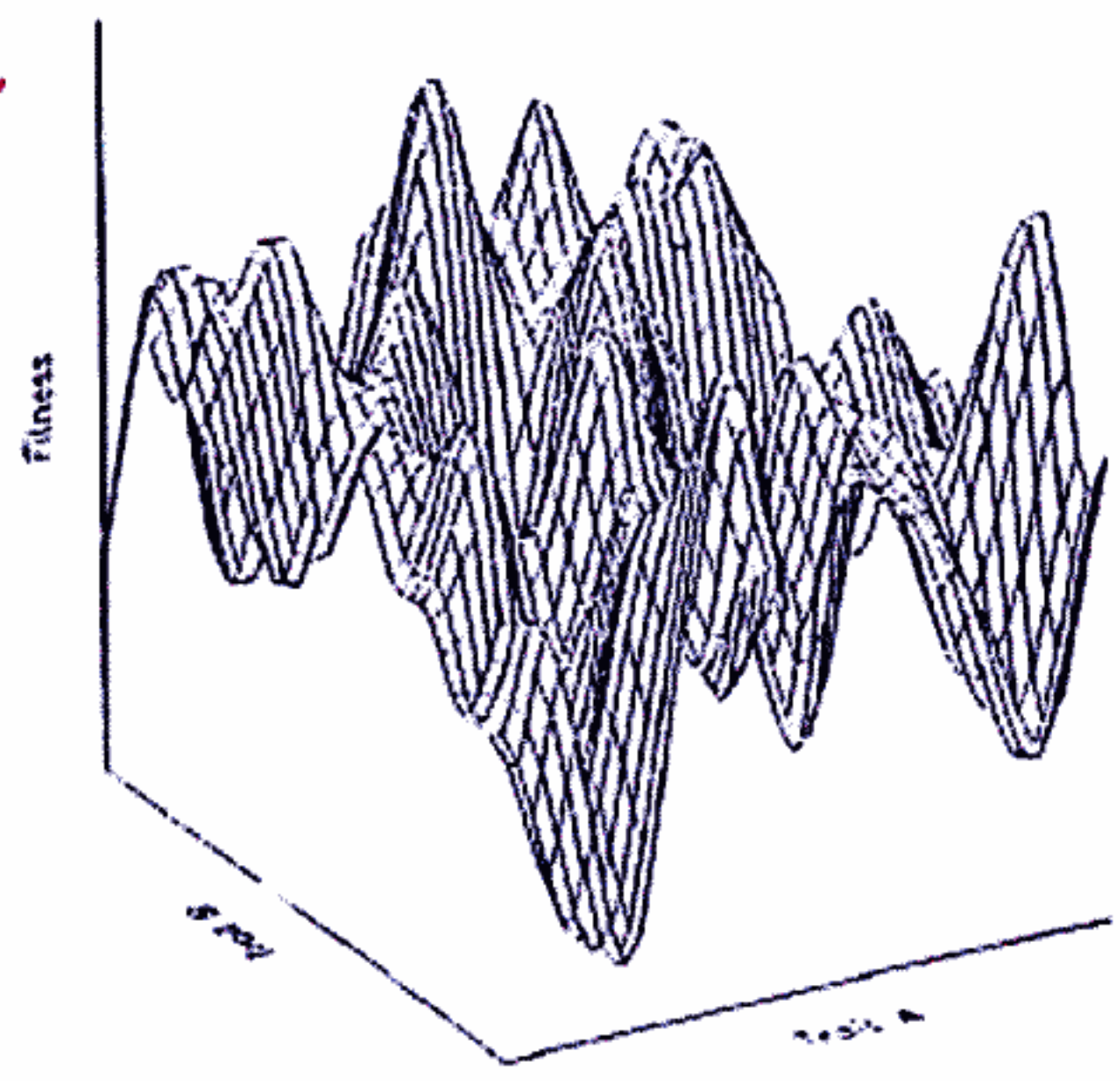
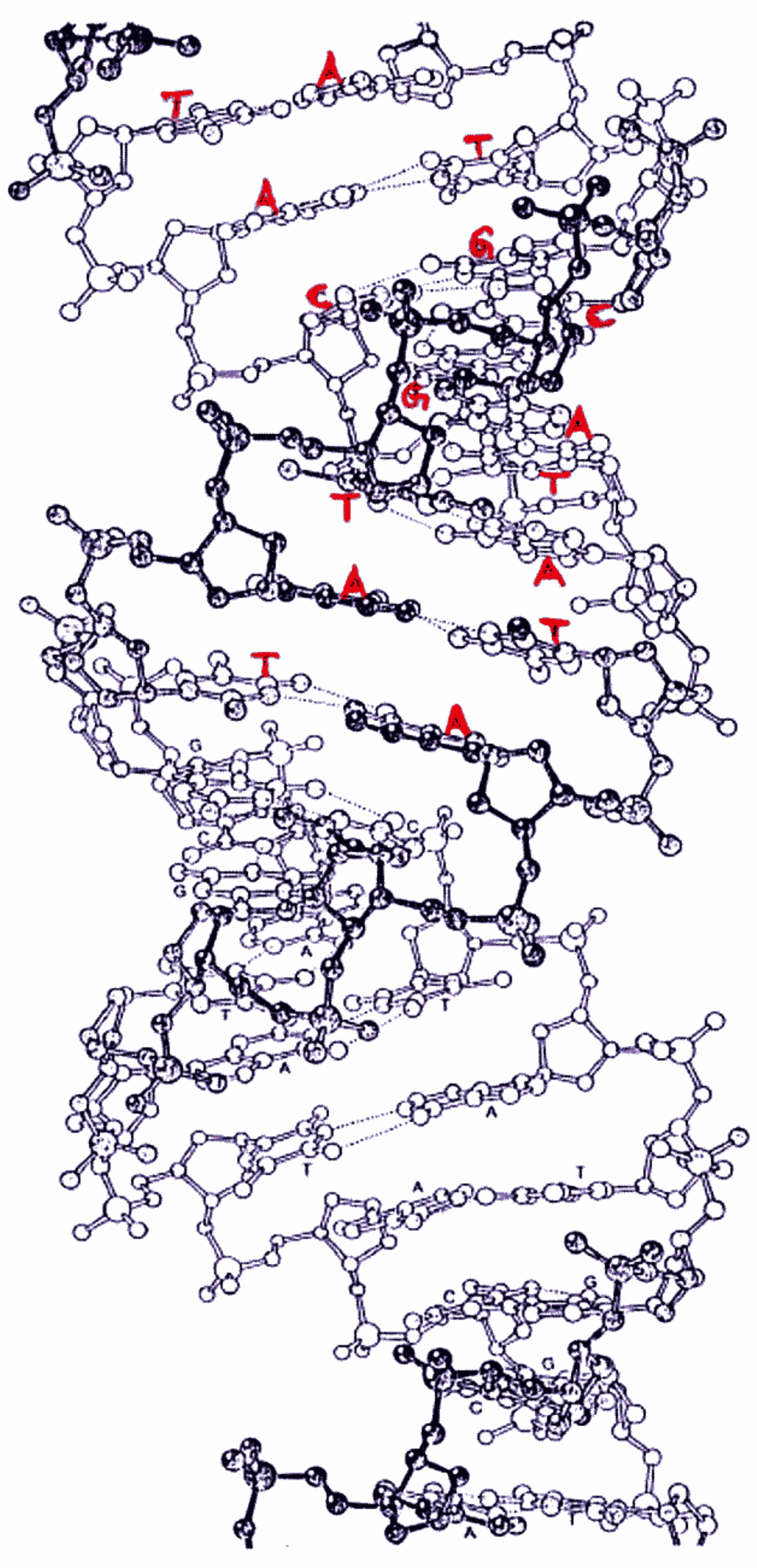


Figure 1.1: Schematic diagram of an adaptive surface.

DNA-Strukturausschnitt



Genetik in Kürze :

in jedem Zellkern befindet sich ein 'Bauplan' des zugehörigen Individuums :

Fadenmolekül DNA Desoxyribonukleinsäure

(spiralförmige Doppelhelix, Watson + Crick;
zwei Einzelstränge aus Zucker + Phosphat,
verbunden über H-Brückenbindungen zwischen

$A \leftrightarrow T$

Adenin, Thymin

$G \leftrightarrow C$

Guanin, Cytosin

(Nukleotidbasen)

DNA ist Charakteristikum (Genotyp) für
Erscheinungsform (Phänotyp) eines Individuums

- Ontogenese Urzelle \rightarrow Lebewesen
- weitere Zellteilungen Bauplan \uparrow Ausprägung
- Umwelt

Mensch: $\sim 3.8 \cdot 10^9$ Nukleotidbasen

(Lungenfisch hat erheblich mehr)

(Bakterien, Hefen am wenigsten)

diploid: 2 Doppelstränge

$4^{3.8 \cdot 10^9} \approx 10^{2.300.000.000}$ mögl. Genotypen

Zentrales Dogma der Molekulargenetik:

vererbt wird der Genotyp, nicht Phänotyp
 d.h. erworbene Eigenschaften eines Individuums
 sind nicht vererbbar (vgl.: Lamarckismus)

? Genotyp - Phänotyp - Abbildung

mehrstufiger Prozeß, Umwelt spielt mit

1. Stufe

ein Basentriplett codiert eine Aminosäure

$$4^3 = 64$$

20

(redundanter genetischer Code, fast universell)
 (heute)

61 Zuordnungen + 3 Stop-Zeichen

2. Stufe

(30...)100 — 1000(... 30000) Aminosäuren Gen
 codieren ein Protein

3.

Zellen

⋮

Organe

n. Stufe

Individuum

Genom

Polygenie: kombinierter Einfluß mehrerer Gene
 auf eine phänotypische Eigenschaft

Pleiotropie: ein Gen beeinflusst mehrere phänotyp.
 Eigenschaften

Der genetische Code

Tabelle I

Der genetische Code. Die 64 Nukleotid-Triplets der mRNA und die dazugehörigen Aminosäuren

		ZWEITES NUKLEOTID				
		U	C	A	G	
ERSTES NUKLEOTID	U	UUU } Phe UUC } UUA } Leu UUG }	UCU } UCC } Ser UCA } UCG }	UAU } Tyr UAC } UAA Kettenende UAG Kettenende	UGU } Cys UGC } UGA Kettenende UGG Trp	U C A G
	C	CUU } CUC } Leu CUA } CUG }	CCU } CCC } Pro CCA } CCG }	CAU } His CAC } CAA } Gln CAG }	CGU } CGC } Arg CGA } CGG }	U C A G
	A	AUU } Ile AUC } AUA } AUG Met	ACU } ACC } Thr ACA } ACG }	AAU } Asn AAC } AAA } Lys AAG }	AGU } Ser AGC } AGA } Arg AGG }	U C A G
	G	GUU } GUC } Val GUA } GUG }	GCU } GCC } Ala GCA } GCG }	GAU } Asp GAC } GAA } Glu GAG }	GGU } GGC } Gly GGA } GGG }	U C A G

Die in der Tabelle abgekürzten Aminosäuren heißen: Ala, Alanin; Arg, Arginin; Asn, Asparagin; Asp, Asparaginsäure; Cys, Cystin; Gly, Glycin; Glu, Glutaminsäure; Gln, Glutamin; His, Histidin; Ile, Isoleucin; Leu, Leucin; Met, Methionin; Phe, Phenylalanin; Pro, Prolin; Ser, Serin; Thr, Threonin; Trp, Tryptophan; Tyr, Tyrosin; Val, Valin.

AUG : auch Start

Protein-Biosynthese

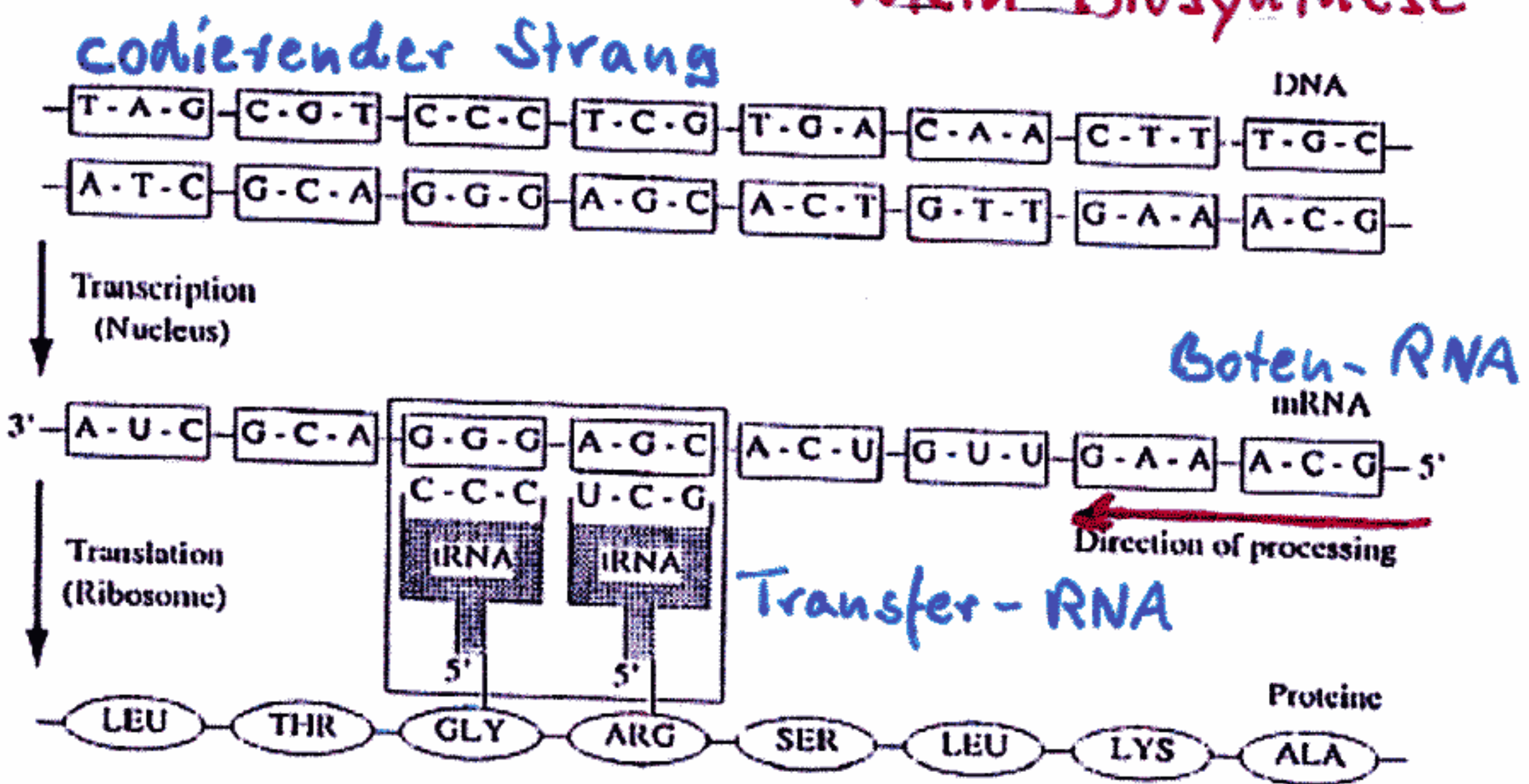


Figure 1.2: Simplified scheme of protein biosynthesis in living cells.

Proteine

U (Uracil) statt T (Thymin) in mRNA

Polyploidie : DNS (Uromeromen-Satz) mehrfach vorhanden

haploid einfach

diploid doppelt (z.B. Mensch)

polyloid 4, 8, 16 ... (z.B. Pflanzen nahe Vulkanen, Gletschern)

nur jeweils eines der Allele (homologe Gene) wird ausgeprägt [Dominanz, Rezessivität]
[homozygot, heterozygot]

Meiose : durch Zellteilung entstehen haploide Keimzellen

Mitose : durch Kernverdopplung und amschl. Zellteilung entstehen wieder diploide Zellen

Vererbung (meiotisch) : Verschmelzung zweier Keimzellen + amschl. 2 Teilungen ; dabei

Umordnungsprozesse : Crossingover = Segmentaustausch zwischen Chromosomenpaaren

fehler : Kopierfehler, Übertragungsfehler

Umwelteinflüsse : Höhenstrahlung, Radioaktivität
...

gegenmaßnahme : Reparaturmechanismen
enzyme

unvollkommene Reparatur! sogar Mutatorgene

↓ Genmutationen

(etwas Unordnung / Chaos muß wohl sein)

Gen - Mutationen

- somatisch nur Körperzellen betreffend
- generativ Keimzellen betreffend
(also vererbbar)

Chromosomen - Mutationen (meist letal)

- Deletionen, Defizienzen
- Duplikationen
- Translokationen
- Inversionen

Genom - Mutationen

- Hyperploidie Zunahme d. Chromosomenzahl
- Hypoploidie Abnahme
- Euploidie komplette Chromosomensätze
zusätzlich

Exon, Introns : noch weitgehend unverständlich:
mehr oder weniger große Teile des Genoms dienen nicht
zur Ausprägung eines Phänotyps

Mensch : 23 Chromosomen
ca. 100 000 Gene

Hierarchie der (genetischen) Information

Quarks

⋮

Atome

⋮

Moleküle

Nukleotid base

Einzel-symbol

Codon

Gen

Entsch. variable

Skripton

Replikon

Chromosom

Genom

Satz von Entsch. variablen

GA: setzen auf Codierung mit Alphabet
minimaler Kardinalität

→ Binärstring - Repräsentation
jedes Individuums

EP: setzen auf Artenentwicklung
→ jedes 'Individuum' ist eine Spezies
kein Crossover / keine Rekombination

ES: setzen auf Individuen mit problemabh.

Codierung der Variablen: reell

ganzzahlig, diskret
...

ahmen Genotyp/Phänotyp Abbildung mittels
korrelierter Variabilität nach

allgemeiner EA

$t := 0$

initialisiere $P(0) := \{a_1(0), \dots, a_\mu(0)\} \in \mathbb{I}^\mu$

bewerte $P(0) : \{f(a_1(0)), \dots, f(a_\mu(0))\}$

while ($L(P(t)) \neq \text{true}$) do

rekombiniere : $P'(t) := r_{\Theta_r}(P(t))$

mutiere : $P''(t) := m_{\Theta_m}(P'(t))$

bewerte $P''(t) : \{f(a_1''(t)), \dots, f(a_\lambda''(t))\}$

selektiere : $P(t+1) := s_{\Theta_s}(P''(t) \cup Q)$

$t := t + 1$

od

μ Zahl der Eltern

λ Zahl der Nachkommen

L Abschaltkriterium

r, m, s Operatoren mit Parametern

$\Theta_r, \Theta_m, \Theta_s$

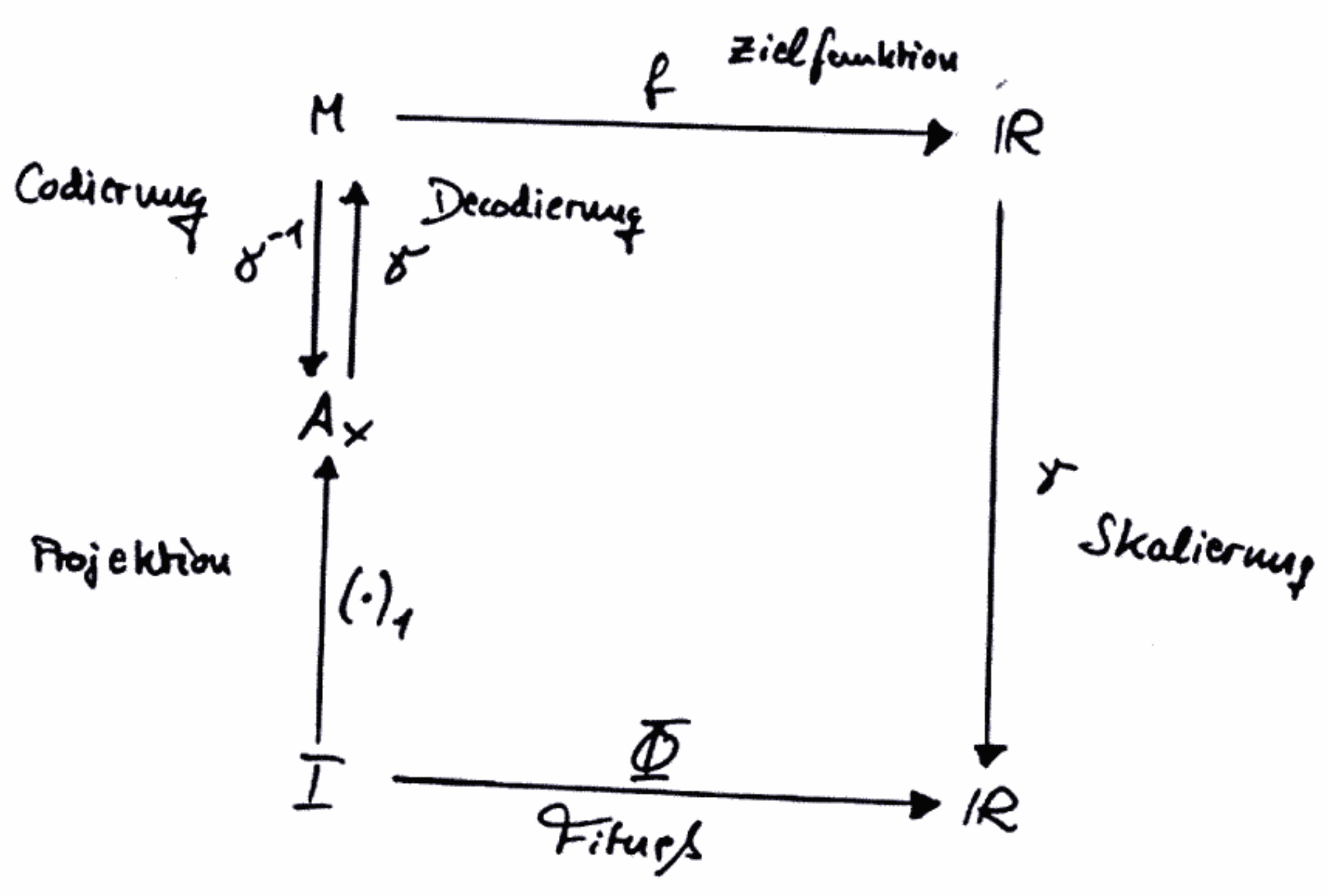
$Q = \{0\}$

$Q = \text{bisher bestes Find.}$

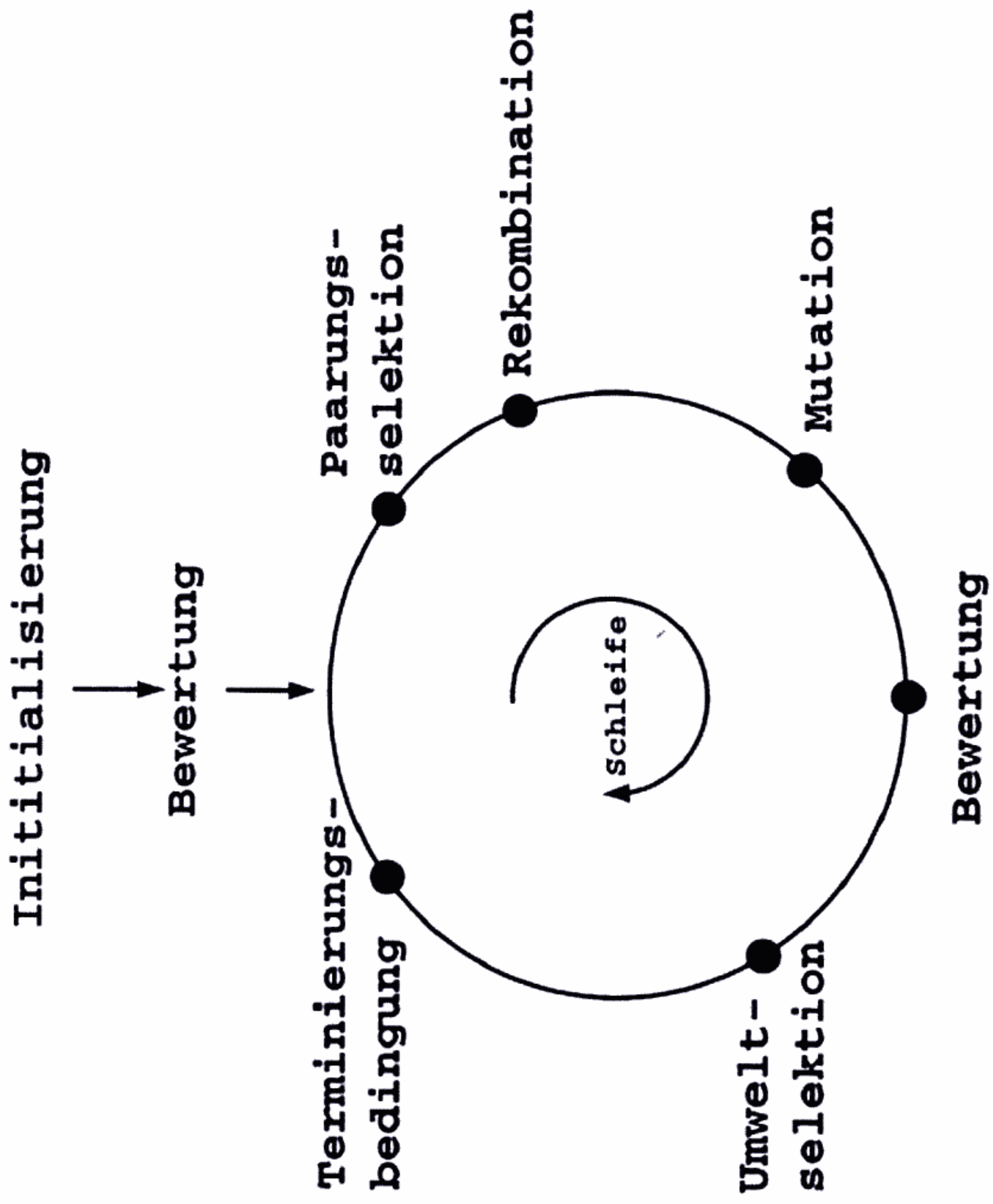
Elitist-Strategie

Evolutionärer Basisalgorithmus

$t := 0;$
 initialisiere $P(0) := \{\vec{a}_1(0), \dots, \vec{a}_\mu(0)\} \in I^\mu;$
 bewerte $P(0) := \{\Phi(\vec{a}_1(0)), \dots, \Phi(\vec{a}_\mu(0))\};$
while $(\text{cond}(P(t)) \neq \text{true})$ do
 $P'(t) := \text{rekombiniere}(P(t));$
 $P''(t) := \text{mutiere}(P'(t));$
 bewerte $P''(t) := \{\Phi(\vec{a}_1''(t)), \dots, \Phi(\vec{a}_\mu''(t))\};$
 $P(t+1) := \text{selektiere}(P''(t) \cup Q);$
 $t := t+1;$
od



Allgemeine EA Iterations-Schleife



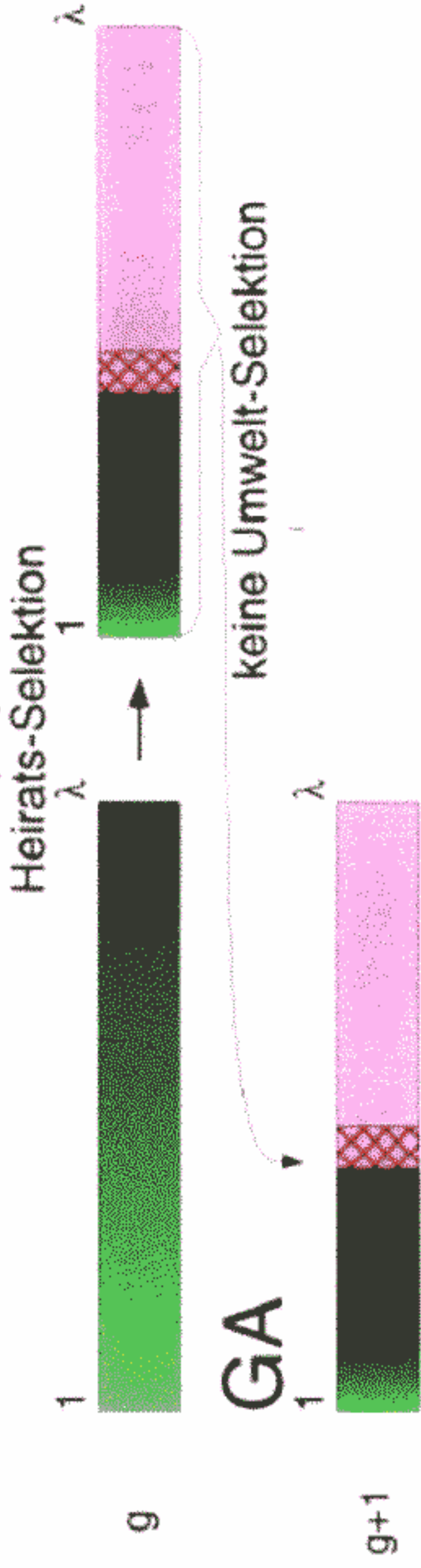
Generationswechsel

Eltern Eltern Eltern Eltern

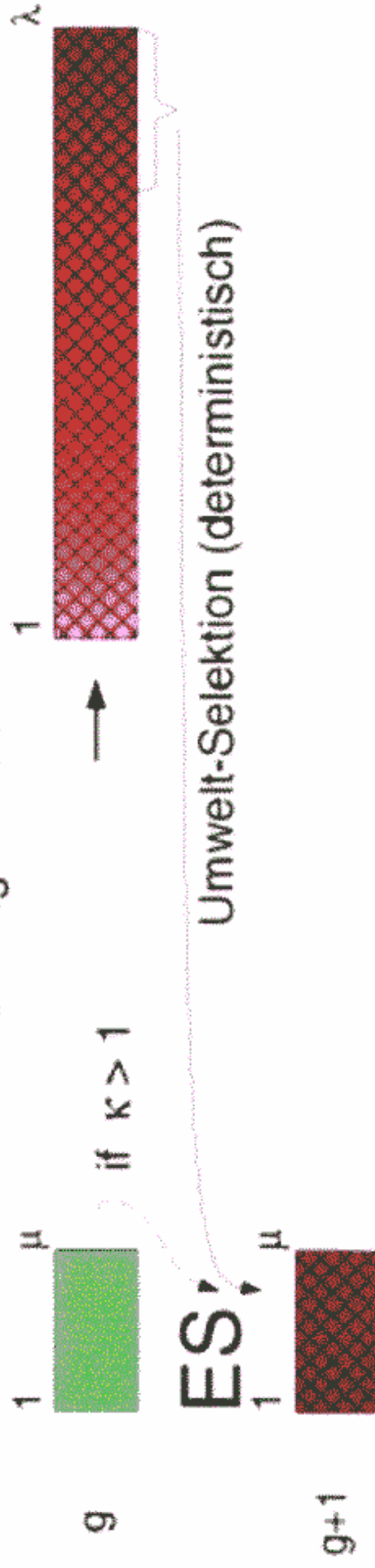
Fitneß-proportionale
Heirats-Selektion

Nachkommen

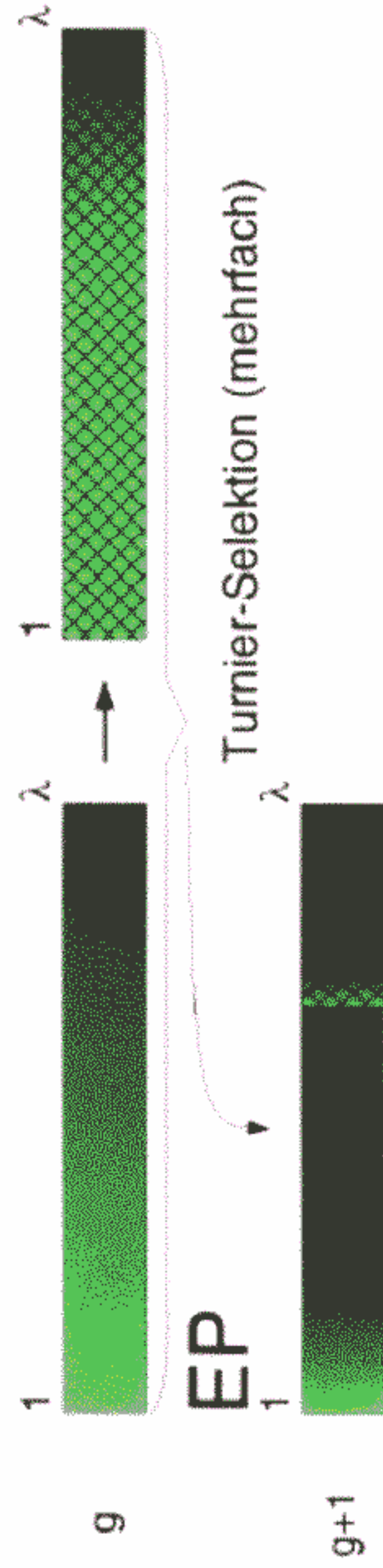
Generation



rein zufällige Heirat



keine Rekombination



Genetische Algorithmen (GA)

anfangs: Modell für Adaptationsprozesse in natürl. u. künstl. Systemen

später: Tool für Pseudo-Boolesche und kombinatorische Optimierung

wichtige Eigenschaften:

- evolvierende Einheit ist das Genom; Zufallsstart in gesamtem Suchgebiet
- Binär-String Repräsentation der Entscheidungsvariablen
- Paarungsselektion: abh. von Fitness; kein Geburtenüberschuß ($\lambda = \mu$)
- Crossover zweier Bitketten und (seltene) Mutation (bit flips)
- keine Umweltselektion

Genetischer Algorithmus (kanonisch)

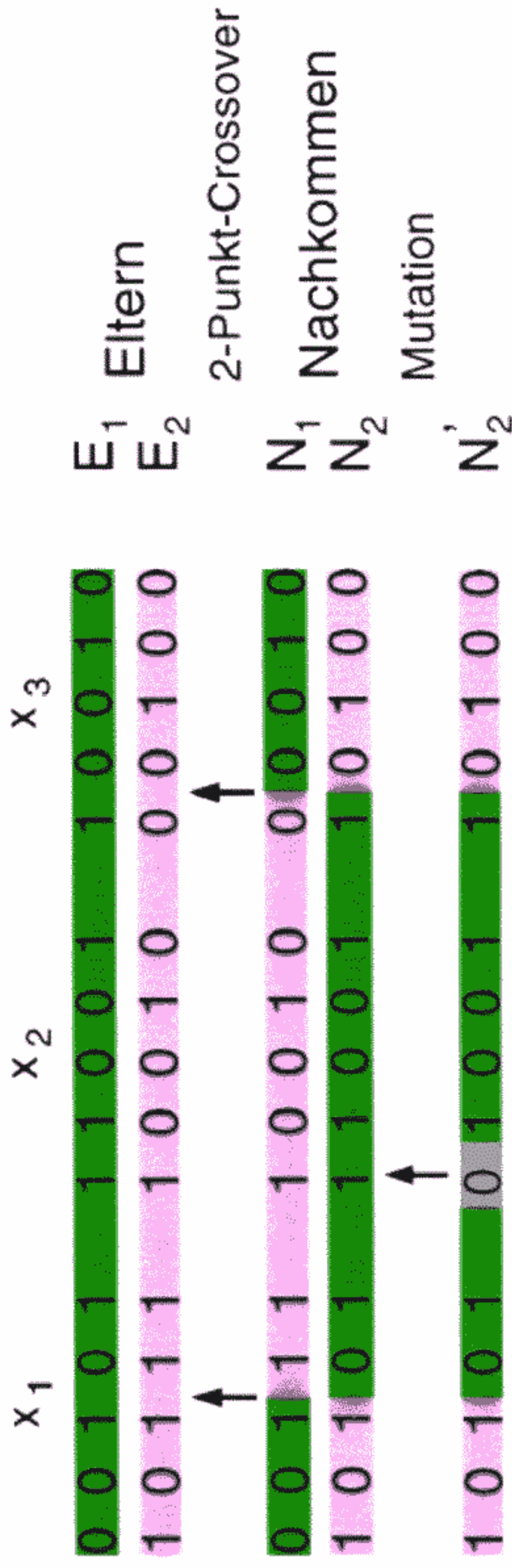
Repräsentation der Individuen : Bitkette

Mutation : Bitinversion, selten ($p_m = 10^{-3}$ fix), Hintergrundoperator

Rekombination : Crossover, häufig ($p_c = 0.6$), Hauptoperator

Reproduktion : $\lambda = \mu$, kein Geburtenüberschuß

Selektion : Nachkommenzahl prop. zu relativer Fitness (mating selection)



Kanonischer GA:

Individuum = Bitstring $\{ \underbrace{1001101011100100 \dots}_{\text{Problem-Variable } x_1} \}$

Problem-Variable x_1

d.h. Binär-Kodierung obligatorisch



für alle Variablen muß ein Wertebereich und eine Darstellungsgenauigkeit gewählt werden, bzw. eine Anzahl l von Bits für die Darstellung

Bsp.

$$u_i \leq x_i \leq v_i \quad \forall i = 1(1)n$$

$$x_i = u_i + \frac{\varepsilon_i}{2^l - 1} \sum_{j=1}^l a_{ij} 2^{l-j}$$

wobei: $\{a_{ij}\}$ Bitstring der Länge l für x_i

$$l=4 \quad u_i = 0 \quad v_i = 15 \quad \varepsilon_i = 1 = \frac{15-0}{2^4-1}$$

$\{a_{ij}\} =$	0000	$x_i =$	0
	0001		1
	0010		2
	0011		3
	0100		4
	0101		5
	0110		6
	0111		7
	1000		8
	1001		9
	1010		10
	1011		11
	1100		12
	1101		13
	1110		14
	1111		15

Table 5.2: Transition probabilities for mutations within a GA

Genotype		x_i new			
		00	01	10	11
x_i old	Phenotype	0	1	2	3
	00	q^2	pq	pq	p^2
	01	pq	q^2	p^2	pq
	10	pq	p^2	q^2	pq
	11	p^2	pq	pq	q^2

keine Änderung

GA - Übergangswahrscheinlichkeiten für Mutation mit $l=2$ $x \in \{0,1,2,3\}$

z.B.

$$p = \mu_m = 0.001$$

$$q = (1 - \mu_m) = 0.999$$

$$q^2 = 0.998001$$

$$pq = 0.000999$$

$$p^2 = 0.000001$$

Übergänge $1 \leftrightarrow 2$ am unwahrscheinlichsten obwohl phänotypisch benachbart

Für sukzessive Verbesserungen (\neq großes Los ziehen) ist entscheidend: Veränderung Δx_i der phänotyp. Variablen von Generation zu Generation

(Volumen- versus pfad-orientierte Suche)

Gray - Code

Bsp. 3 bits für ganzzahlige Variable $x \in \{0, \dots, 7\}$

x	Standard-Binärcode	Gray-Code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

$\{a_{ij}\}$ $\{b_{ij}\}$

Umwandlung Standard \leftrightarrow Gray

$$z: \quad b_{ij} = \begin{cases} a_{ij} & \text{für } j=1 \\ a_{ij-1} \oplus a_{ij} & \text{für } j>1 \end{cases}$$

$$z^{-1}: \quad a_{ij} = \bigoplus_{k=1}^j b_{ik}$$

\oplus : modulo 2 - Addition

$p(\Delta x)$
↑

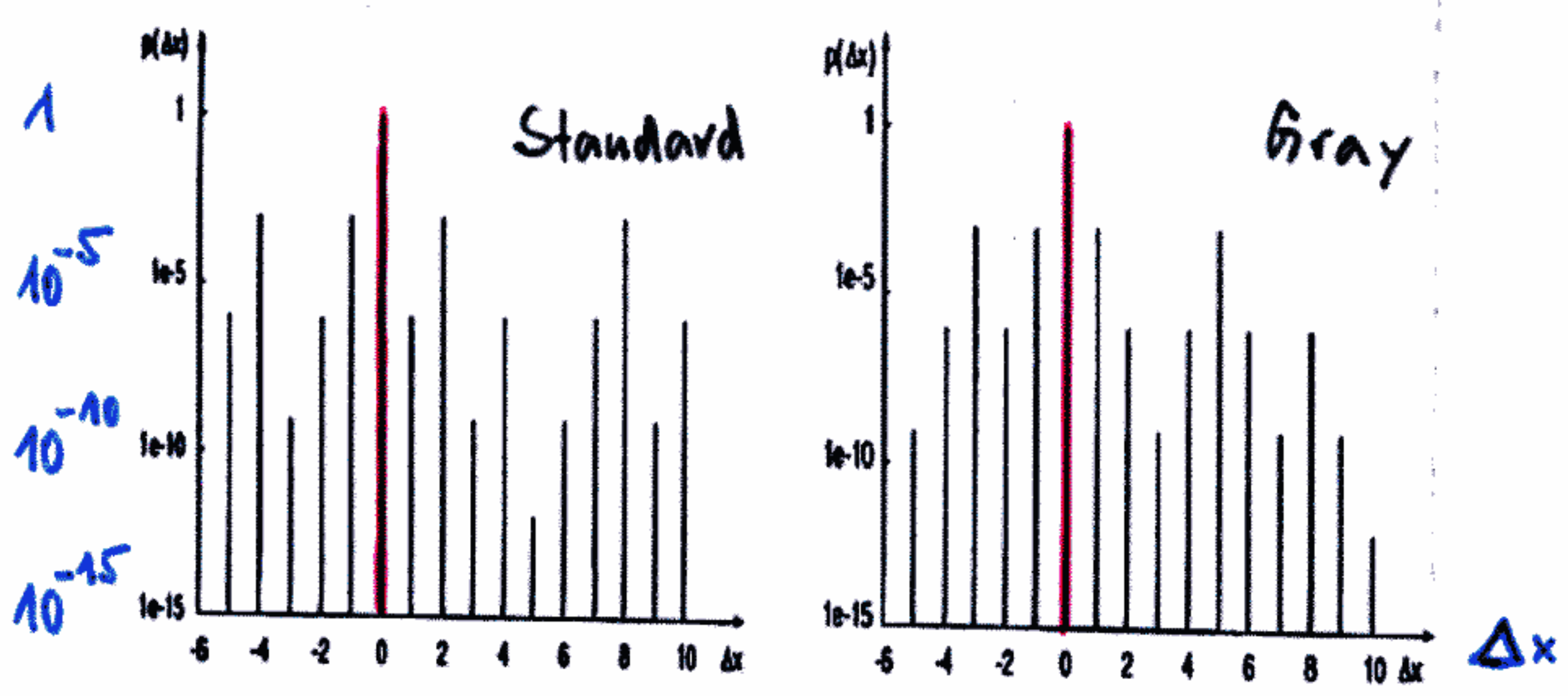


Figure 5.10: Probability distributions for GA mutations / left: normal binary code; right: Gray code

GA - Mutationen am Beispiel

$x^{(0)} = 5 ; l = 4 ; p_m = 0.001$

{0101} in Standard-Binärdarstellung

$\Delta x = x^{(1)} - x^{(0)} ; x^{(1)} = \text{mutiertes } x^{(0)}$

$p(\Delta x)$ nicht symmetrisch
nicht unimodal

Erwartungswert $\neq 0$

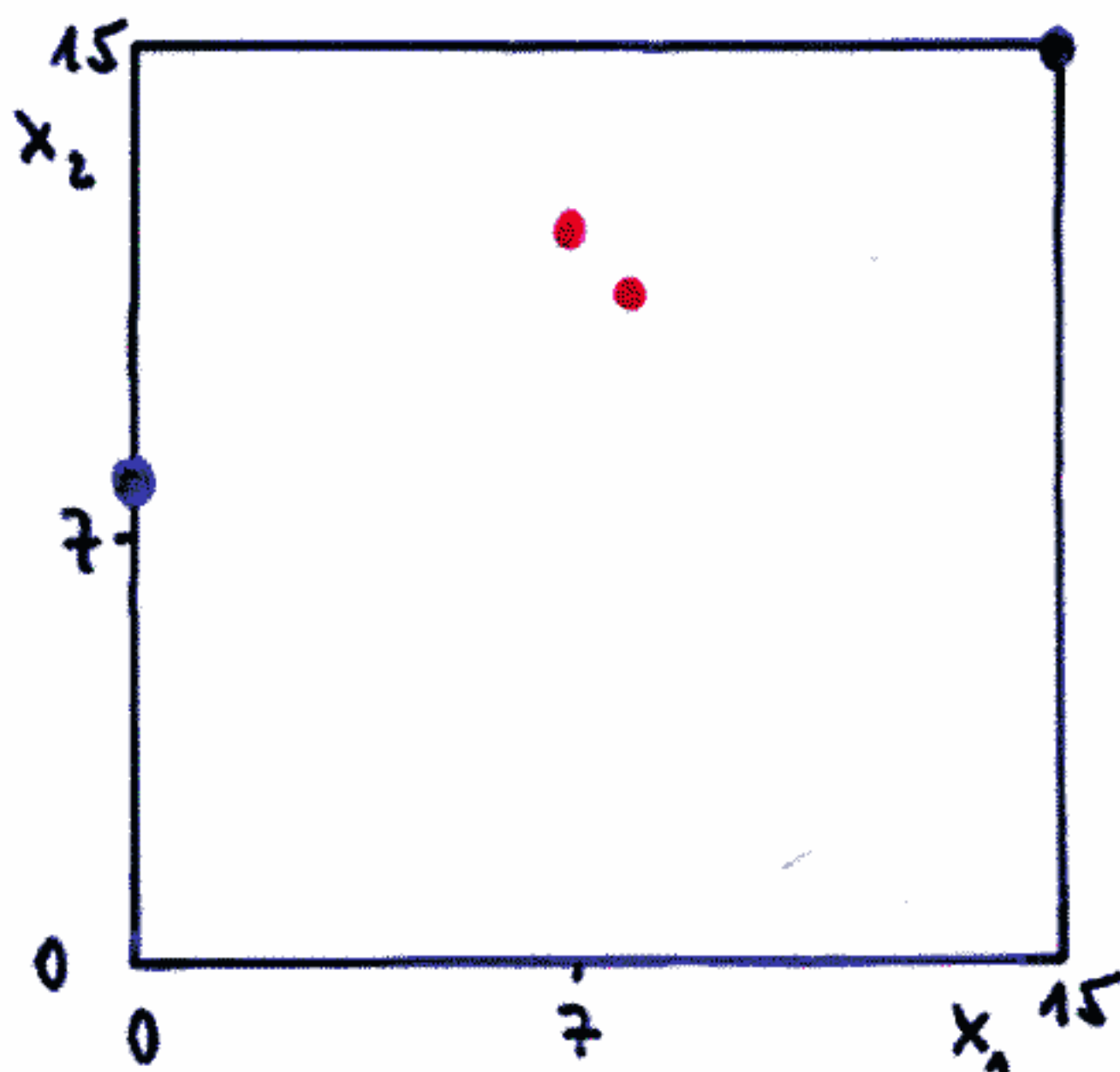
(positionenabhängig)

Table 5.1: Two point crossover within a GA and its effect on the phenotypes

	x_1	x_2	x_1	x_2
Parent 1	0111	1100	7	12
Parent 2	1000	1011	8	11
Offspring 1	0000	1000	0	8
Offspring 2	1111	1111	15	15

Bruchstellen

GA - Zweipunkt - Crossover am Beispiel



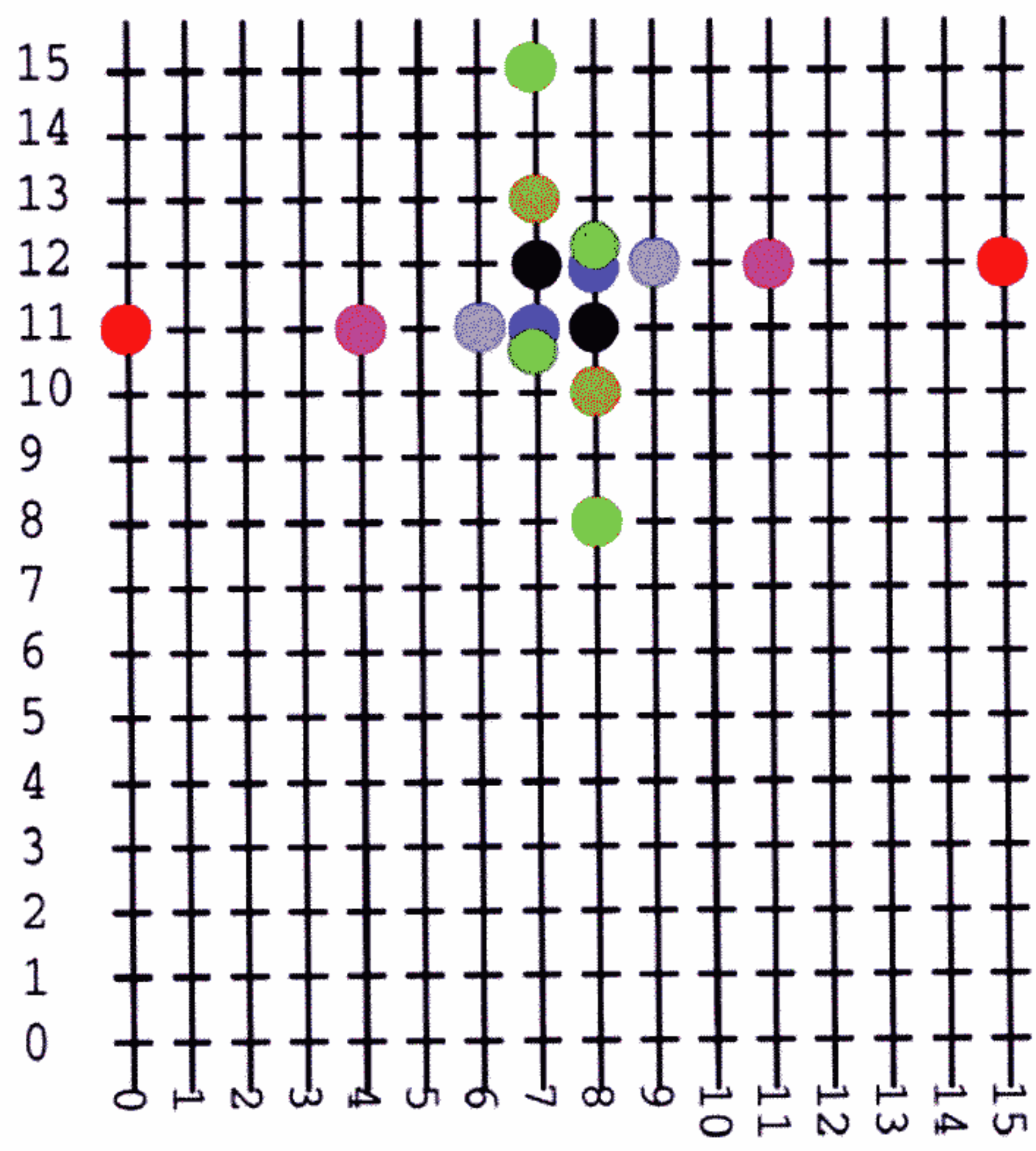
- Eltern
- Nachkommen

Rekombination kann zu 'wilden' Neutypen führen

aber: viele verschiedene Vorschläge

z.B. Uniform Crossover

gene pool recombination



● Eltern

Genotyp

**Zwischen-
(phäno)typ**

0111 1100
1000 1011

7 12
8 11

0000 1011
1111 1100
0100 1011
1011 1100
0110 1011
1001 1100
0111 1011
1000 1100

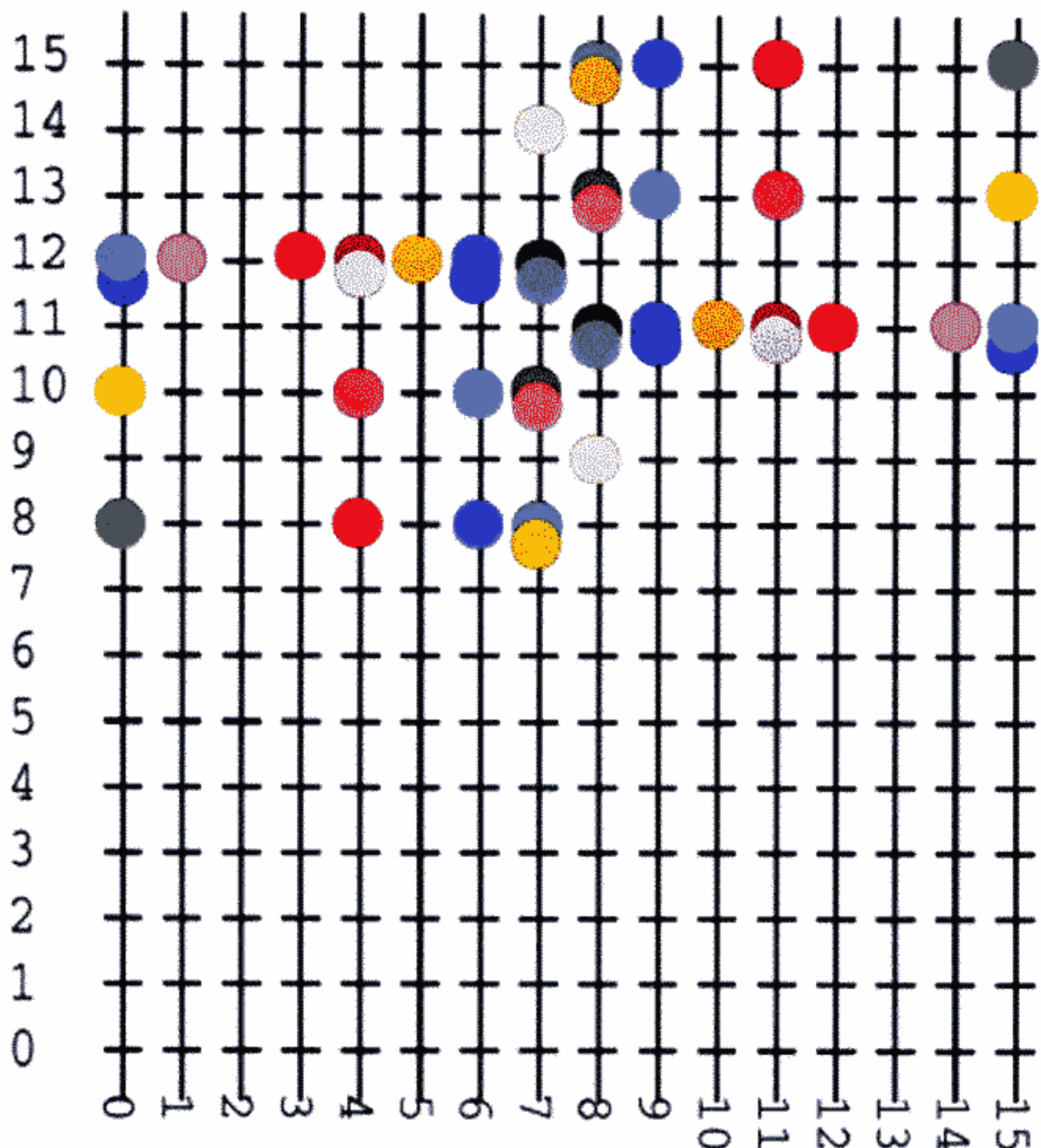
0 11
15 12
4 11
11 12
6 11
9 12
7 11
8 12

11 ●
12 ●
11 ●
12 ●
11 ●
12 ●
11 ●
12 ●

0111 1111
1000 1000
0111 1101
1000 1010

7 15
8 8
7 13
8 10

●
●
●
●

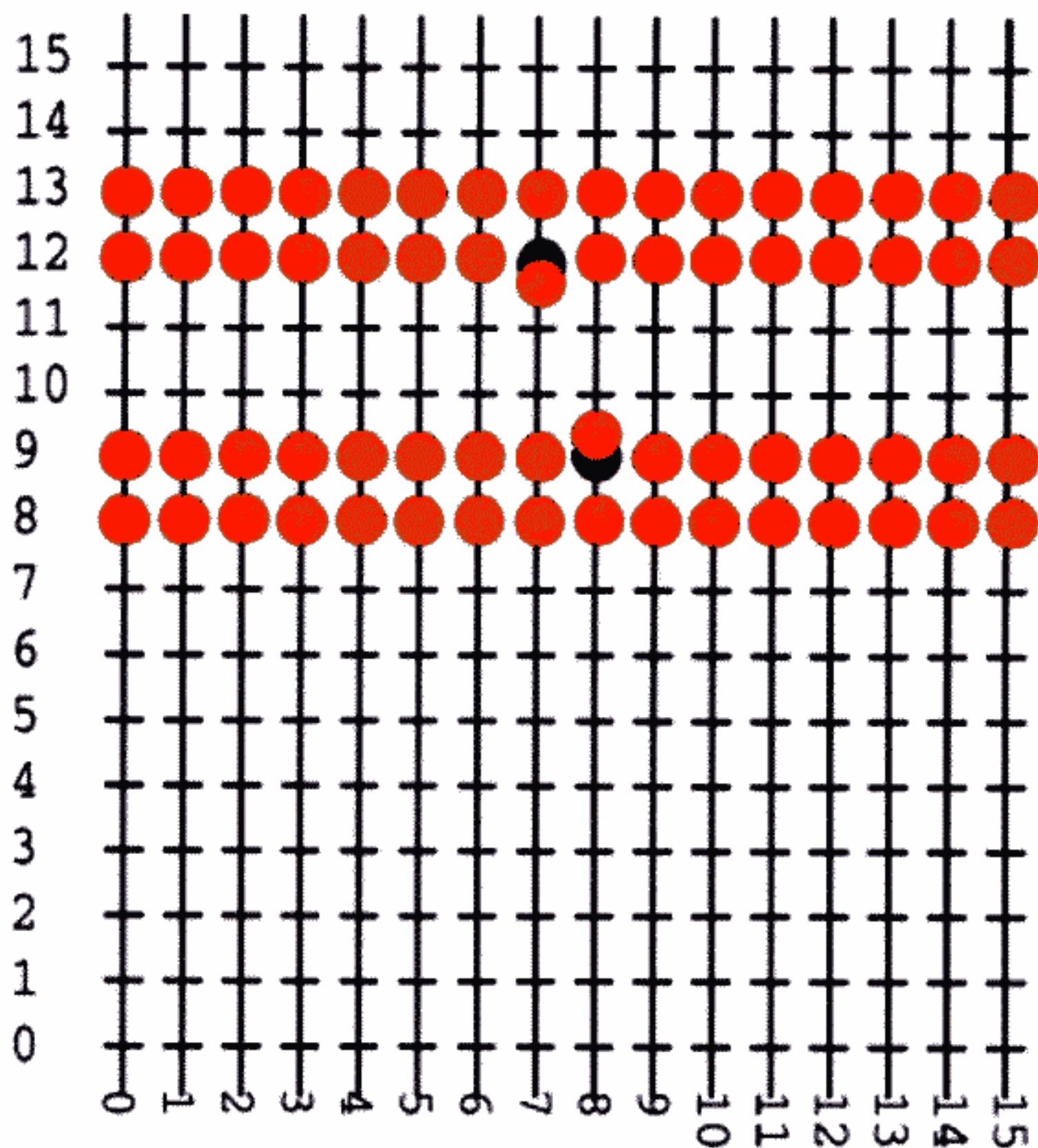


● Eltern

Genotyp **Zwischen-
(phäno)typ**

0111 1100 7 12
1000 1011 8 11

0011 1100 ●	0000 1000 ●	0100 1100 ●	0110 1100 ●	0111 1000 ●
1100 1011 ●	1111 1111 ●	1011 1011 ●	1001 1011 ●	1000 1111 ●
0001 1100 ●	0000 1010 ●	0100 1000 ●	0110 1000 ●	0111 1010 ●
1110 1011 ●	1111 1101 ●	1011 1111 ●	1001 1111 ●	1000 1111 ●
0000 1100 ●	0101 1100 ●	0100 1010 ●	0110 1010 ●	0111 1010 ●
1111 1011 ●	1010 1011 ●	1011 1101 ●	1001 1101 ●	1000 1101 ●
0000 1100 ●	0100 1100 ●	0110 1100 ●	0111 1100 ●	0111 1110 ●
1111 1011 ●	1011 1011 ●	1001 1011 ●	1000 1011 ●	1000 1001 ●



● Eltern

Genotyp

**Zwischen-
(phäno)typ**

0111 1100
1000 1001

7 12
8 9

1110 1000
1110 1001
1110 1100
1110 1101
1111 1000
1111 1001
1111 1100
1111 1101

1100 1000
1100 1001
1100 1100
1100 1101
1101 1000
1101 1001
1101 1100
1101 1101

0000 1000	0010 1000	0100 1000	0110 1000	1000 1000	1010 1000
0000 1001	0010 1001	0100 1001	0110 1001	1000 1001	1010 1001
0000 1100	0010 1100	0100 1100	0110 1100	1000 1100	1010 1100
0000 1101	0010 1101	0100 1101	0110 1101	1000 1101	1010 1101
0001 1000	0011 1000	0101 1000	0111 1000	1001 1000	1011 1000
0001 1001	0011 1001	0101 1001	0111 1001	1001 1001	1011 1001
0001 1100	0011 1100	0101 1100	0111 1100	1001 1100	1011 1100
0001 1101	0011 1101	0101 1101	0111 1101	1001 1101	1011 1101

Beispiel für Standard - Codierung (6A)

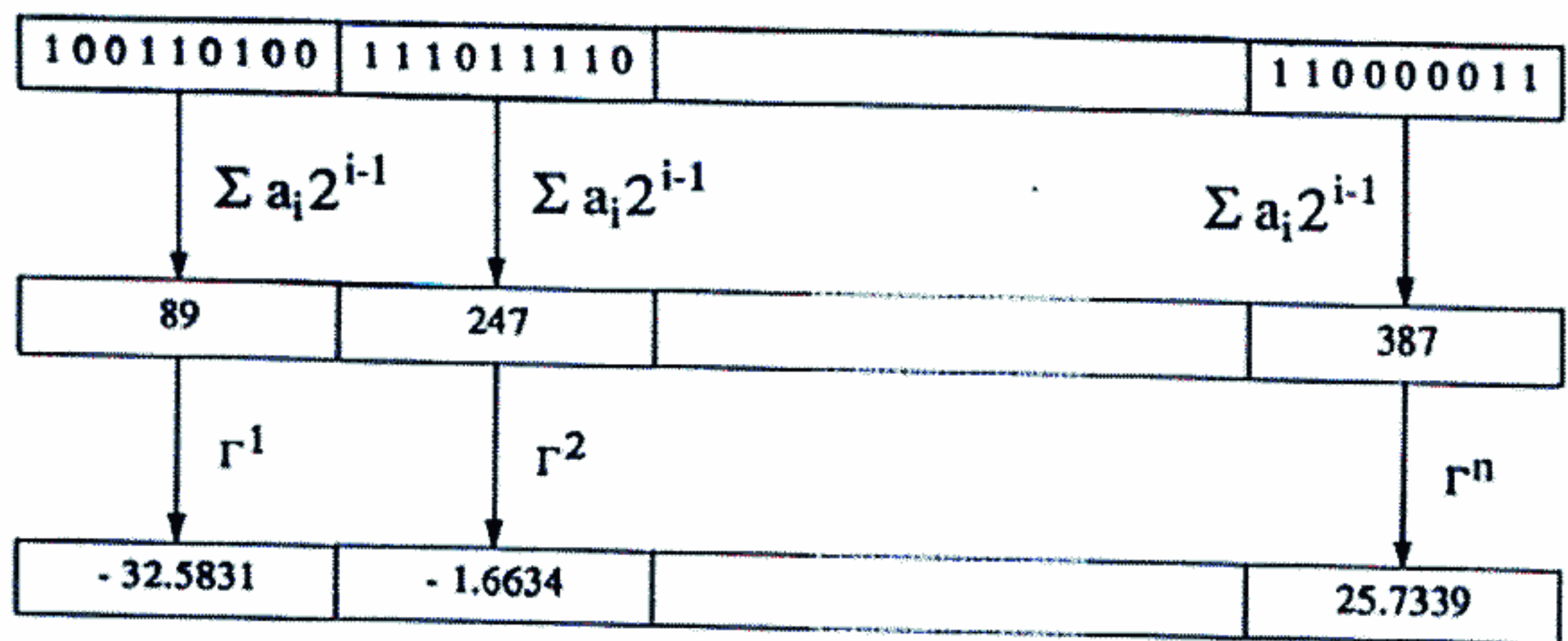
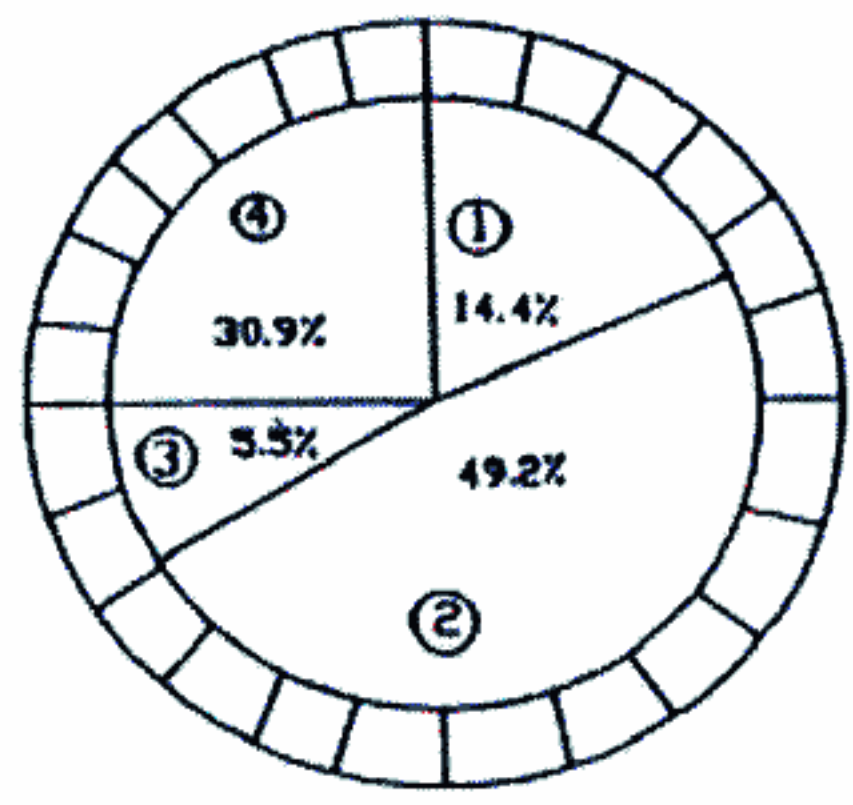


Abbildung 3: Der Dekodierungs-Prozeß.

proportionale Selektion (Roulette-Rad)



κ	Φ_{κ}	$\rho_{s\kappa} (\%)$
1	169	14.4
2	576	49.2
3	64	5.5
4	361	30.9

FIGURE 1.7 Simple reproduction allocates offspring strings using a roulette wheel with slots sized according to fitness. The sample wheel is sized for the problem of Tables 1.1 and 1.2.

Wahrsch., bei Rekombination als Elter 'gezogen' zu werden, abh. von relativer Fitness

$$\rho_{s\kappa} = \frac{\Phi_{\kappa}}{\sum_{j=1}^n \Phi_j}$$

(Bed.: $\Phi > 0$)

meist: Umskalierung notwendig

Notation

Individuum $a \in I$

Individuen-Raum $I = B^l$ für GA

Population $P^{(t)} = \{a_1, \dots, a_\lambda\} \in I^\lambda$

Strategieparameter (fest für einen Lauf)

p_c Rekombinationswahrscheinl.

z Zahl der Crossover Stellen

p_m Mutationswahrscheinlichkeit

Skalierungsparameter für Bewertung (Fitness)

Rekombination $\text{rec}_{\{p_c, z\}} : I^\lambda \times I^\lambda \rightarrow I^\lambda$

Mutation $\text{mut}_{\{p_m\}} : I \rightarrow I$

Selektion $\text{sel}_{\{\dots\}} : I^\lambda \rightarrow I^\lambda$

entsprechend Paarungswahrsch. p_s

$$\text{z.B. } p_s(a_k(t)) = \phi(a_k(t)) / \sum_{i=1}^{\lambda} \phi(a_i(t))$$

$$\phi(a(t)) = \alpha f(x(t)) + \beta > 0$$

t : Iterationszähler

λ : Populationsgröße ($\lambda = \mu$)

Genetischer Algorithmus

 $t := 0$ initialisiere $P^{(0)} := \{a_1(0), \dots, a_\lambda(0)\} \in I^\lambda$ mit $I \in B^l$ bewerte $P^{(0)} := \{\phi(a_1(0)), \dots, \phi(a_\lambda(0))\}$ while $(L(P(t)) \neq \text{true})$ dorekombiniere $a'_k(t) := \text{rec}_{\xi_3}(P(t)) \forall k \in \{1 \dots \lambda\}$ mutiere $a''_k(t) := \text{mut}_{\xi_3}(a'_k(t)) \forall k \in \{1 \dots \lambda\}$ bewerte $P''(t) := \{a''_1(t), \dots, a''_\lambda(t)\} :$
 $\{\phi(a''_1(t)), \dots, \phi(a''_\lambda(t))\}$ selektiere $P^{(t+1)} := \text{sel}_{\xi_3}(P''(t))$

mit Selektionswahrscheinlichkeiten

$$p_s(a''_k(t)) = \frac{\phi(a''_k(t))}{\sum_{i=1}^{\lambda} \phi(a''_i(t))}$$

 $t := t+1$

od

$$\phi(a) = \delta(f(\underbrace{g(a)}_x))$$

$x = \text{decode}(a)$

Kodierung

SRM \rightarrow \rightarrow RMS
 $\angle \cup Q$
 wobei
 $Q = P(t)$
 oder
 $Q = I_{\text{best}}(t)$

Selektionsvarianten bei GAs

proportionale Selektion

nur hierfür 'gilt' das Schema-Theorem

probabilistische 'Auslese' (mating selection)

rangbasierte Selektion

$$p_s = f(\text{Rang})$$

tournament selection

wähle λ mal $g < \lambda$ Konkurrenten;

bester 'überlebt' (ggf. mehrfach)

truncation selection

à la ES

Reproduktion

stets $\lambda = \mu$ (non-Darwinian!)

heute: Hunderte von speziellen GAs

cGA canonical GA (Holland)

sGA simple GA (Goldberg)

Michigan approach
Pittsburgh approach

⋮
PGA parallel GA

⋮
mGA messy GA (unordentlich; Goldberg)

⋮
BGA breeder GA (Mühlenbein)

truncation selection } à la ES
gene pool recombination }

Buch von Zbigniew Michalewicz:

Data Structures + Genetic Algorithms
= Evolution Programs

! Problem - Modellierung; Variablen - Codierung

! angepasste Operatoren für Mutation } Variation
Rekombination }

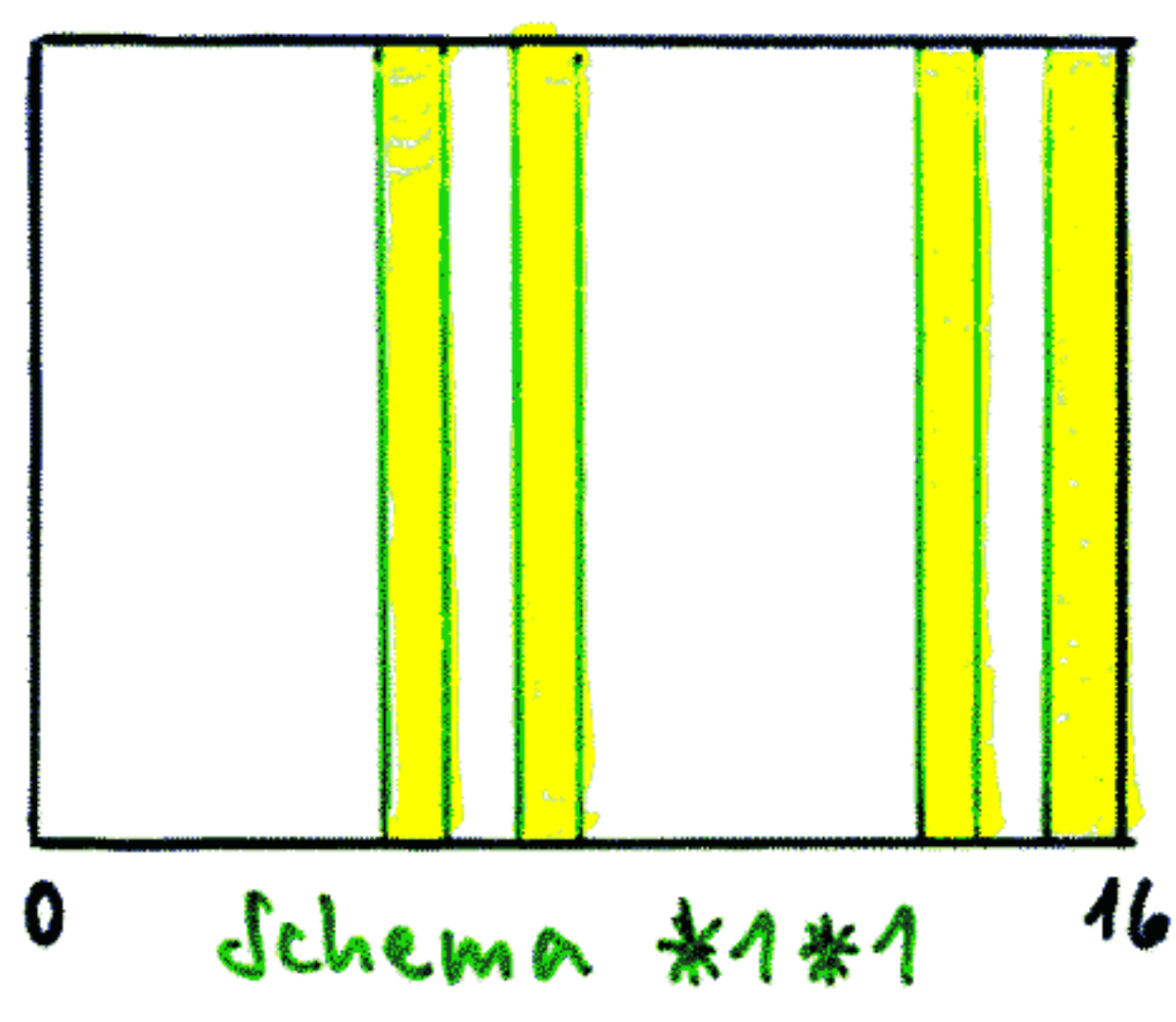
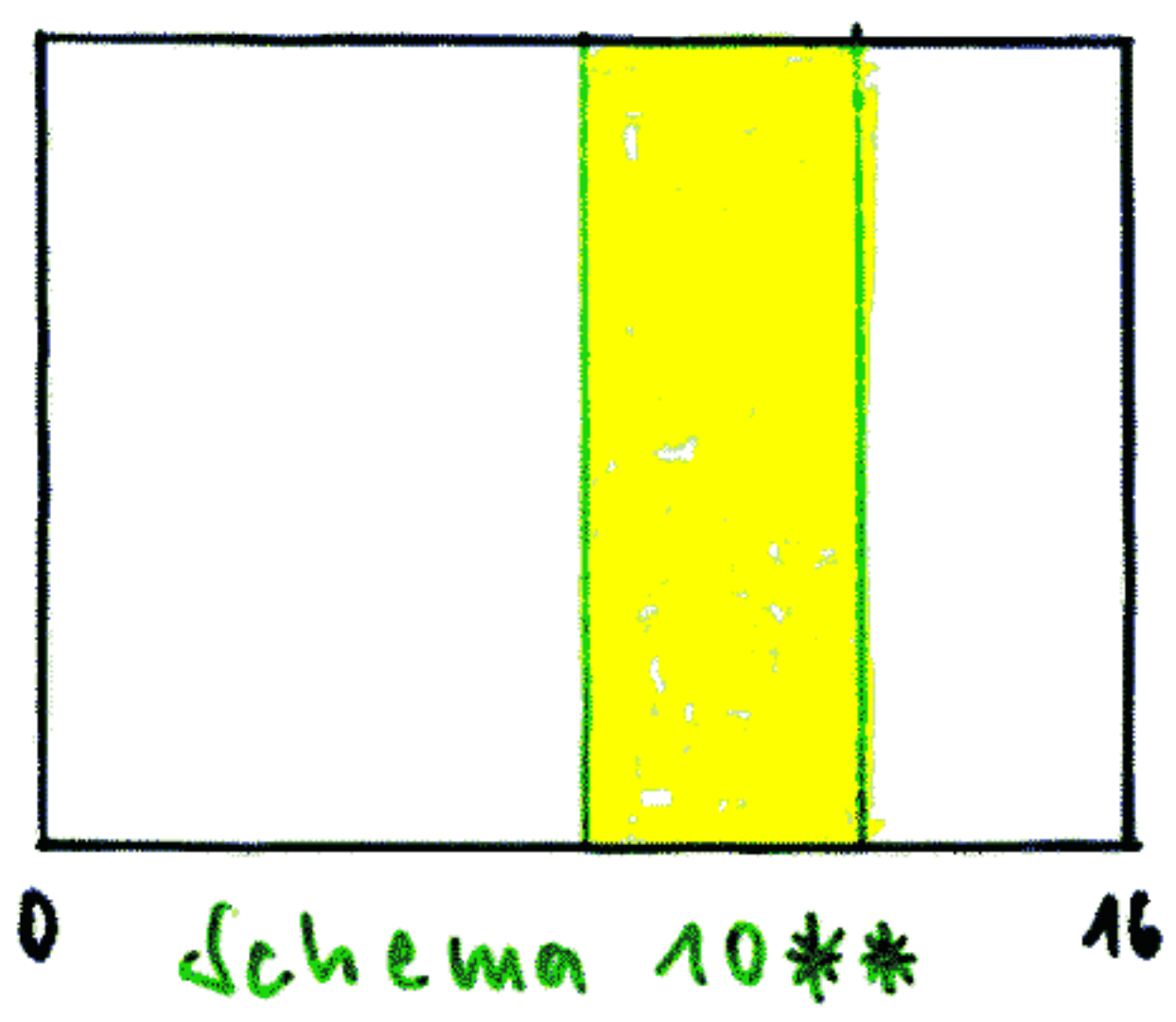
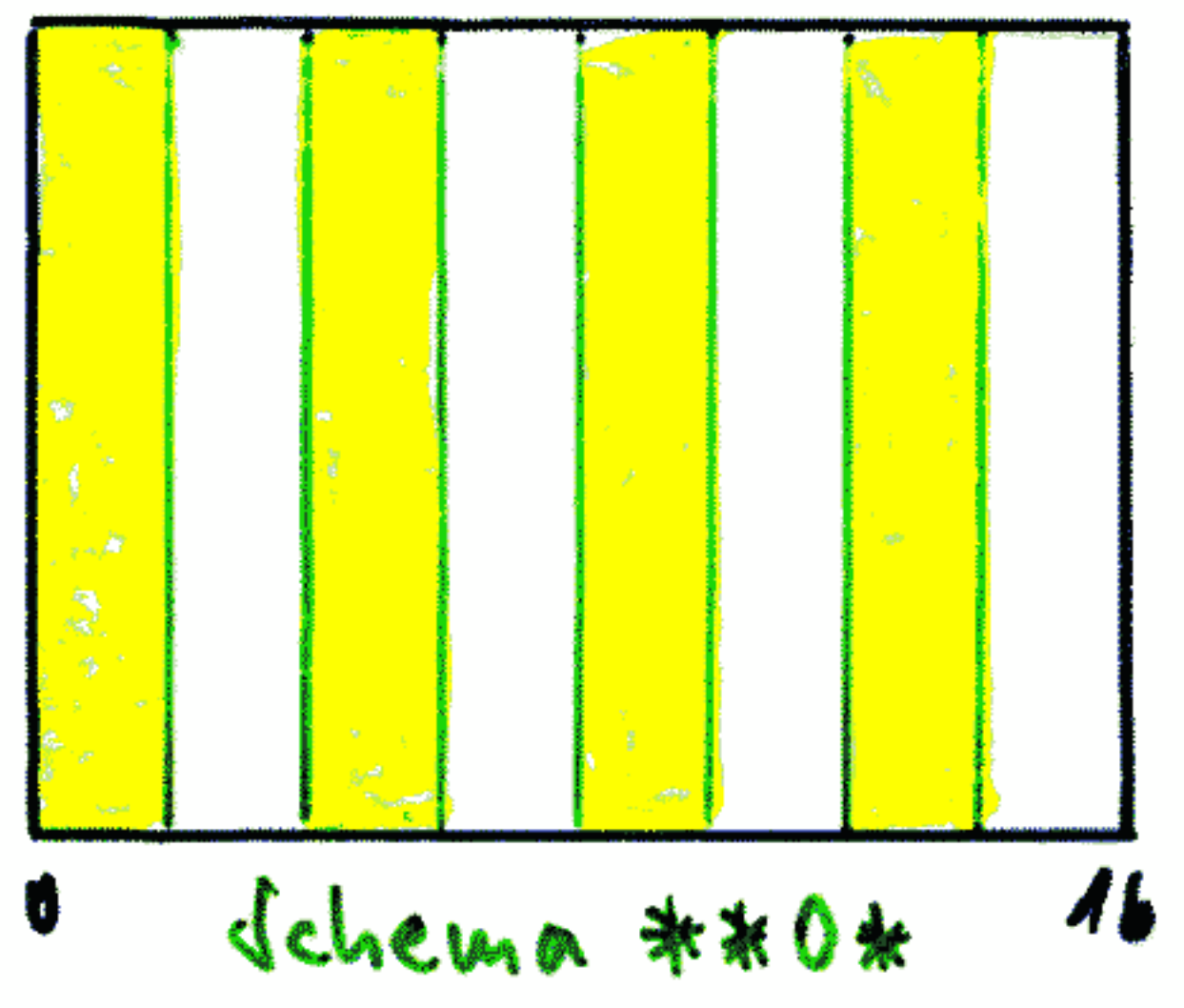
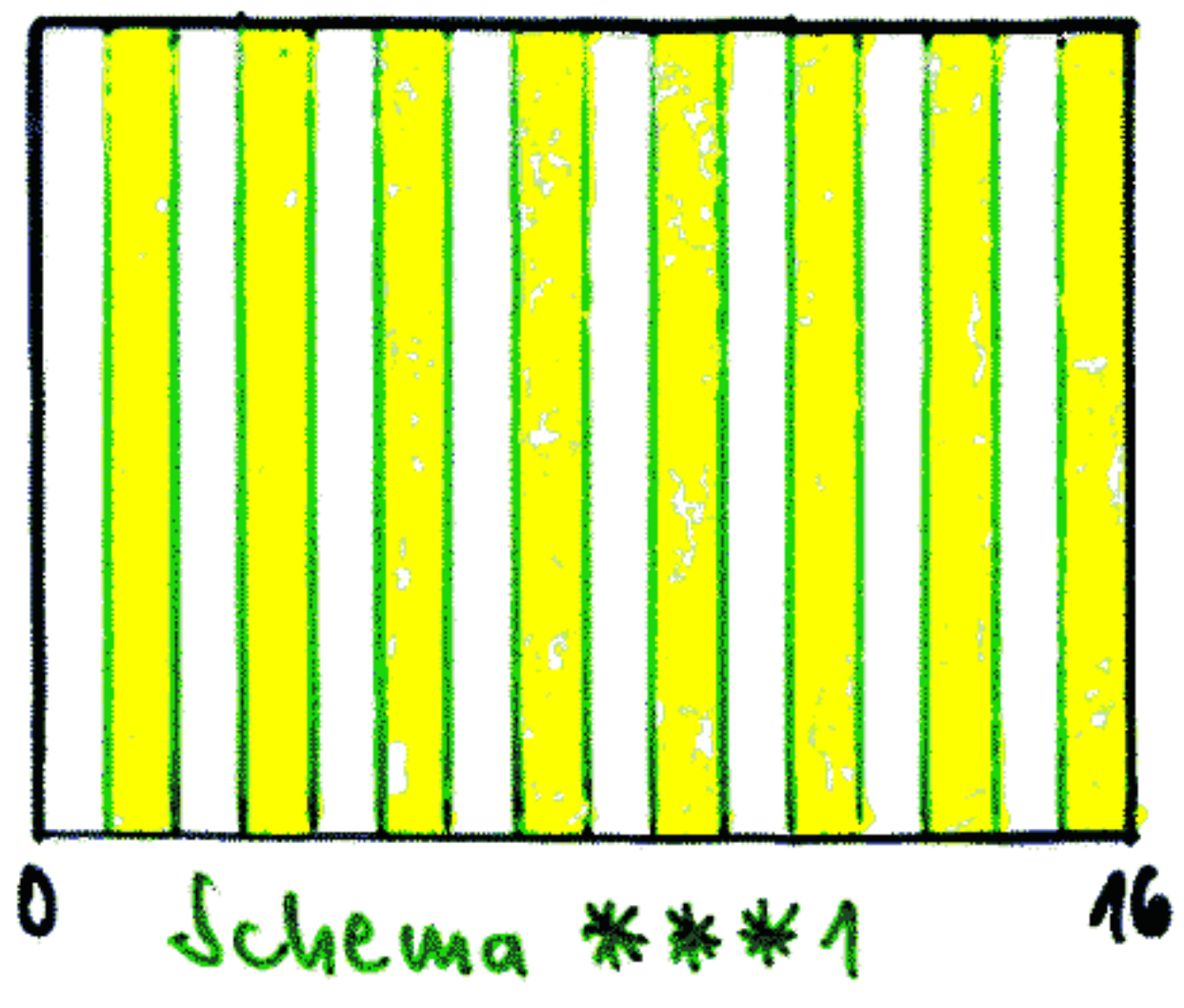
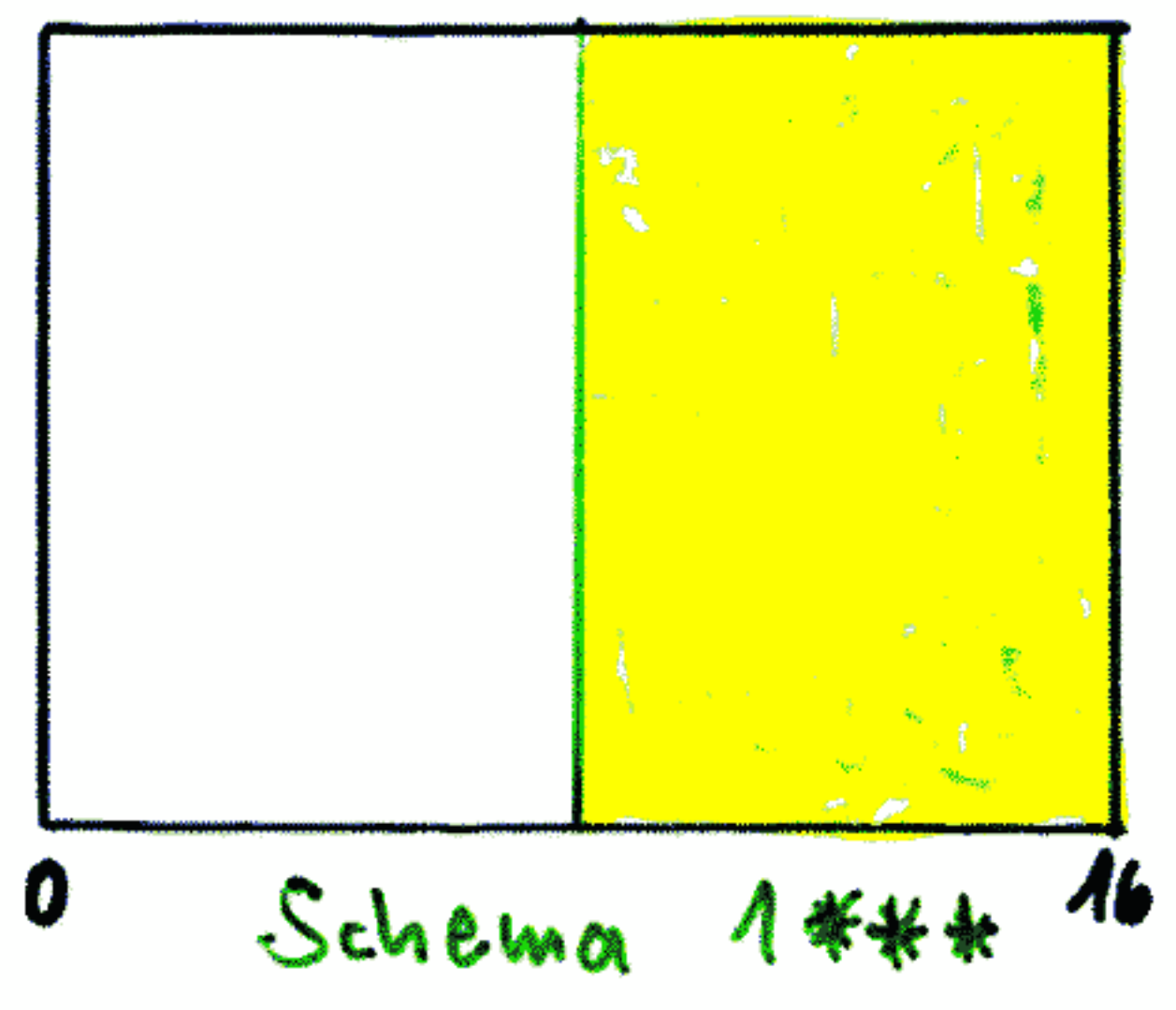
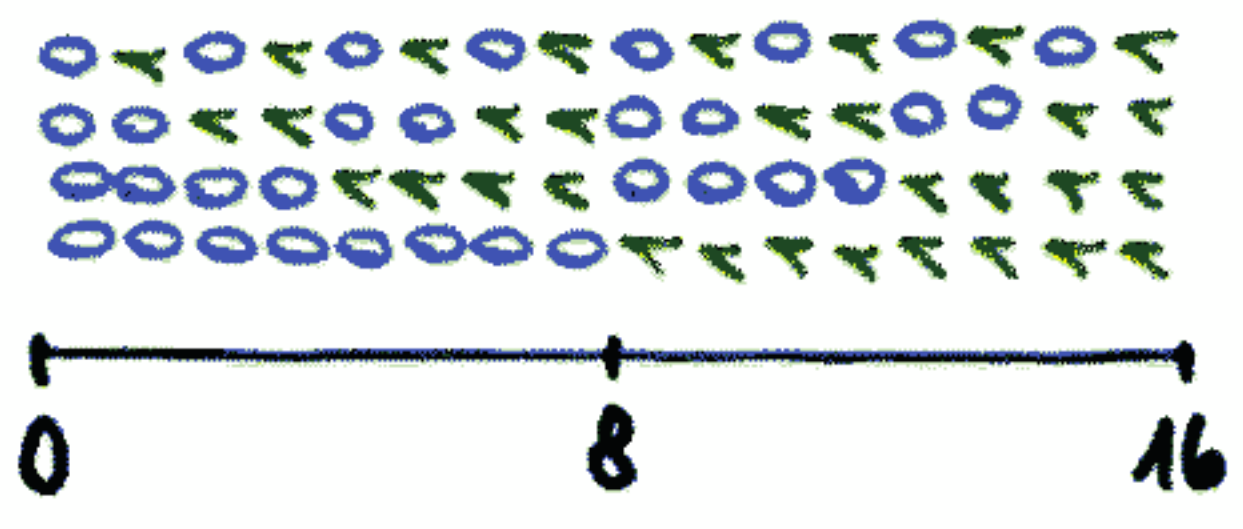
real coded GAs

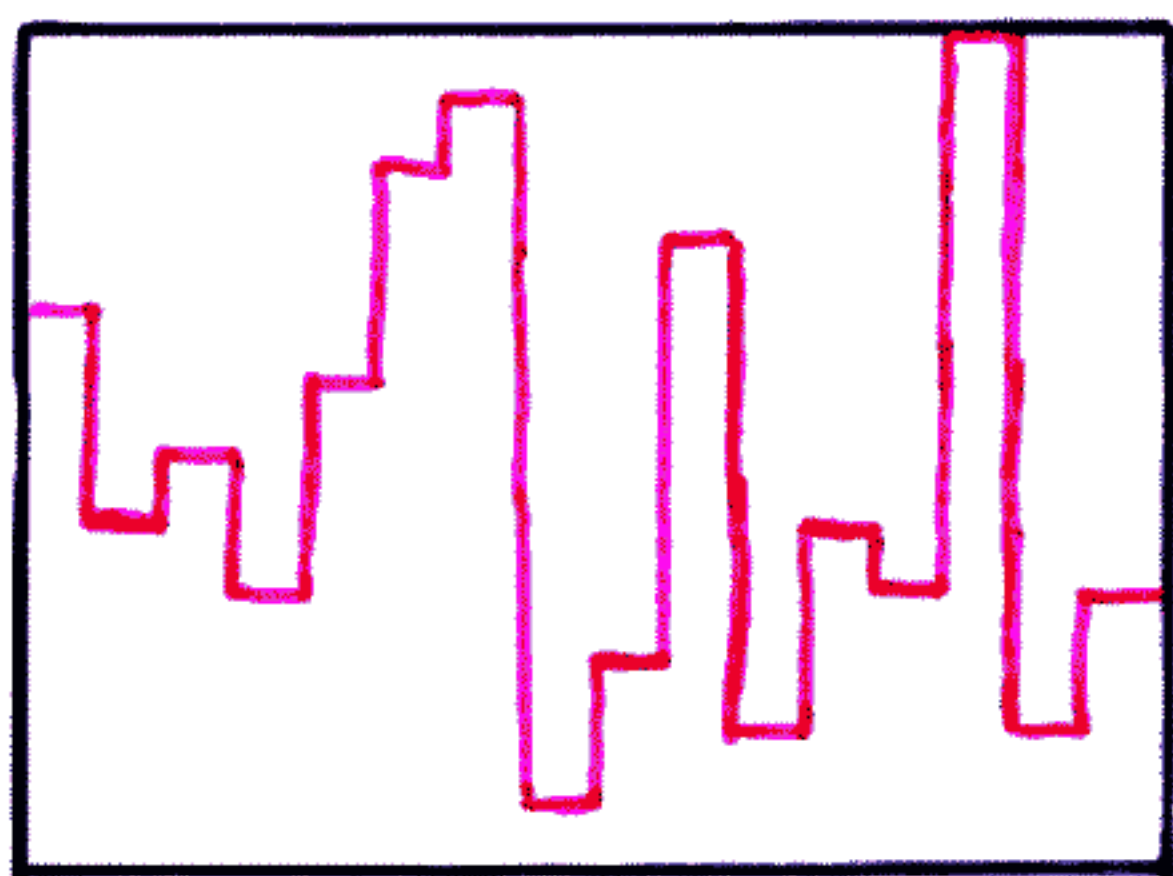
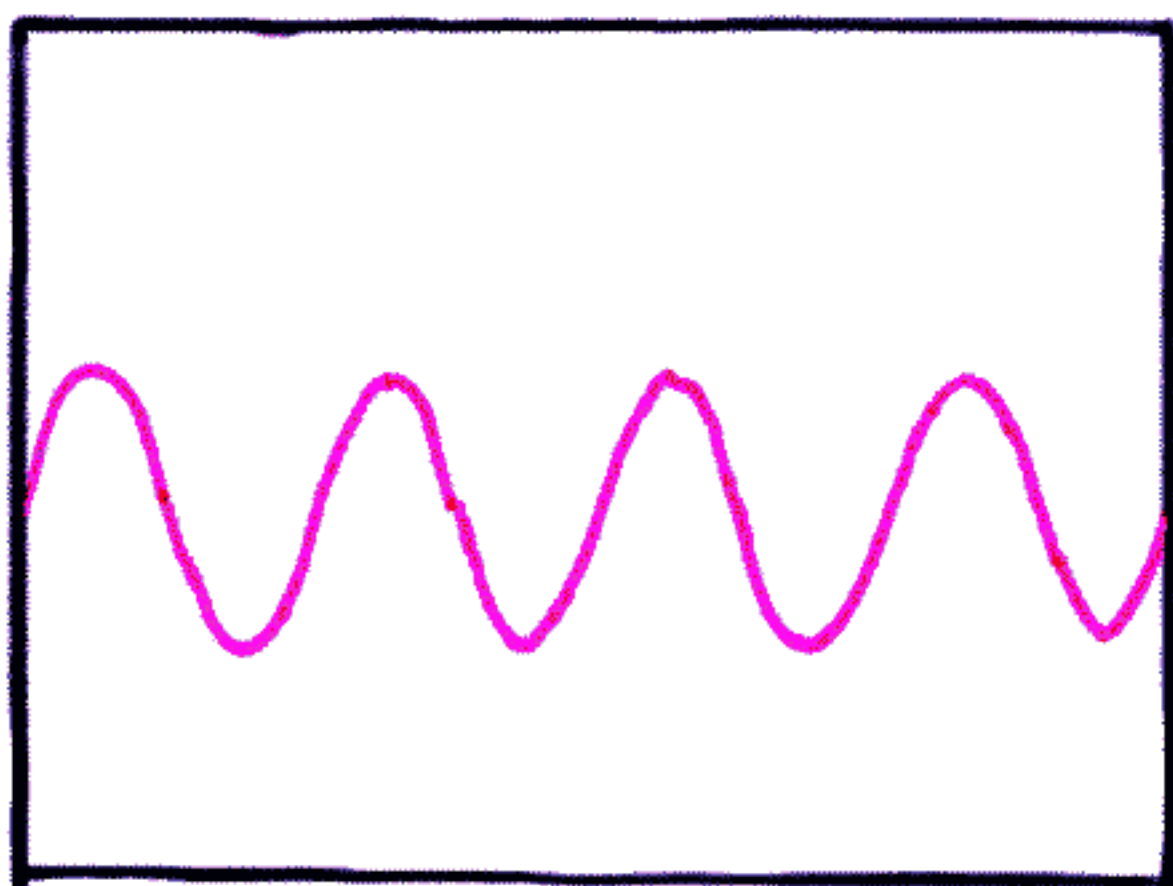
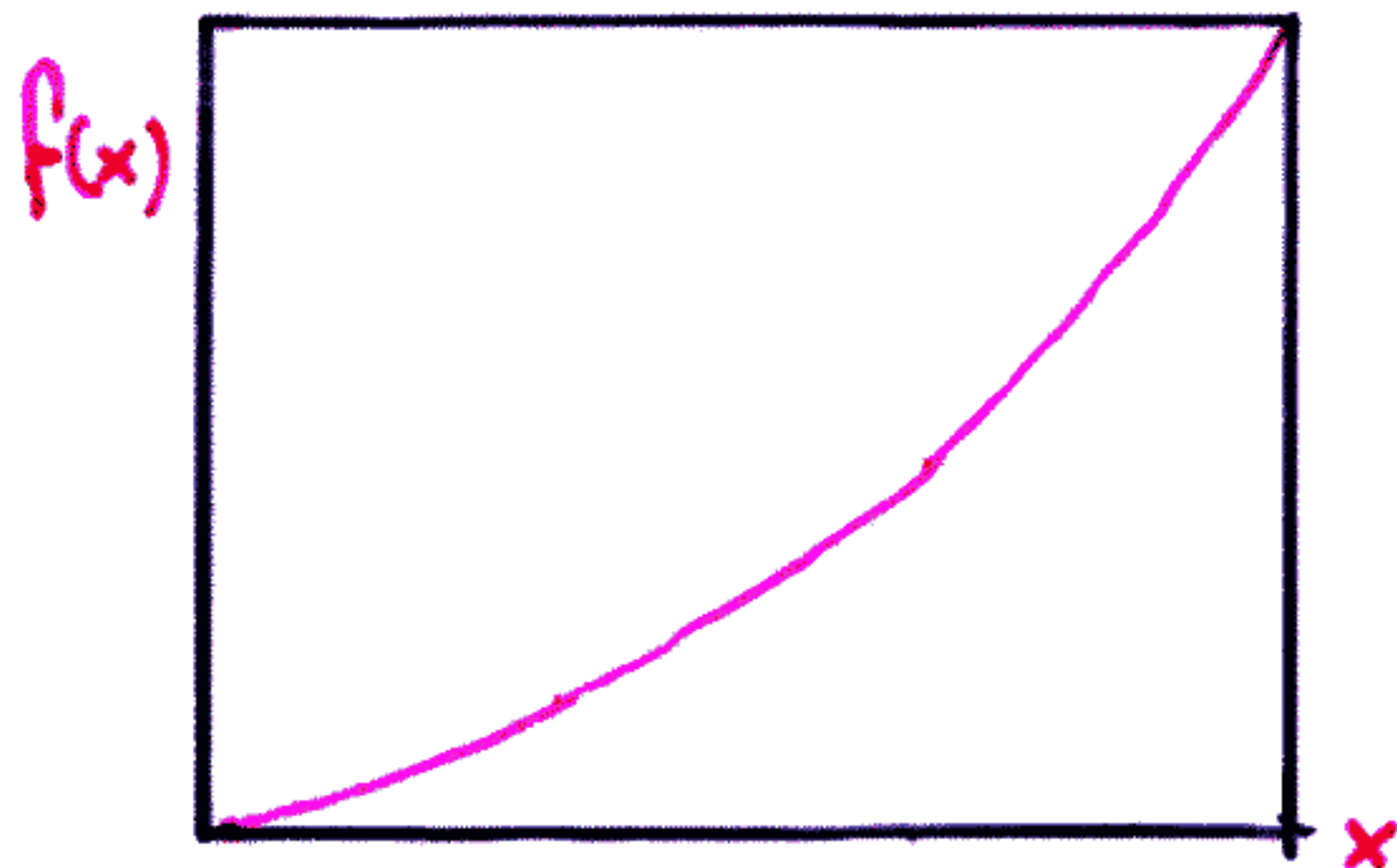
ESs gleichwertig oder besser

EP ?

Schemata $\hat{=}$
Hyperebenen

Bsp.: $l = 4$ Bits





Schema - Theorem (J. Holland, 1975)

binäre Hyperebene $H \in \{0, 1, *\}^L$
| wild card

z.B. $H_1 = (1****)$

$H_2 = (01*1*)$

zugehörige Instanzen $I(H)$

$I(H_2) = \{(01010), (01011), (01110), (01111)\}$
| h_1 | h_2

Ordnung $o(H) = |\{i \mid h_i \in \{0, 1\}\}|$

Bsp. $o(0**1*) = 2$

definiierende Länge $\Delta(H) = \max\{i \mid h_i \in \{0, 1\}\} - \min\{i\}$

Bsp. $\Delta(0**1*) = 4 - 1 = 3$

Anzahl

$$N_{t+1}(H) = N_t(H) \cdot \frac{\Phi(H)}{\bar{\Phi}} \cdot \left[1 - \frac{p_c \Delta H}{L-1} \left(1 - \frac{N_t(H)}{\lambda} \right) \right] (1 - p_m)^{o(H)}$$

$$\approx N_t(H) \cdot \frac{\Phi(H)}{\bar{\Phi}} \cdot \left[1 - \frac{p_c \Delta H}{L-1} - o(H) \cdot p_m \right]$$

↳ kurze überdurchschnittlich gute Schemata geringer Ordnung (building blocks) vermehren sich exponentiell

[und diese werden dann durch Crossover zusammengefügt]

$$\bar{\Phi} = \frac{1}{\lambda} \sum_{i=1}^{\lambda} \Phi_i$$

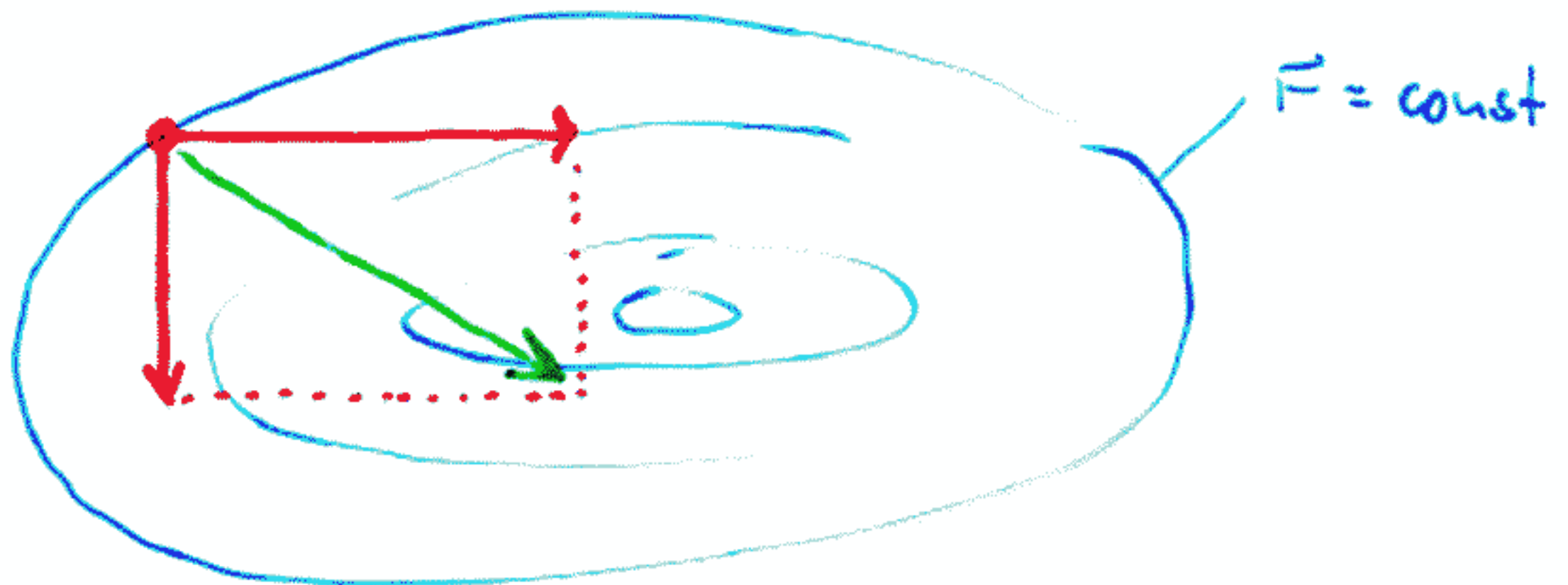
Kritik am Schema-Theorem

- Approximationen im Beweis gültig für $\lambda \rightarrow \infty$
- in endlicher Population kommen nicht alle Schemata vor
- bei exponentiellem Wachstum in endlichen Pop. sterben andere Schemata vollständig aus
- Crossover und Mutation nur destruktiv?
- Wachstum wäre nur dann exponentiell, wenn Wachstumsrate const.
(das ist aber nicht der Fall; sie geht gegen Null)
- ‚Zusammenfügen‘ brauchbarer Bruchstücke setzt Dekomponierbarkeit voraus
(i.a. müssen sich Variablen gleichzeitig ändern)

Nur noch ‚Hardliner‘ verteidigen das Schema-Theorem

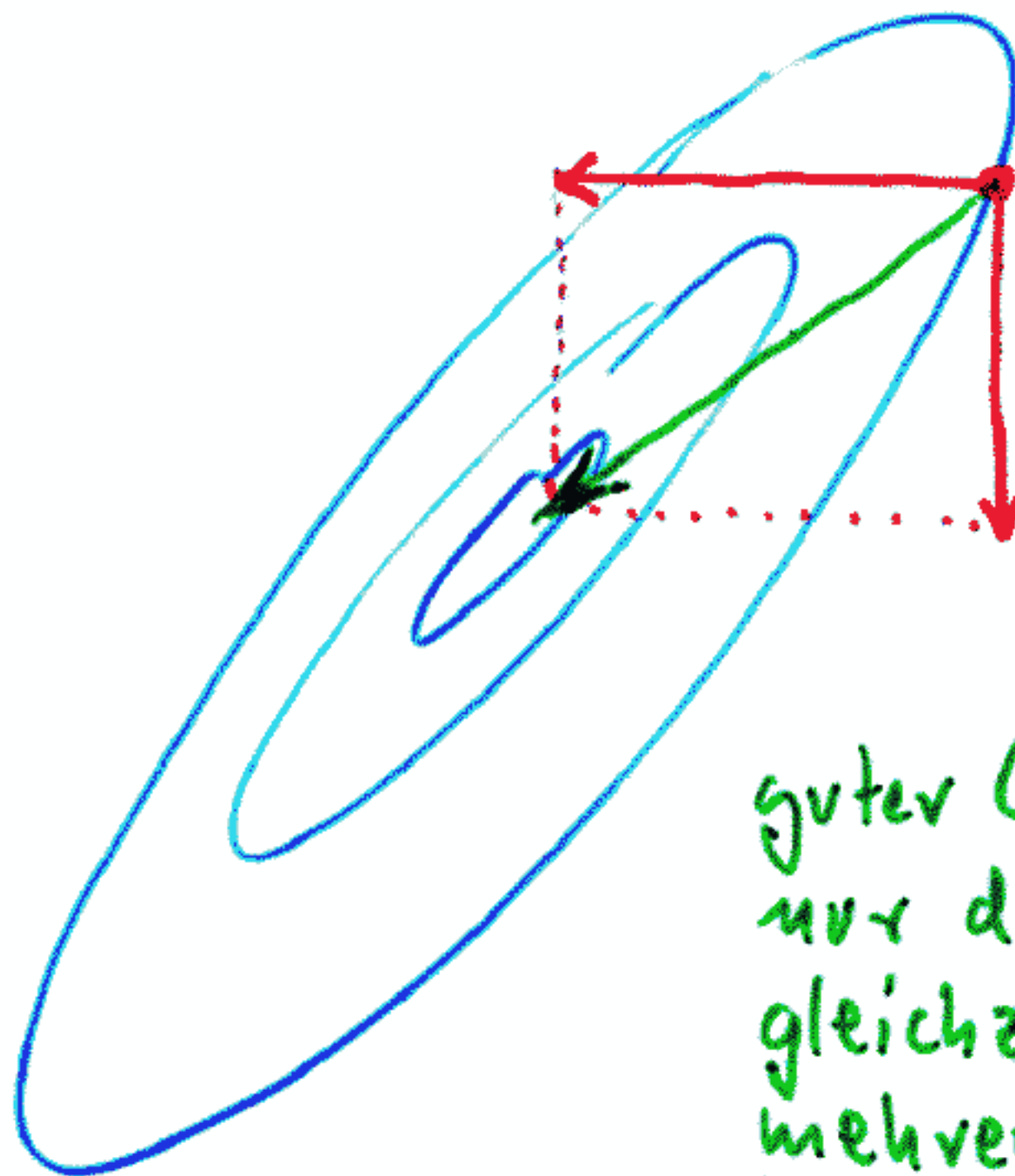
Kritik an der building block Hypothese

Fall 1



guter Gesamtschritt = Zusammensetzung
verbessernder Einzelschritte

Fall 2



guter Gesamtschritt
nur durch koordinierte
gleichzeitige Änderungen
mehrerer Variabler
(Teile des Bitstrings)

bb Hypothese gilt für separable Zielfunktion

Grgs: deceptive problems
(täuschend, betrügerisch)

implicit parallelism

jedes Schema testet gleichzeitig die
Brauchbarkeit einer ganzen Hyperebene

es werden aber nur gerade vorhandene Instanzen,
keine Schemata (summarisch) bewertet

Schema-Theorem geht von ∞ großer Pop. aus;
darin enthalten ist dann aber bereits
global optimale Lösung;
diese muß sich nur noch durchsetzen!

Alle 'optimalen' Bits sind schon in relativ
kleiner Population vorhanden

Sie brauchen 'nur noch' richtig zusammengestellt
zu werden (\rightarrow Primat der Rekombination)

? Wahrscheinlichkeit, daß zufällige Startpopulation alle korrekten Bits der optimalen Lösung enthält

L Länge des Bitstrings

λ Zahl der Individuen

gesuchte Wahrsch. : $w = 1 - L 2^{-\lambda}$

Bsp. $L = 1000$

$\lambda = 50$

$w \approx 0.999\ 999\ 999\ 999 = 1 - 10^{-12}$

Leider gehen korrekte Bits oft zwischendurch verloren. \rightarrow Mutationen, um sie wiederzugewinnen

aber

Mutation ist sog. Hintergrundoperator

CGA : $p_m = 10^{-3}$

Rekombination (Crossover) ist Hauptoperator

CGA : $p_c = 0.6$

Selektion : heute meist rangbasiert

\rightarrow geringere Abh. von Gradient

oder tournament selection

Beispiel: TSP

Travelling Salesperson Problem
man



n Städte

? kürzeste Tour

lange Zeit Paradebeispiel für Brauchbarkeit der GAs:
liefern relativ schnell gute Näherungslösungen

spezialisierte deterministische Strat. versus
spezialisierte GA

Frage der Modellierung
Datenstruktur für Individuen/Zustände

z.B. Städte durchnummern $1, 2, \dots, n$

gesucht: Permutation π

z.B. Adjazenzmatrix

gesucht: Belegung mit Nullen u. Einsen
(je eine Eins pro Zeile u. Spalte)

Zyklus-Crossover (CX):

$$\begin{array}{r}
 C = (\underline{9} \ 8 \ 2 \ \underline{1} \ 7 \ \underline{4} \ 5 \ 10 \ \underline{6} \ 3) \\
 D = (\underline{1} \ 2 \ 3 \ \underline{4} \ 5 \ \underline{6} \ 7 \ 8 \ \underline{9} \ 10) \\
 \quad \quad \quad \quad \quad \quad \downarrow \\
 C' = (9 \ - \ - \ 1 \ - \ 4 \ - \ - \ 6 \ -) \\
 \quad \quad \quad \quad \quad \quad \uparrow \uparrow \quad \quad \uparrow \quad \quad \uparrow \uparrow \quad \quad \uparrow \\
 D' = (1 \ - \ - \ 4 \ - \ 6 \ - \ - \ 9 \ -) \\
 \quad \quad \quad \quad \quad \quad \downarrow \\
 \hat{C} = (9 \ 2 \ 3 \ 1 \ 5 \ 4 \ 7 \ 8 \ 6 \ 10) \\
 \hat{D} = (1 \ 8 \ 2 \ 4 \ 7 \ 6 \ 5 \ 10 \ 9 \ 3)
 \end{array}$$

Garantiert, daß *jede* Stadt positionstreu entweder von Elter 1 oder Elter 2 übernommen wird.

Fazit zur TSP-Rekombination:

⇒ Kreativität bei der Operatorentwicklung ist nötig !

- Kaum theoretische Hinweise.
- Experimentell:
 OX ca. 11% besser als PX,
 OX ca. 15% besser als CX.
- Literatur: Oliver, Smith, Holland 1987.

Mutation / Inversion

Paaraustausch-Mutation:

$$A = (9 \ 8 \ \underline{4} \ 5 \ 6 \ 7 \ 1 \ \underline{3} \ 2 \ 10)$$

$$\Downarrow$$

$$\hat{A} = (9 \ 8 \ 3 \ 5 \ 6 \ 7 \ 1 \ 4 \ 2 \ 10)$$

Gute Näherungslösungen ($n = 50, 100$) (Fogel 1988).

Inversion:

$$A = (9 \ | \ 8 \ 4 \ 5 \ 6 \ 7 \ | \ 1 \ 3 \ 2 \ 10)$$

$$\Downarrow$$

$$\hat{A} = (9 \ | \ 7 \ 6 \ 5 \ 4 \ 8 \ | \ 1 \ 3 \ 2 \ 10)$$

(Entspricht einem 2-er Kantenaustausch; wie in Lin's 2-opt Heuristik).

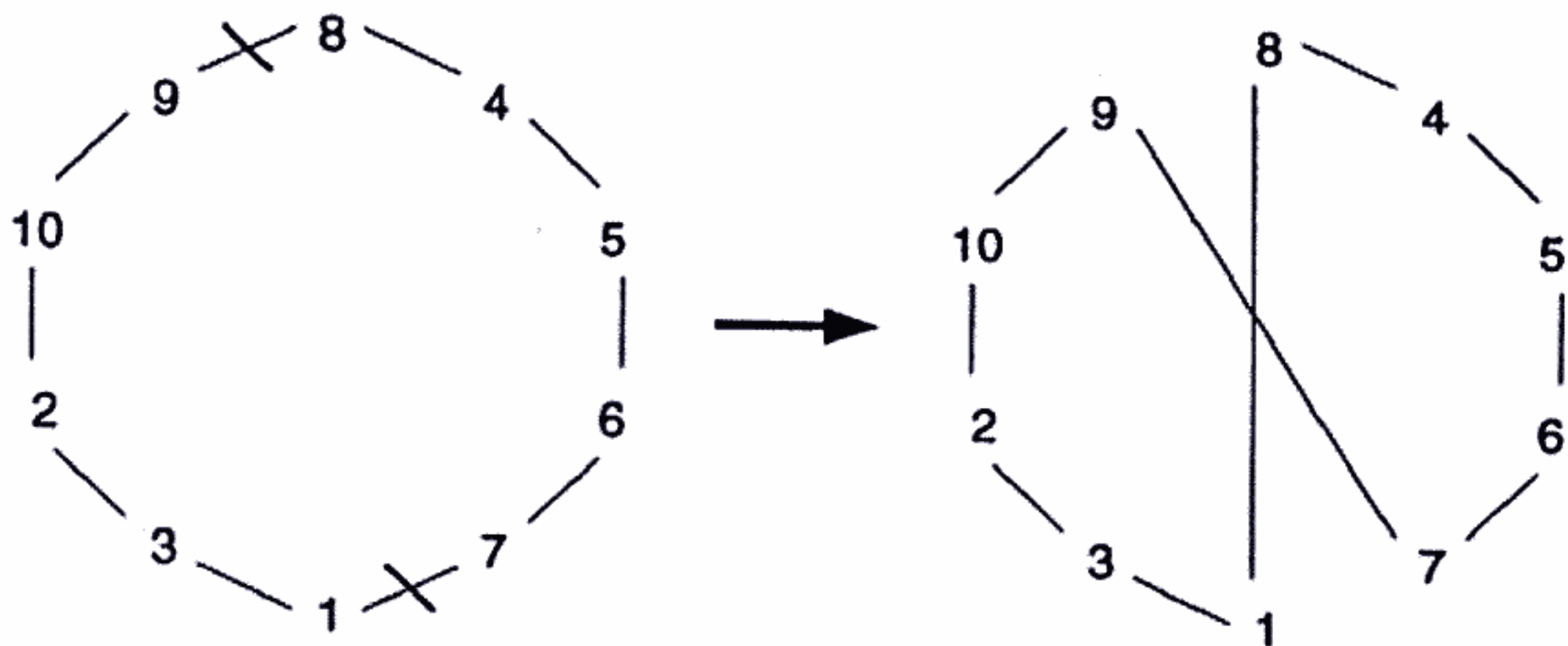


Abbildung 12: 2-er Kantenaustausch durch Inversion.

etwas Theorie (hier GA)

- GA (kanonisch) konvergiert nicht, sondern fluktuiert um einen Fixpunkt (hoffentlich ein lokales Optimum); J. Holland beharrt: GA ist keine Optimierungsmethode.
- ‚Premature convergence‘ besagt ebendies.
- Laut Fischer-Eigen Modell kann ein cGA nicht einmal ein Optimum halten, wenn die Mutationsrate zu groß ist und/oder die Bitstrings zu lang.
- Hollands Schema-Theorem ist nicht konstruktiv, es behandelt beide Variationsmechanismen (Crossover und Mutation) als nur schädlich, gilt für unendlich große Populationen und wird oft falsch interpretiert (‚exponentielles Wachstum guter Schemata niedriger Ordnung und geringer Länge‘).
- Die Building-Block Hypothese gilt nur für separable Fitness-Funktionen.
- Ein ‚elitist‘ GA (bisher beste Lösung bleibt) konvergiert aber.
Im linearen Fall: $O(n \log n)$ Schritte benötigt (z.B. ‚counting ones‘ Problem).
In schlimmsten Fall (‚epistasis‘): $O(n^n)$ Schritte benötigt.

EP 1962

finite-state automaton for prediction

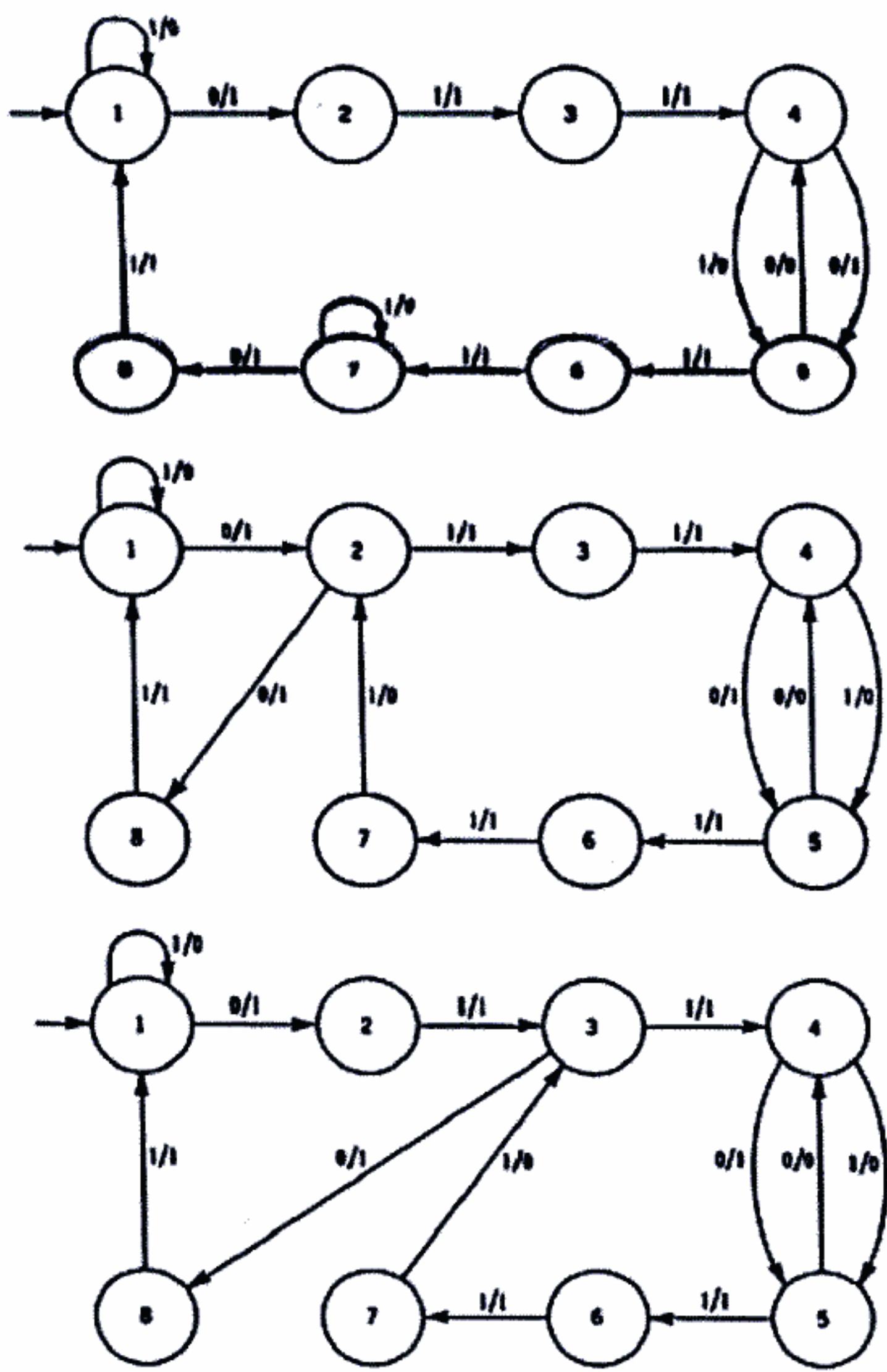


Fig. 3.4. Three different eight-state machines which perfectly predict 101110011101 cyclic environment.

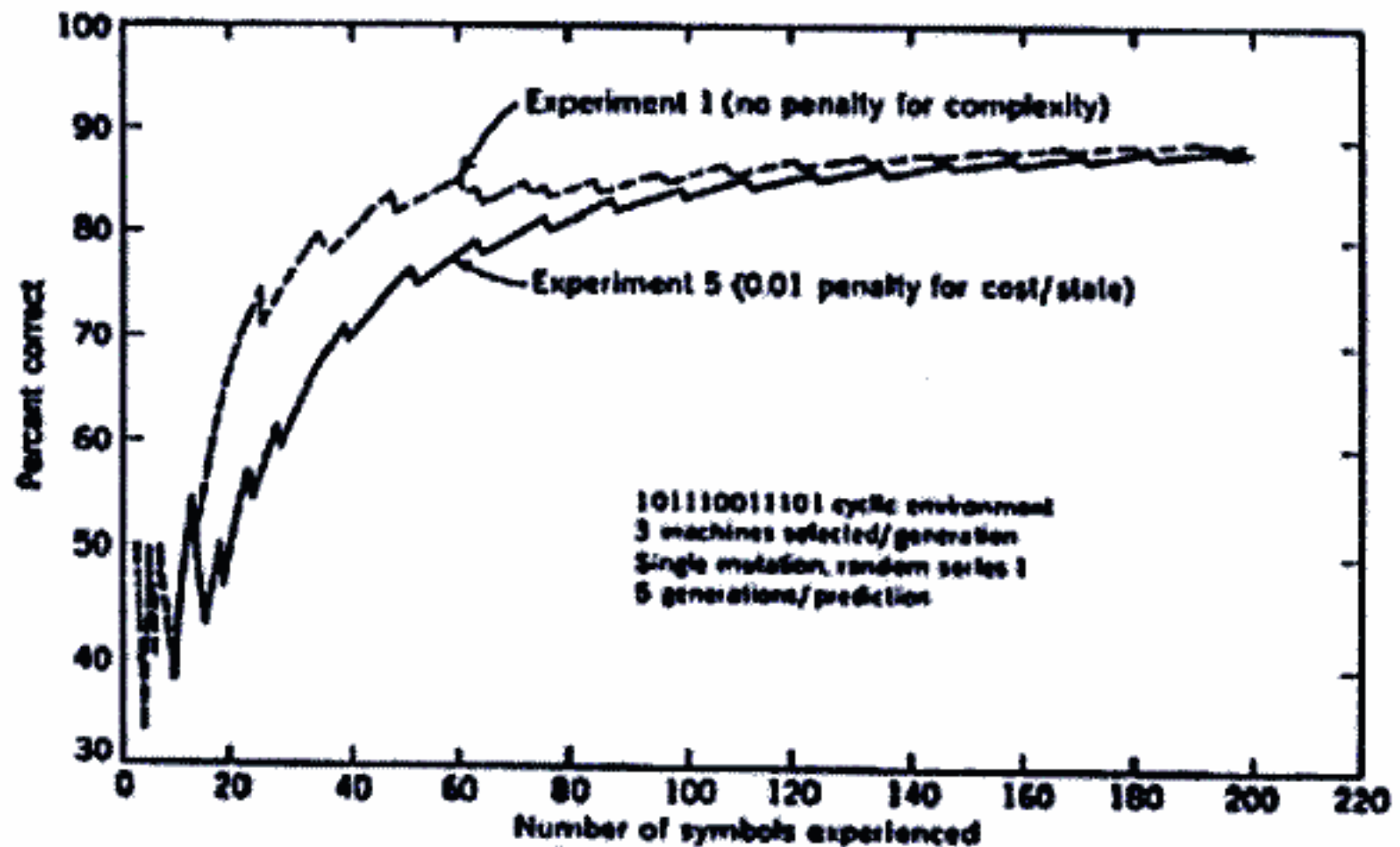


Fig. 3.2. A comparison of the prediction score with and without a penalty for complexity.