4 Genetische Algorithmen

Gegeben: ***Problem*** P

Menge ***potentieller Lösungen*** $S= \{L_1,\ldots,L_s\}$
(auch **Suchraum**)

Zielfunktion $z : S \rightarrow [0,1] \subset IR$
(manchmal auch $z : S \rightarrow IR$)

$z(L) > z(L') \quad \Leftrightarrow$ L ***besser*** als L'

Gesucht: **optimale Lösung L**

(manchmal auch: **Menge optimaler Lösungen**)

$m := \max \{ z(L) \mid L \in S \}$

$opt(P) := \{ L \in S \mid z(L)=m \}$

Beispiele: (kleine zufällige Auswahl)

- Punktepaar minimalen (max.) Abstands
- Inversion von Matrizen
- Nullstelle für reelle Funktion
- Tourenplanung
- Stundenplanentwurf
- Mengenüberdeckungsproblem
- Entwurf effizienter Schaltkreise
- Pflanzenzüchtung (z.B. möglichst hoher Ertrag)

verfügbare Methoden:

- direkte Berechnung der Lösung (Matrixmultiplikation)
- iteratives Verfahren (Newton,…)
- systematische Suche (Branch-&-Bound, …)
- probabilistische Verfahren (randomisierte Suche,…)
- Züchtungsmethoden (Selektion, Kreuzung, Mutation,…)

Ansatz der genetischen Algorithmen:

Verwende Prinzipien aus dem Bereich der (Tier- oder Pflanzen-) Züchtung für eine

- systematische randomisierte Suche.

Beispiel für die Vorgehensweise der Züchtung:

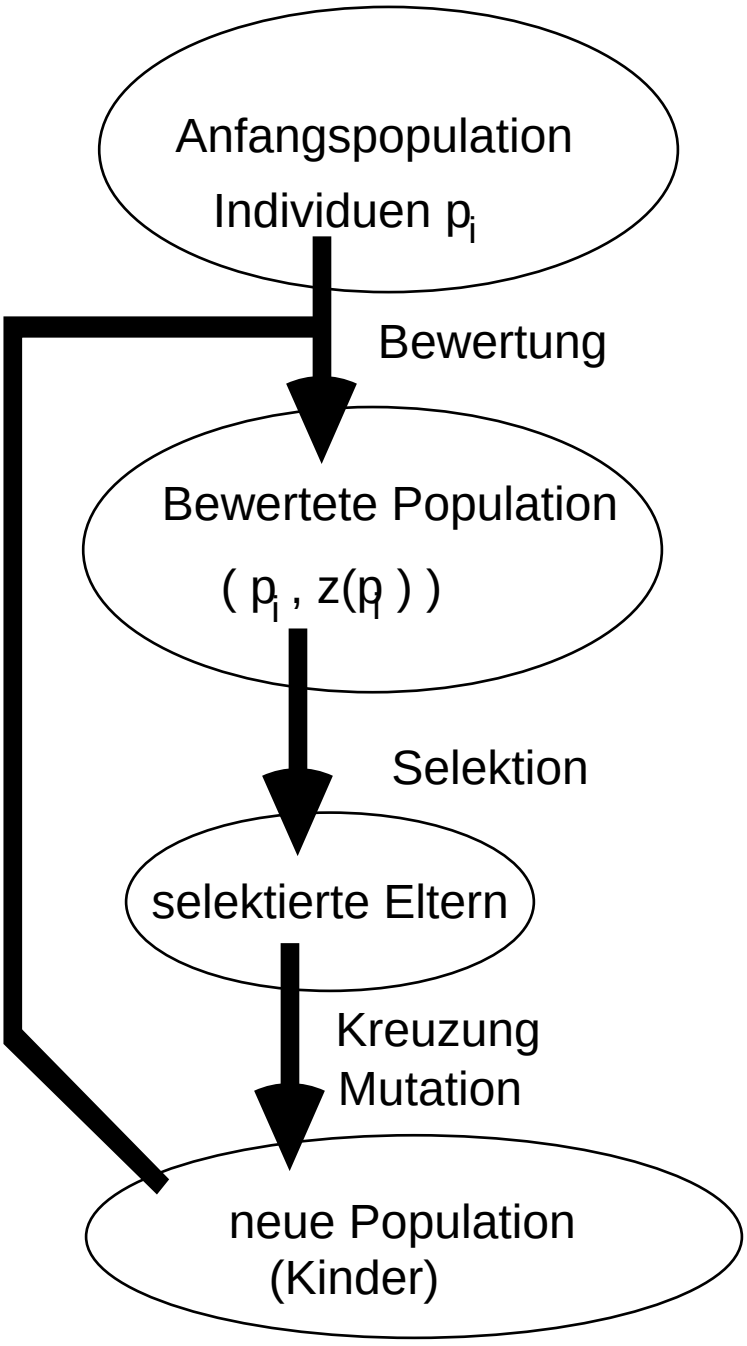
Aufgabe: Züchte Schweine mit möglichst langen Ohren.

Gegeben: Beliebige Menge von Ferkeln.

Vorgehensweise:

1) Ermittle die Ohrlängen der ausgewachsenen Schweine. (*Bewertung*)
2) Wähle eine Menge von Schweinen (männlich und weiblich) mit möglichst langen Ohren. (*Selektion*)
3) Paare diese Schweine (jeden Eber mit jeder Sau) . (*Kreuzung*)
4) Wiederhole dies Verfahren mit dem erzeugten Nachwuchs, bis die Ohren lang genug sind.

Abstraktion des Züchtungsvorgangs:



⇒ **allgemeines Muster für genetischen Algorithmus**

zu behandelnde Fragen:

- **Wie repräsentiert man die Individuen (Objekte) einer Population?**

  Begriffe der Genetik/Züchtung:

  – **Phänotyp** = Erscheinungsform eines Individuums als Ergebnis aller erblichen und umweltbedingten Einwirkungen

  – **Genotyp** = Gesamtheit der im (Zell-)Kern (also in den Chromosomen) lokalisierten Gene, d.h. Gesamtheit des erblichen Anteils am Phänotyp

  – **Chromosom** = Träger der linear angeordneten Gene (d.h. der Erbanlagen) mit artspezifischer *Anzahl* , *Form* , *Struktur* und *Organisation*

  – **Gen** = Einheit des genetischen Materials (Erbanlage).
  Jedes Gen sitzt an eindeutig bestimmtem Locus eines Chromosoms. Die verschiedenen möglichen Werte eines Gens heißen **Allele**

  – **Genom** = haploider (einfacher) Chromosomensatz, Individuen können mehrfache gleichartige Genome besitzen (Diploide, Triploide, …, Polyploide)

<u>in klassischen genetischen Algorithmen:</u>

Indiviuum wird repräsentiert durch ein Chromosom, bestehend aus einer Folge binärer Gene

aber: viele andere Darstellungsformen (Datenstrukturen) denkbar und möglich

Beispiele:

1) Berechnung des Maximums einer reellen Funktion f im Intervall [a,b] mit Genauigkeit 10^{-6} :

   Jedes $x \in [a,b]$ ist ein Individuum, zu unterscheiden sind $(b-a)10^6$ verschiedene Individuen, darstellbar durch $\lceil \log(b-a)10^6 \rceil$ Binärziffern

   für a=-1, b=2 sind dies 22 Bit

2) Entwurf eines k-fach zusammenhängenden Graphen mit n Knoten :

   Jedes Individuum (d.h. jeder Graph) wird durch Boolesche n×n-Matrix (die Adjazenzmatrix) beschrieben, also durch n^2 binäre Gene.

3) Tourenplanung; Traveling Salesperson Problem

   Jedes Individuum (d.h. jede mögliche Tour) kann beschrieben werden durch

   - Folge von Knoten, die in dieser Reihenfolge besucht werden

   - Permutation von [n]
     aber: nicht jede Permutation ist eine Tour!

   - Folge von Kanten
     aber: nicht jede Folge ist eine Tour!

- **Wie bewertet man ein Individuum?**

  in der Züchtung:

  Ermittlung der **Leistung** eines Genotyps durch Messung von Eigenschaften des Phänotyps, die für die Erreichung
  des Zuchtziels von Bedeutung sind.

  – **Fitness** = relative Eignung oder Selektionswert, den (mindestens) 2 den gleichen Lebensraum bewohnenden Individuengruppen oder Genotypen haben,

  genauer: relative Zahl der an die nächste Generation weitergegebenen Nachkommen bei natürlicher Selektion

  $\Rightarrow$ Fitness ist kein absolut meßbarer Wert eines Individuums, sondern ein relativer Wert, ermittelt durch Betrachtung nachfolgender Generationen

  in genetischen Algorithmen:

  durch Berechnung des Zielfunktionswerts eines Individuums

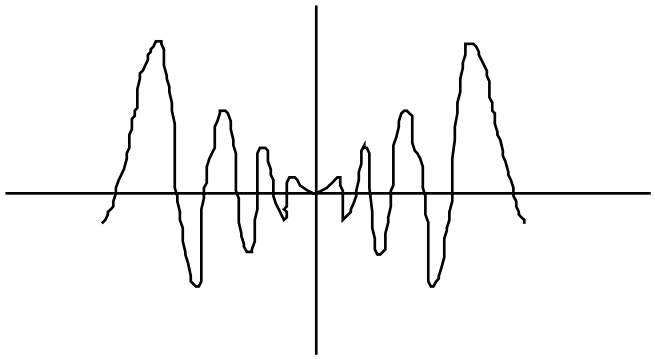
  leider wird dieser absolute Wert meist als Fitness bezeichnet, dies widerspricht jedoch der biologischen Bedeutung

Beispiele:

1) Berechnung des Maximums einer reellen Funktion f im Intervall [a,b] mit Genauigkeit 10^{-6} :

   Bewertung der Leistung durch Berechnung von f(x).

   z.B. $f(x) = x \cdot \sin(10\pi \cdot x) + 10$



2) Entwurf eines k-fach zusammenhängenden Graphen mit n Knoten :

   Berechnung des Zusammenhangsgrades .

3) Tourenplanung; Traveling Salesperson Problem

   Berechnung der Weglänge

- **Wie selektiert man Eltern?**

in der Züchtung:

– Auswählen der Individuen mit hoher Leistung

– Aussondern von Individuen mit schlechter Leistung

unterschiedlich je nach Züchtungsverfahren

Auslese-, Kreuzungs-, Mutationszüchtung,…

in genetischen Algorithmen:

– Auswahl der besten k Individuen

– Bestimmung der relativen Leistung, d.h. im Verhältnis zur Gesamtleistung der Population, Auswahl der Eltern entsprechend dieser relativen Leistung; dabei ist die Wahrscheinlichkeit für die Selektion eines Individuums durch diese relative Leistung bestimmt.

genauer: Population $M=\{p_1,\ldots,p_n\}$

$$\lambda(M):=\sum_{i=1}^{n} z(p_i) \quad \pi_i:=z(p_i)/\lambda(M) \quad q_i:=\sum_{j=1}^{i}\pi_i \quad q_0:=0$$

wiederhole k mal:

r:= randomreal(0,1) (* Zufallszahl zwischen 0 und 1*)

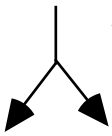
Selektiere p_i mit $q_{i-1} < r \leq q_i$.

$\Rightarrow$ Bei dieser Art der Selektion entspricht die relative Leistung der Fitness im biologischen Sinn.

- Wie entsteht aus den Eltern eine neue Generation?

<u>in der Natur/Züchtung</u>:

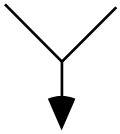
diploide Genotypen



Aufspaltung (Meiose)

mit Kopplungsbrüchen (crossing-over) zwischen homologen Chromosomen verschiedener Genome

Gameten mit halber Chromosomenzahl

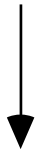


<u>Kreuzung</u>

Vereinigung verschiedener Gameten

neue diploide Genotypen (Zygoten)

neue Genomkombinationen sowie -
durch crossing-over - neue Genome



<u>Mutation</u>

Veränderung der genetischen Information

neue Genotypen

Es gibt

– **Genmutation** : Entstehung neuer Allele

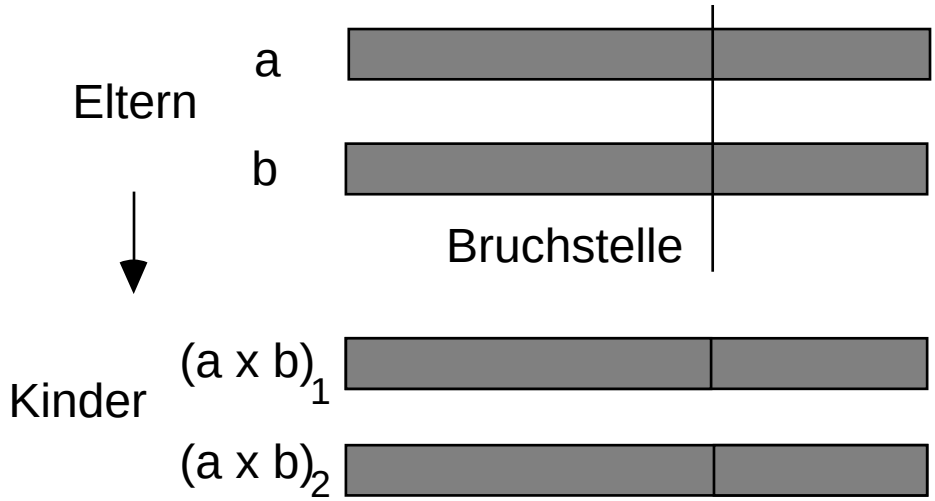
– **Chromosomenmutation** : Veränderung der Struktur von Chromosomen

– **Genommutation** : Veränderung der Zahl von Chromosomen und/oder Genomen

in genetischen Algorithmen:

Erzeugung von **Nachkommen** durch

- **Rekombination** (Kreuzung, **crossing-over**)



Die Bruchstelle wird zufällig gewählt.
Eltern werden ersetzt durch die Kinder.
(**Alternative**: Wähle aus Eltern und Kindern die beiden besten aus)

Welche Eltern werden gepaart?

Sei M_S die Menge der selektierten Individuen.

a) Für jedes $(a,b) \in M_S$ **Kreuzungswahrscheinlichkeit** p_c.

b) Wähle $a \in M_S$ mit Wahrscheinlichkeit p_c als Elter. Paare alle so erhaltenen aktiven Eltern zufällig.

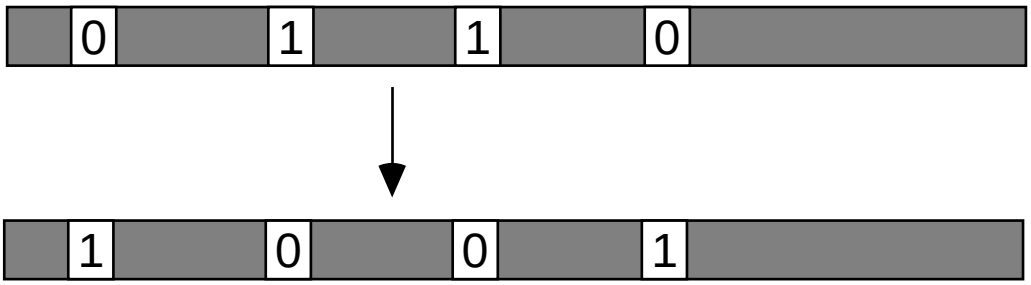
c) Wähle Teilmengen $E, E' \subseteq M_S$ aus (zufällig oder nach anderen Selektionskriterien). Paare jedes $a \in E$ mit jedem $b \in E'$.

Spezialfälle: $E \cap E' = \emptyset \Rightarrow$ **hierarchische Kreuzung**

$E=E'=M_S \Rightarrow$ **diallele Kreuzung**

Durch Kreuzung entstehende Population M_K
(je nach Verfahren mit oder ohne Eltern)
wird durch **Mutation** weiter verändert:

– **Mutation:** zufällige Veränderung von Genen



Für jedes Bit **Mutationswahrscheinlichkeit** p_m.

$\Rightarrow$ neue Generation M'.

Erzeugung der nächsten Generation durch genetischen Algorithmus hängt ab von

- der Datenstruktur für Genotypen
- dem Selektionsverfahren
- dem Kreuzungsverfahren
- der Kreuzungswahrscheinlichkeit p_c
- der Entscheidung, ob Eltern ersetzt oder übernommen werden.
- der Mutationswahrscheinlichkeit p_m

alle diese Parameter können auch von Generation zu Generation verändert werden
(dynamische Anpassung an aktuellen Züchtungserfolg)

angestrebtes Ziel:

- nächste Generation bringt **höhere Gesamtleistung**
- nächste Generation enthält **höhere Maximalleistung**
  (in der Genetik heißt dies **Heterosis**)

Bemerkungen:

- Wenn Eltern stets durch ihre Nachkommen ersetzt werden, kann es durchaus sein, daß die nächste Generation schlechtere Leistung bringt als die Elterngeneration!
- Durch Rekombination und Mutation können unter Umständen unzulässige Genotypen entstehen. (z.B. unzulässige Touren beim TSP).

$\Rightarrow$ für jedes Problem müssen eventuell unterschiedliche Rekombinations- und Mutationsoperationen definiert werden, um sicherzustellen, daß stets zulässige Genotypen erzeugt werden!

$\Rightarrow$ sehr viele Freiheitsgrade, wieso führt das überhaupt zu sinnvollen Algorithmen?

Wieso verspricht man sich von genetischen Algorithmen die Erzeugung von Individuen besonders großer Leistung?

Annahmen: (gemäß Michalewicz)

- Genotyp = 1 Chromosom = Folge $a \in \{0,1\}^m = IB^m$
- Selektion entsprechend dem Anteil der individuellen Leistung an der Gesamtleistung
- Auswahl der Eltern mit Kreuzungswahrscheinlichkeit p_c
- zufällige Paarung der ausgewählten Eltern
- ersetze Eltern durch die Nachkommen
- Mutation aller Bits der neuen Population mit Mutationswahrscheinlichkeit p_m

Frage:

Wie groß ist für diesen Typ genetischer Algorithmen die Überlebenswahrscheinlichkeit (Fitness) von Genen oder Genmustern, insbesondere solcher Gen(muster), die einen großen Anteil an der Leistung der Population haben?

Genotyp $a \in \{ 0, 1 \}^m = I B^m$

Schema $s \in \{ 0, 1, * \}^m$

*: don' t care - Symbol

für $i \in [m]$ $s_i \in \{0,1\} \Rightarrow$ Gen hat festgelegten Wert

$s_i = * \Rightarrow$ Gen kann beliebigen Wert annehmen

Schema ist Muster, Maske, Baustein für Genotyp:

Schema s ist in Genotyp a enthalten: $s \leq a$

$\Leftrightarrow \quad \forall\ i \in [m] \qquad s_i \neq * \ \Rightarrow\ s_i = a_i$

Bsp.: $s = (* * 1 * 0 0 1 *)$

$s \leq a = (1 0 1 0 0 0 1 1)$

es gibt
- 2^m verschiedene Genotypen
- 3^m verschiedene Schemata

(offensichtlich)

Schema - Eigenschaften:

Sei $s \in \{ 0, 1, * \}^m$ ein Schema.

Ordnung von s: $o(s) := |\{i \in [m] \mid s_i \neq * \}|$

definierende Länge von s:

$$d(s) := \max_i s_i \neq * \; - \; \min_i s_i \neq *$$

Bsp.: $s = (* * 1 * 0 0 1 *)$

$s' = (1 0 * * * * 0 1)$

$\Rightarrow$ $o(s) = 4 = o(s')$

$d(s) = 4 \quad d(s') = m - 1 = 8$

Lemma

In jeder Population $M \in I B^m$ gibt es für jedes Schema $s \in \{ 0, 1, * \}$ maximal $2^{m-o(s)}$ verschiedene Genotypen a mit $s \leq a$.

(Beweis trivial)

Lemma

(a) Zu jedem Genotyp $a \in I\!B^m$ gibt es maximal 2^m verschiedene Schemata s mit $s \subseteq a$.

(b) In jeder Population $M \subseteq I\!B^m$ können zwischen 2^m und $\min(3^m, |M| \cdot 2^m)$ verschiedene Schemata enthalten sein.

(Beweis trivial)

Definitionen

a) s Schema, M Population, z Zielfunktion

$$\mu(s,M) := \left| \{a \in M \mid s \subseteq a\} \right|$$

Leistung von s in M:

$$\lambda(s,M) := \sum_{\substack{a \in M \\ s \subseteq a}} z(a) \Big/ \mu(s,M)$$

d. h. $\lambda(s, M)$ = mittlere Leistung aller s enthaltenden Genotypen von M.

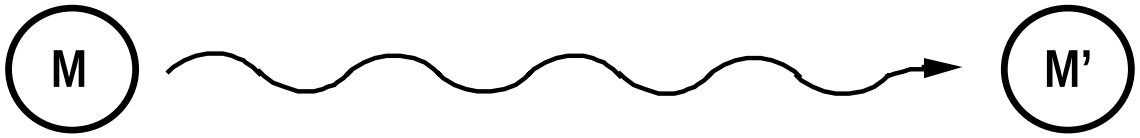
b) s Schema, $M_0, M_1, M_2, \ldots$ Populationsfolge

für $t \in I\!N_0$ sei

$$\lambda(s,t) := \lambda(s,M_t)$$
$$\mu(s,t) := \mu(s,M_t)$$
$$\lambda(t) := \lambda(M_t)$$

s Schema, M Population

Wie gut übersteht s eine Iteration des genetischen Algorithmus,
d. h. wie ist das Verhältnis von $\mu(s, M)$ und $\mu(s, M')$ wobei M' die Folgegeneration sei?



Antwort:

Wegen des randomisierten Vorgehens nur Aussagen über den erwarteten Wert von $\mu(s, M')$ möglich:

$\mu(s, M' \mid M)$:= erwartete Anzahl $\mu(s, M')$ ausgehend von M.

Schrittweise Antwort:

1) **Selektion:** $M_0 := M \rightarrow M_1$ und $|M_1| = n$

Bei jeder Ziehung: $a \in M$ mit $p_a = \frac{z(a)}{\lambda(M)}$

$\Rightarrow$ Wahrscheinlichkeit, daß irgendein Element a mit $s \subseteq a$ gezogen wird:

$$\sum_{s \subseteq a} z(a) / \lambda(M) = \mu(s,M) \cdot \lambda(s,M) / \lambda(M)$$

$$\Rightarrow \quad \mu(s, M_1 | M) = n \cdot \mu(s,M) \cdot \lambda(s,M) / \lambda(M)$$

$$= \mu(s,M) \cdot \lambda(s,M) / \overline{\lambda}(M)$$

$$\text{mit } \overline{\lambda}(M) = \lambda(M)/n$$

mittlere Leistung

$\Rightarrow$ Anzahl der s enthaltenden Genotypen verändert sich entsprechend dem Verhältnis der Leistung des Schemas zur mittleren Leistung der Population.

Sei $\lambda(s,M) = (1+\varepsilon)\,\bar{\lambda}\,(M)$.
Dann ist also

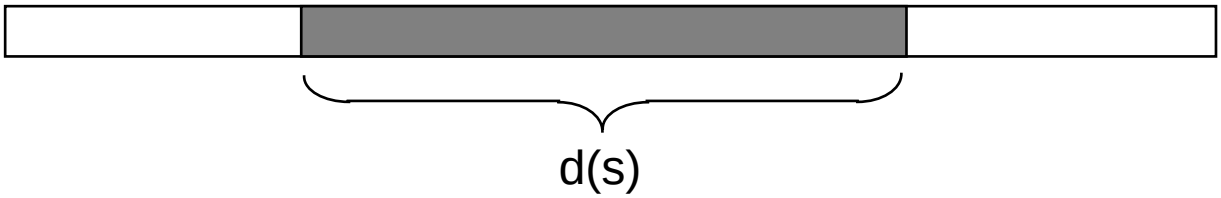
$$\mu(s,M_1|M) = \mu(s,M)\,(1+\varepsilon)$$

2) **<u>Rekombination / Kreuzung:</u>** $M_1 \rightarrow M_2$

Sei $a \in M_1$ mit $s \subseteq a$.

P(a wird als Elter gewählt) = p_c

a:



Bruchstelle ß wird zufällig gewählt, jede Position ist gleichwahrscheinlich.

$\Rightarrow$ P(ß zerteilt s) = $\frac{d(s)}{m\text{-}1}$

$\Rightarrow$ Wahrscheinlichkeit, daß nur ein Teil von s mit (einem Teil von) a nach M_2 übernommen wird, ist gleich

$$p_c \cdot \frac{d(s)}{m\text{-}1}\,.$$

Außerdem: durch Rekombination kann Schema s neu entstehen

$\Rightarrow$ $$\mu(s,M_2|\,M_1) \geq \mu(s,M_1)\cdot\left(1-\frac{d(s)}{m-1}\cdot p_c\right)$$

3) **Mutation**: $M_2 \rightarrow M_3$

In jedem $a \in M_2$ mit $s \subseteq a$ überlebt s die Mutation mit Wahrscheinlichkeit

$(1-p_m)^{o(s)} \approx 1 - o(s) \cdot p_m$ (da $p_m << 1$)

zusätzlich könnte s durch Mutation neu entstehen.

$$\Rightarrow \mu(s, M_3 | M_2) \geq \mu(s, M_2) \cdot (1 - o(s) \cdot p_m)$$

$\Rightarrow$ Kombination der drei Schritte:

$M_0 = M \rightarrow M_1 \rightarrow M_2 \rightarrow M_3 = M'$:

$$\mu(s, M' | M) \geq \mu(s, M_2 | M_1) \cdot (1 - o(s) \cdot p_m)$$

$$(\mu(s, 3|0)) = \mu(s, M_1 | M_0) \cdot \left(1 - \frac{d(s)}{m-1} \cdot p_c\right) \cdot (1 - o(s) \cdot p_m)$$

$$= \mu(s, M)(1+\varepsilon) \cdot \left(1 - \frac{d(s)}{m-1} \cdot p_c\right) \cdot (1 - o(s) \cdot p_m)$$

$\Rightarrow$ nach 2 Iterationen:

$M \rightsquigarrow M' \rightsquigarrow M''$

$$\mu(s, M'' | M) \geq \mu(s, M)(1+\varepsilon)^2 \cdot \left(1 - \frac{d(s)}{m-1} \cdot p_c\right)^2 (1 - o(s) \cdot p_m)^2$$

$\Rightarrow$ kurze, überdurchschnittlich leistungsfähige Schemata haben exponentiell wachsende Fitness!

Schema Theorem

Kurze Schemata überdurchschnittlicher Leistung vermehren sich exponentiell in aufeinanderfolgenden Generationen eines genetischen Algorithmus.

Dies ist ein Haupsatz der Theorie genetischer Algorithmen, er erklärt die Vorgehensweise, Systematik

aber: daraus folgt nicht, daß diese Vorgehensweise zu optimalen Lösungen führt!

aber trotzdem:

Bausteinhypothese:

Genetische Algorithmen suchen möglichst gute Lösungen (Individuen möglichst hoher Leistung) durch Aneinanderreihung kurzer Schemata überdurchschnittlicher Leistung, genannt Bausteine.

bisher nur empirische Hinweise, keine theoretischen Ergebnisse, die diese Hypothese unterstützen.

Schwierigkeiten:

- da der Erfolg eines Schemas s von d(s) abhängt, spielt die Anordnung der Gene eine entscheidende Rolle

⇒ Wahl der Datenstrukur hat entscheidenden Einfluß auf Erfolg eines genetischen Algorithmus

- wie ändern sich die Aussagen des Schema-Theorems, wenn man
  - p_m, p_c dynamisch verändert
  - multi-point cross-over macht
  - Genotypen aus mehreren Chromosomen betrachtet?
  - andere Kreuzungsverfahren wählt (hierarchisch?, diallel?)

⇒ wahnsinnig viele Fragen, die schwierig zu beantworten sind.

Vermutung:

wesentlicher Effekt beim Crossover ist die Tatsache, daß jeweils ein Teil der Gene eines Elters mit einem Teil der Gene des anderen Elters kombiniert wird.

Also: naheliegende Variante:

Wähle für die Rekombination eine Teilmenge $C \subseteq [m]$ zufällig aus.
Seien a, b Eltern. Erzeuge Nachkommen wie folgt:

$$\forall i \in [m] \quad (a \times b)_1 (i) := \begin{cases} a_i, \text{ falls } i \in C \\ b_i, \text{ sonst} \end{cases}$$

$$(a \times b)_2 (i) := \begin{cases} b_i, \text{ falls } i \in C \\ a_i, \text{ sonst.} \end{cases}$$

Mit welcher Wahrscheinlichkeit überlebt ein Schema s die Rekombination?

$s \in \{0, 1, *\}^m$, $C \subseteq [m]$

s überlebt $\Leftrightarrow$ entweder $\forall\ i \in C \quad s_i = *$
oder $\forall\ i \in [m]\ s_i \neq * \Rightarrow i \in C$

Jede Teilmenge von [m] ist gleichwahrscheinlich

$\Rightarrow$ $P(s \text{ überlebt}) \geq 2 \cdot 2^{-o(s)}$

$\Rightarrow$ Diese Variante des genetischen Algorithmus kann keine Bausteine bilden.

$\Rightarrow$ **Vermutung war falsch!**

aber:

- Es kann durchaus sein, daß auch diese Variante des GA zum Erfolg führt.
- Da keine Schemata bevorzugt überleben, werden in aufeinanderfolgenden Generationen viel mehr verschiedene Genotypen erzeugt.

  $\Rightarrow$ dies kann sich vorteilhaft auswirken

Wie aufwendig sind genetische Algorithmen?

Speicheraufwand:

$$\underbrace{\#\,\text{Gene}}_{m} * \underbrace{\text{Populationsgröße}}_{n}$$

+ Speicheraufwand bei Bewertung

Zeitaufwand:

$$\#\,\text{Generationen} * * \left(\underbrace{n \cdot t_z}_{\text{Bewertung}} + \underbrace{n \cdot t_{near}}_{\text{Selektion}} + \underbrace{n \cdot t_{cross}}_{\text{Kreuzung}} + \underbrace{n \cdot t_{mut}}_{\text{Mutation}} \right)$$

genauere Aussagen so allgemein kaum möglich.

aber sicherlich polynominelle Zeit, sofern n polynominell ist bzgl. der Anzahl Gene pro Genotyp und auch die Anzahl der Generationen nicht zu groß ist.

⇒ **Was für Aussagen kann die Analyse genetischer Algorithmen bringen?**

analog zu probablistischer Algorithmen:

- mit einer gewissen Wahrscheinlichkeit p wird nach k Generationen ein Wert erwartet, der um x Prozent vom Optimum abweicht.

aber

selbst derartige Aussagen sind für genetische Algorithmen schwierig.