Fachhochschule
Münster University of
Applied Sciences

Adaptive Systeme

Genetische Algorithmen

Prof. Dr. rer. nat. Nikolaus Wulff

Überblick

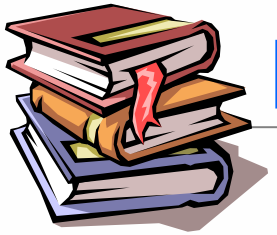
Einleitung

Adaptive Filter

Künstliche neuronale Netze

Adaptive Vektorquantisierung

Evolutionäre Algorithmen



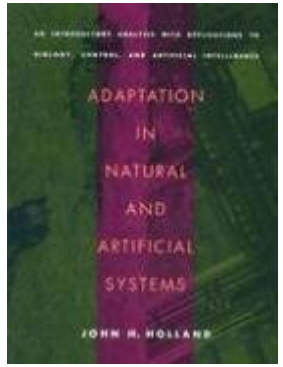
Literatur

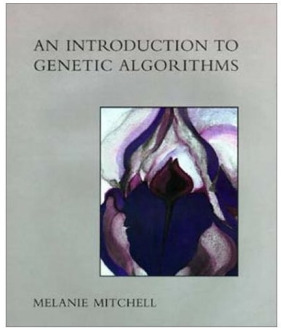
Holland, J.H.: Adaptation in Natural and Artifical Systems, First MIT Press Edition, 1992

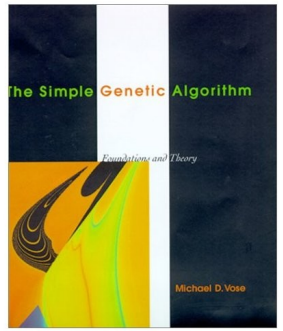
Mitchell, M.: An Introduction to Genetic Algorithms, Cambridge: MIT Press, 1996

Rechenberg, I.: Evolutionsstrategie ´94, frommann-holzboog, 1994

Vose, M.D.: The Simple Genetic Algorithm, MIT Press, 1999







Evolutionäre Algorithmen

Strategieansätze

Evolutionsstrategien

- „Kontiuumsgenetik“
- Lokale „Gradientendiffusion“
- Theorie auf Basis eines lokalen Fortschrittsgesetzes und asymptotisch für $N \gg 1$

Genetische Algorithmen

- „Diskrete Genetik“
- Nachbildung molekulargenetischer Mechanismen wie Crossover und Mutation
- Schema-Theorem

Evolutionäre Algorithmen

Genetische Algorithmen

Diskrete Repräsentation

Repräsentation der zu adaptierenden oder zu optimierenden individuellen Strukturen $a \in \Omega$ durch das *l*-Tupel
$a = (a_{0,} a_{1,} \ldots, a_{l-1})$
in dem Suchraum Ω

Für binäre *l*-Tupel gilt z.B. $a_i \in \mathbb{Z}_2 = \{0,1\}$ und $\Omega = \mathbb{Z}_2^l = \{0,1\}^l$

Population

Ein genetischer Algorithmus arbeitet auf einer Population
$P(t) = \{a_0(t), a_1(t), \ldots, a_{r-1}(t)\}$
bestehend aus $r > 1$ Individuen in der Generation t. Hierbei bezeichnet r die Populationsgröße.

Evolutionäre Algorithmen

Genetische Algorithmen

Selektion

Stochastische Selektion in Abhängigkeit der Fitness $f(a)$ eines Individuums a (z.B. Fitnessproportionale Selektion, *Linear Ranking*-Selektion, *Tournament*-Selektion, ...)

Crossover χ_Ω

Diskrete Rekombination der Allele zweier Individuen mit der Wahrscheinlichkeit χ

Mutation μ_Ω

Zufällige Änderung der Allele eines Indivduums mit der Wahrscheinlichkeit μ

Evolutionäre Algorithmen

Genetische Algorithmen

Exploration vs. *Exploitation*:

Suche nach besseren neuen Strukturen (*Exploration*) bei gleichzeitiger Ausnutzung guter bekannter Strukturen (*Exploitation*)

Crossover führt durch die diskrete Rekombination hinreichend verschiedener Eltern zu großen „Sprüngen“ im Suchraum (→ *Exploration*).

Mutation führt durch die wenigen zufälligen Änderungen in dem elterlichen l-Tupel zu einem ähnlichen Nachkommen in der Nähe des Elters (→ *Exploitation*).

Evolutionäre Algorithmen

Genetische Algorithmen

Pseudocode

$t = 0;$
Initialisiere Population $P(t) = \{a_0(t), a_1(t), \ldots, a_{r-1}(t)\};$
while end of adaptation $\neq$ true ***do***
    for $(k = 0; k \leq r-1; k{+}{+})$
        Selektiere Eltern a_{Vater} und a_{Mutter};
        Bilde Nachkomme $a_k(t+1) = \mu_\Omega(\chi_\Omega(a_{\text{Vater}}, a_{\text{Mutter}}))$
            mit Crossover χ_Ω und Mutation μ_Ω;
        Berechne Fitness von $a_k(t+1)$;
    end
    $t{+}{+};$
end

Evolutionäre Algorithmen

Genetische Algorithmen

Codierung

Die Codierung einer individuellen Struktur mit Hilfe eines (z.B. binären) *l*-Tupels $a=(a_0, a_1, \ldots, a_{l-1})$ ist abhängig von dem zu lösenden Adaptionsproblem.

Für einen reellen Parametervektor $\vec{x}=\left(x_0, x_1, \ldots, x_{N-1}\right)^T \in \mathbb{R}^N$ mit den Wertebereichen $x_{n,\min} \le x_n \le x_{n,\max}$ kann z.B. die folgende diskrete Repräsentation auf der Basis eines Binärcodes mit *b* Binärzeichen pro Parameter x_n verwendet werden.

$$x_n = x_{n,\min} + \frac{x_{n,\max} - x_{n,\min}}{2^b - 1} \cdot \sum_{i=0}^{b-1} a_{n \cdot b + i} \cdot 2^i$$

Es gilt $l = N \cdot b$.

Evolutionäre Algorithmen

Genetische Algorithmen

Fitnessproportionale Selektion

Gegeben sei eine Population $P(t)=\{a_0(t), a_1(t), \ldots, a_{r-1}(t)\}$ mit den individuellen Fitnesswerten $f(a_k(t)) \geq 0$.

Die Wahrscheinlichkeit, dass ein Individuum $a_k(t)$ in der Generation t als Elter selektiert wird, ist proportional zu der individuellen Fitness $f(a_k(t))$.

$$p_k(t) = \Pr\{\text{Individuum } a_k(t) \text{ wird selektiert}\} \sim f(a_k(t))$$

Normierung der Wahrscheinlichkeiten $\sum_{k=0}^{r-1} p_k(t) = 1$

$$\Rightarrow \quad p_k(t) = \frac{f(a_k(t))}{\sum_{j=0}^{r-1} f(a_j(t))}$$

Evolutionäre Algorithmen

Genetische Algorithmen

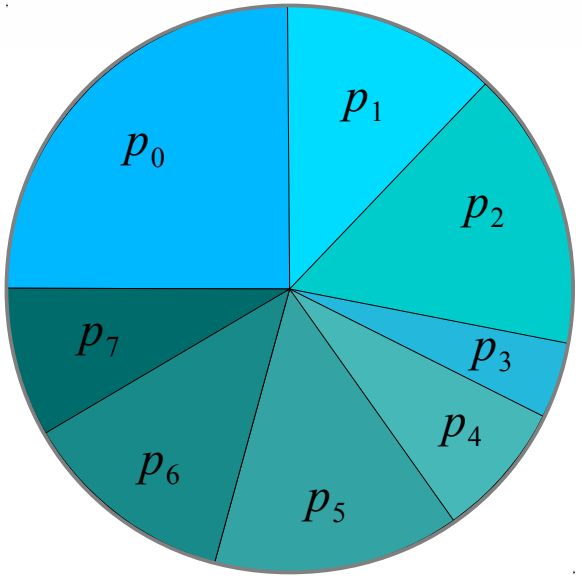
Fitnessproportionale Selektion

Die Selektionswahrscheinlichkeiten bei der fitness-proportionalen Selektion können mit Hilfe eines Rouletterads mit der Fläche 1 veranschaulicht werden (*Roulette Wheel*-Selektion).

Beispiel für $r = 8$

$$p_k = \frac{f(a_k)}{\sum_{j=0}^{7} f(a_j)}$$

mit der Normierung $\sum_{k=0}^{7} p_k = 1$



Evolutionäre Algorithmen

Genetische Algorithmen

Fitnessproportionale Selektion

Bei der fitnessproportionalen Selektion müssen die individuellen Fitnesswerte $f\left(a_k(t)\right) \geq 0$ nicht negativ sein.

Wird ferner vorausgesetzt, dass ein Individuum mit mittlerer Fitness $\overline{f(t)} = r^{-1} \sum_{k=0}^{r-1} f\left(a_k(t)\right)$ genau einmal und das schlechteste Individuum mit der Fitness $f_{\min}(t) = \min\left\{ f\left(a_k(t)\right) \;:\; 0 \leq k \leq r-1 \right\}$ genau keinmal im Mittel selektiert werden, so kann die folgende lineare Fitnessskalierung verwendet werden.

$$f_{\text{skaliert}}\left(a_k(t)\right) = \frac{f\left(a_k(t)\right) - f_{\min}(t)}{\overline{f(t)} - f_{\min}(t)} \quad \Rightarrow \quad p_k(t) \sim f_{\text{skaliert}}\left(a_k(t)\right)$$

Evolutionäre Algorithmen

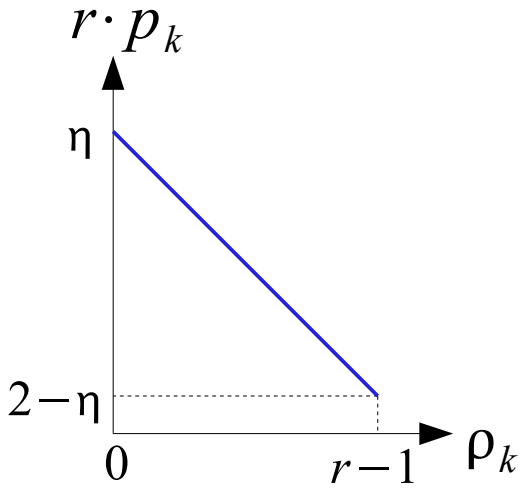
Genetische Algorithmen

Linear Ranking-Selektion

Bei der *Linear Ranking*-Selektion werden Individuen $a_k(t)$ in Abhängigkeit ihres Ranges $\rho_k(t)$ mit $0 \le \rho_k(t) \le r-1$ selektiert. Hierbei erhält das beste Individuum den Rang 0, das zweitbeste Individuum den Rang 1, ... und das schlechteste Individuum den Rang $r-1$.

⇒ Selektionswahrscheinlichkeiten $(1 < \eta < 2)$

$$p_k(t) = \frac{1}{r} \cdot \left(\eta - 2 \cdot [\eta - 1] \cdot \frac{\rho_k(t)}{r-1} \right)$$



Vorteil: keine Skalierung erforderlich

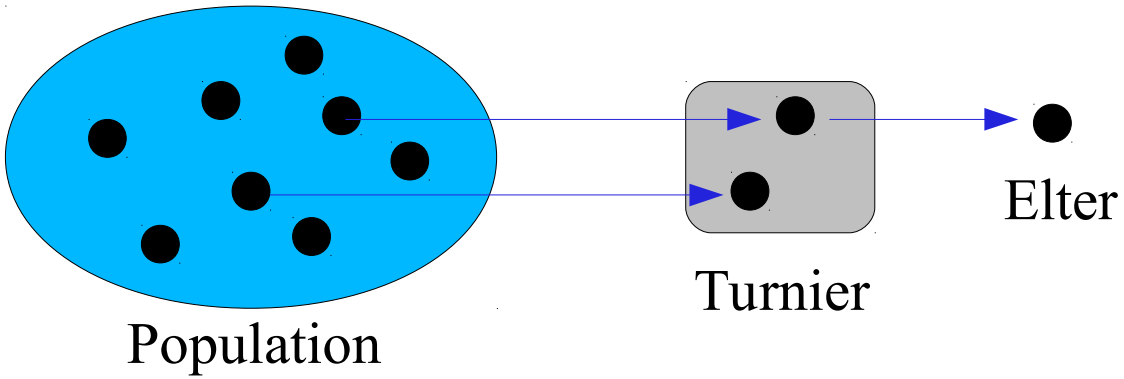
Nachteil: Sortierung der Population

Genetische Algorithmen

Tournament-Selektion

Bei der *Tournament*-Selektion wird zur Selektion eines Elters ein Turnier veranstaltet, indem $\rho \geq 2$ Individuen zufällig mit gleicher Wahrscheinlichkeit $1/r$ aus der Population gewählt werden und das beste dieser ρ Individuen als Elter selektiert wird.

Beispiel mit $\rho = 2$



Übung

Genetische Algorithmen

Implementieren Sie in MATLAB oder der Programmiersprache C geeignete Module für die fitnessproportionale Selektion, die *Linear Ranking*-Selektion sowie die *Tournament*-Selektion.

Analysieren Sie den Mittelwert und die Streuung der Nachkommenzahl eines Individuums mit einer gegebenen Selektionswahrscheinlichkeit.

Implementieren Sie in Matlab oder der Programmiersprache C für die fitnessproportionale Selektion die diskutierte lineare Fitnessskalierung.

Evolutionäre Algorithmen

Genetische Algorithmen

Implementierung der Selektion

Die erwartete Zahl $n_k(t)$ von Nachkommen eines Individuums $a_k(t)$ mit der Selektionswahrscheinlichkeit $p_k(t)$ beträgt

$$n_k(t) = r \cdot p_k(t)$$

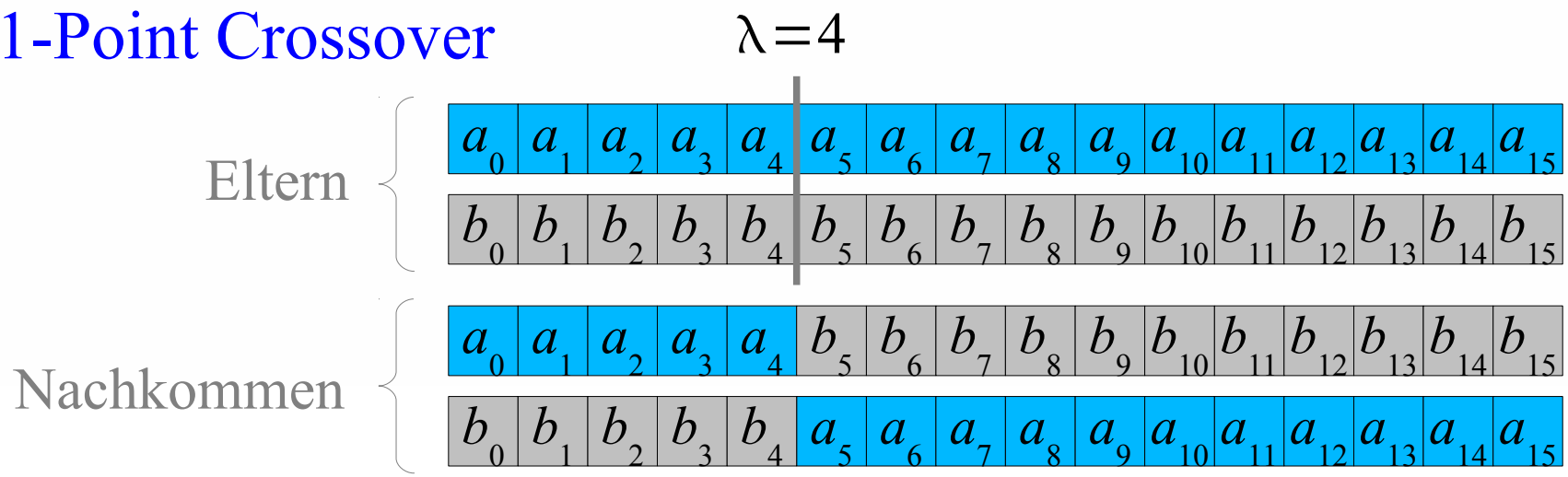
Stochastic Universal Sampling $\Rightarrow$ r Eltern $a_{i_0(t)}, a_{i_1(t)}, \ldots, a_{i_{r-1}(t)}$

Bilde die gleichverteilte reelle Zufallszahl $z \in [0,1]$;
for $(\zeta = 0, h = 0, j = 1;\ h < r-1;\ h{+}{+})$
  for $(\hat{n}_h = 0, \zeta += n_h;\ \zeta > z;\ z{+}{+})$
    $\hat{n}_h{+}{+};\ i_j = h;\ j{+}{+};$
  end
end
Permutiere Indextupel $(i_0, i_1, \ldots, i_{r-1})$;

Evolutionäre Algorithmen

Genetische Algorithmen

1-Point Crossover



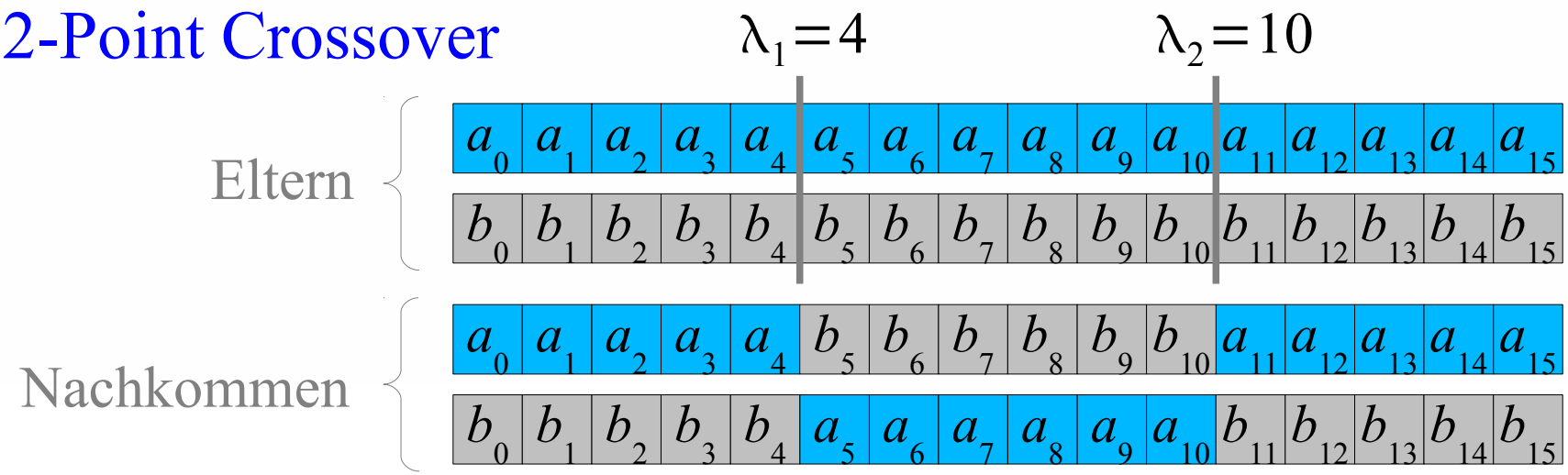
Als Nachkomme wird mit gleicher Wahrscheinlichkeit einer der beiden Nachkommen gewählt.

Der Crossover-Operator wird mit der Wahrscheinlichkeit χ angewendet. Der Schnittpunkt λ wird zufällig aus der Menge $\lambda \in \{0, 1, \ldots, l-2\}$ gewählt.

Evolutionäre Algorithmen

Genetische Algorithmen

2-Point Crossover



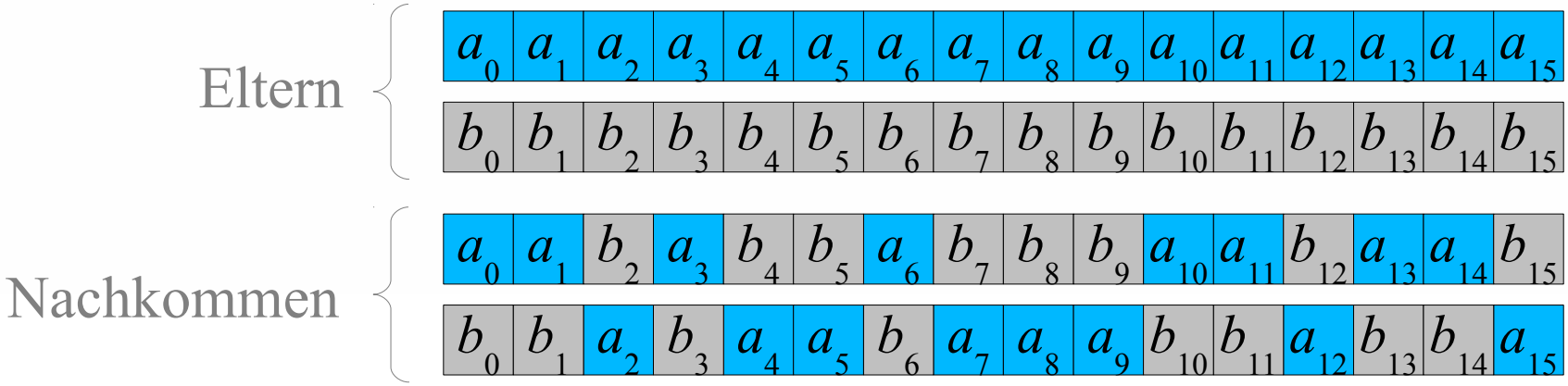
Als Nachkomme wird mit gleicher Wahrscheinlichkeit einer der beiden Nachkommen gewählt.

Der Crossover-Operator wird mit der Wahrscheinlichkeit χ angewendet. Die Schnittpunkte $\lambda_1 < \lambda_2$ werden zufällig aus der Menge $\lambda_1, \lambda_2 \in \{0, 1, \ldots, l-2\}$ gewählt.

Evolutionäre Algorithmen

Genetische Algorithmen

Uniform Crossover



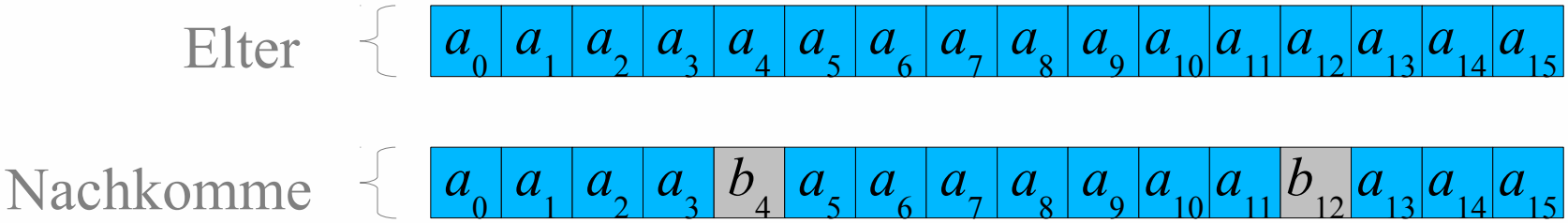
Als Nachkomme wird mit gleicher Wahrscheinlichkeit einer der beiden Nachkommen gewählt.

Der Crossover-Operator wird mit der Wahrscheinlichkeit χ angewendet. Die Allele eines Nachkommens stammen zufällig mit der Wahrscheinlichkeit ½ von einem der Eltern.

Evolutionäre Algorithmen

Genetische Algorithmen

Mutation



Jedes Allel wird mit der Wahrscheinlichkeit μ mutiert.

Die Zahl der mutierten Allele folgt einer Binomialverteilung.

$$\Pr\{\lambda \text{ Allele mutiert}\} = \binom{l}{\lambda} \cdot \mu^{\lambda} \cdot (1-\mu)^{l-\lambda}$$

Im Mittel werden $\mu \cdot l$ Allele mutiert.

Die Mutationswahrscheinlichkeit μ wird üblicherweise umgekehrt proportional zur Länge der individuellen l-Tupel gewählt, z.B. $\mu = l^{-1}$

Evolutionäre Algorithmen

Genetische Algorithmen

Realisierung einer Ereigniswahrscheinlichkeit

Das Ereignis E soll mit der Wahrscheinlichkeit q eintreten.

Zu diesem Zweck wird eine reelle gleichverteilte Zufallszahl z in dem Intervall [0,1] erzeugt.

Das Ereignis E tritt mit der Wahrscheinlichkeit q ein, wenn gilt: Ereignis E tritt ein, wenn $z \leq q$

Beweis:

Mit der Wahrscheinlichkeitsdichte $p(z) = \begin{cases} 1 & , \text{ für } 0 \leq z < 1 \\ 0 & , \text{ sonst} \end{cases}$ folgt

$$\Pr\{\text{Ereignis } E \text{ tritt ein}\} = \Pr\{z \leq q\} = \int_{-\infty}^{q} p(z)\,dz = \int_{0}^{q} 1\,dz = q$$

Evolutionäre Algorithmen

Genetische Algorithmen

SGA – Simple Genetic Algorithm

$t = 0$;
Initialisiere Population $P(t) = \{a_0(t), a_1(t), \ldots, a_{r-1}(t)\}$;
while end of adaptation $\neq$ true ***do***
    for $(k = 0; k \leq r-1; k{+}{+})$
        Selektiere Eltern a_{Vater} und a_{Mutter} mit
            fitnessproportionaler Selektion;
        Bilde Nachkomme $a_k(t+1) = \mu_\Omega(\chi_\Omega(a_{\text{Vater}}, a_{\text{Mutter}}))$
            mit 1-Point Crossover χ_Ω und Mutation μ_Ω;
        Berechne Fitness von $a_k(t+1)$;
    end
    $t{+}{+}$;
end

Evolutionäre Algorithmen

Genetische Algorithmen

Einführendes Beispiel

SGA mit fitnessproportionaler Selektion und Populationsgröße $r = 4$ zur Optimierung der Funktion

$$f(x) = x^2$$

in dem Intervall

$$0 \leq x \leq 31$$

Binäre l-Tupel $a = (a_0, a_1, a_2, a_3, a_4)$ mit $l = 5$ und der Codierung

$$x = a_0 \cdot 2^4 + a_1 \cdot 2^3 + a_2 \cdot 2^2 + a_3 \cdot 2^1 + a_4 \cdot 2^0$$
$$= a_0 \cdot 16 + a_1 \cdot 8 + a_2 \cdot 4 + a_3 \cdot 2 + a_4$$

Quelle: D.E. Goldberg: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989

Evolutionäre Algorithmen

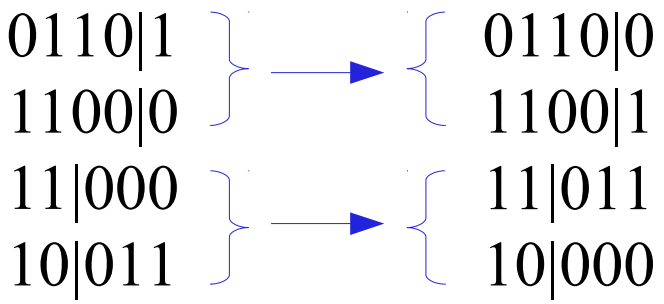
Genetische Algorithmen

Einführendes Beispiel (Fortsetzung)

Generation $t = 0$

k	a_k	x	$f(a_k)$	p_k	n_k	$\hat{n}_k$
0	01101	13	169	0,14	0,56	1
1	11000	24	576	0,49	1,96	2
2	01000	8	64	0,06	0,24	0
3	10011	19	361	0,31	1,24	1
Mittelwert			293			
Maximum			576			

Crossover (keine Mutation)



Hinweis: In diesem Beispiel werden beide erzeugten Nachkommen in die nächste Generation übernommen,

Evolutionäre Algorithmen

Genetische Algorithmen

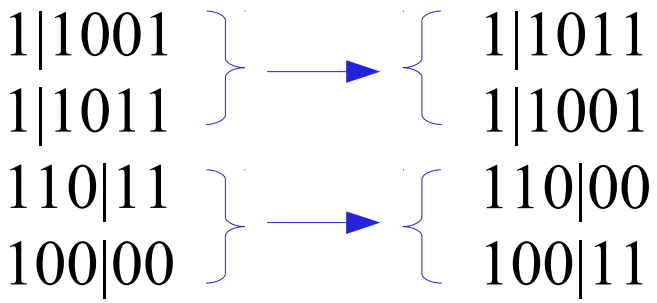
Einführendes Beispiel (Fortsetzung)

Generation $t = 1$

k	a_k	x	$f(a_k)$	p_k	n_k	$\hat{n}_k$
0	01100	12	144	0,08	0,33	0
1	11001	25	625	0,31	1,22	1
2	11011	27	729	0,34	1,35	2
3	10000	16	256	0,18	0,72	1

Mittelwert	439
Maximum	729

Crossover (keine Mutation)



Genetische Algorithmen

Populationsdiversität

Mit zunehmender Anzahl der Generationen werden sich die individuellen l-Tupel der Population $P(t)$ ähnlicher.

Mittlerer relativer Hamming-Abstand

$$d_P(t) = \frac{2}{r \cdot (r-1)} \cdot \sum_{k=0}^{r-1} \sum_{k'=0}^{k-1} \frac{1}{l} \sum_{i=0}^{l-1} \left[a_{k,i}(t) \neq a_{k',i}(t) \right]$$

mit der Indikatorfunktion

$$[a=b] = \delta_{a,b} = \begin{cases} 1 & , \quad a=b \\ 0 & , \quad a \neq b \end{cases}$$

Es gilt $0 \leq d_P(t) < \frac{1}{2}$

Übung

Genetische Algorithmen

Implementieren Sie einen genetischen Algorithmus mit fitnessproportionaler Selektion, 1-Point Crossover und Mutation zur Adaption binärer *l*-Tupel in MATLAB oder der Programmiersprache C.

Minimieren Sie die so genannte Rastrigin-Funktion

$$f(\vec{x}) = 10 \cdot N + \sum_{n=0}^{N-1} \left[x_n^2 - 10 \cdot \cos\left(2\pi\, x_n\right) \right]$$

in dem Intervall $-5{,}12 \le x_n \le 5{,}12$ mit einer Auflösung von $\Delta x_n = 0{,}01$ für $N = 2, 5, 10, 20, 50, 100$.

Hinweis: In der bisherigen Formulierung wurde die Maximierung der Fitnessfunktion vorausgesetzt.

Evolutionäre Algorithmen

Genetische Algorithmen

Varianten der Selektion

Elitismus

Aufgrund der stochastischen Selektion kann das bisher beste Individuum aussterben. Bei der Selektion mit Elitismus wird das beste Individuum einer Population unverändert in die nächste Generation kopiert.

Nischenbildung

Im Verlauf der Generationen werden sich die Individuen in einer Population ähnlicher, d.h. die Population ist auf einen Teil des Suchraums konzentriert. Hierdurch verringert sich die Populationsdiversität und somit die Exploration des Suchraums.

Evolutionäre Algorithmen

Genetische Algorithmen

Nischenbildung durch *Crowding*

Anstelle der vollständigen Erzeugung einer neuen Population in der nächsten Generation (*generational* GA) wird bei einem *steady state* GA ein Nachkomme erzeugt, der in der existierenden Population ein Individuum ersetzt.

Bei der Nischenbildung durch *Crowding* ersetzt der gebildete Nachkomme das diesem Nachkommen ähnlichste Individuum in der Population.

Die Ähnlichkeit kann z.B. mit Hilfe des relativen Hamming-Abstands ermittelt werden.

$$d_H(a,b) = \frac{1}{l}\sum_{i=0}^{l-1}\left[a_i \neq b_i\right] \qquad \text{mit} \quad 0 \leq d_H(a,b) \leq 1$$

Evolutionäre Algorithmen

Genetische Algorithmen

Nischenbildung durch *Sharing*

Aufgrund der von der individuellen Fitness abhängigen Selektionswahrscheinlichkeit wächst die mittlere Anzahl der Nachkommen eines Individuums mit der Fitness.

Bei der Nischenbildung durch *Sharing* wird die Fitness eines Individuums in Abhängigkeit der diesem Individuum ähnlichen Individuen reduziert. In der Biologie entspricht dies der gemeinsamen Ressourcen-Nutzung in einer Nische.

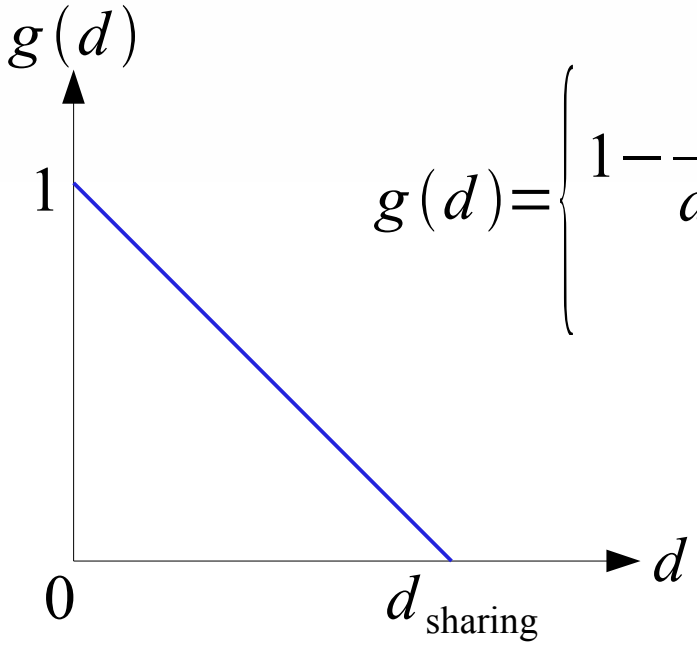
$$f_{\text{sharing}}\left(a_k(t)\right) = \frac{f\left(a_k(t)\right)}{\sum_{j=0}^{r-1} g\left(d_H\left(a_j(t), a_k(t)\right)\right)} \quad \Rightarrow \quad p_k(t) \sim f_{\text{sharing}}\left(a_k(t)\right)$$

Evolutionäre Algorithmen

Genetische Algorithmen

Nischenbildung durch *Sharing*

Beispiel einer *Sharing*-Funktion mit $0 \leq d_{\text{sharing}} \leq 1$



$$g(d) = \begin{cases} 1 - \dfrac{d}{d_{\text{sharing}}} & , \quad \text{falls } 0 \leq d \leq d_{\text{sharing}} \\ 0 & , \quad \text{sonst} \end{cases}$$

Evolutionäre Algorithmen

Genetische Algorithmen

Diskrete Optimierungsprobleme

Bei diskreten Optimierungsproblemen ist häufig die Reihenfolge von Komponenten entscheidend (z.B. Sortierung, Rundreise, Maschinenbelegung, ...).

Die individuellen Strukturen werden als Permutationen codiert, d.h. für eine Menge von l Komponenten stellt ein individuelles l-Tupel eine Liste dieser l Komponenten in einer bestimmten Reihenfolge dar, z.B. (2,4,6,0,5,1,7,3).

Auf diese Optimierungsprobleme sind die bisherigen genetischen Operatoren nicht direkt anwendbar.

⇒ Spezialisierte genetische Operatoren

Genetische Algorithmen

TSP – Travelling Salesman Problem

Gegeben ist eine Menge von l Städten. Gesucht wird eine vollständige Rundreise, die jede Stadt genau einmal besucht, mit minimaler Länge.

Codierung der Rundreise durch l Städte durch eine Permutation des l-Tupels $(0,1,2,...,l-1)$.

Die Größe des Suchraums ist $l!$

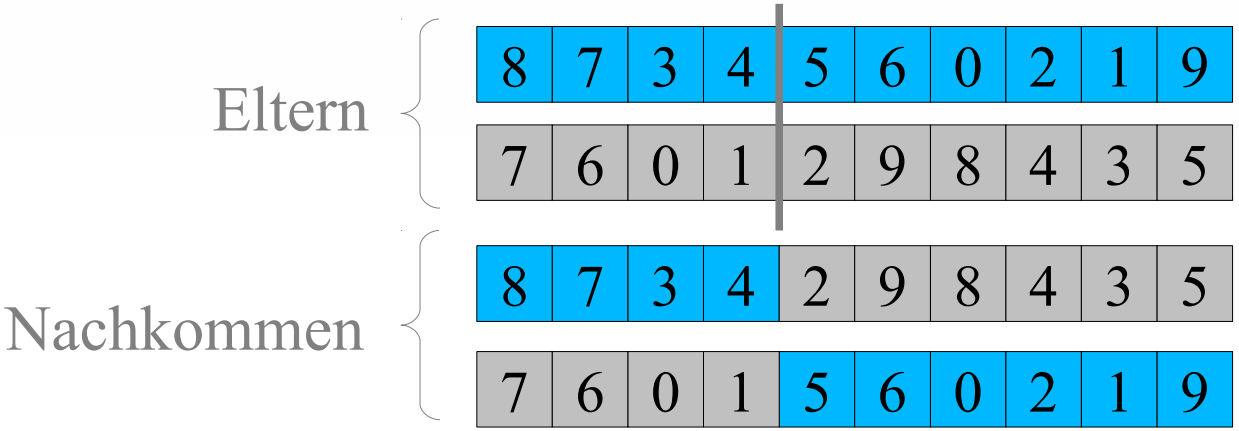
⇒ Der Suchraum ist riesig!
Für $l = 30$ beträgt die Größe des Suchraums $30! \approx 10^{32}$.

Evolutionäre Algorithmen

Genetische Algorithmen

TSP – Travelling Salesman Problem

Die bisherigen Crossover-Operatoren führen zu nicht erlaubten Codierungen, z.B.



Idee:

Austausch von relativen Ordnungen zwischen den Eltern.

Evolutionäre Algorithmen

Genetische Algorithmen

TSP – Travelling Salesman Problem

Idee: Order Crossover für Permutationen

Wähle zufällig einen Teil des l-Tupels des ersten Elters.

Kopiere diesen Teil in den Nachkommen.

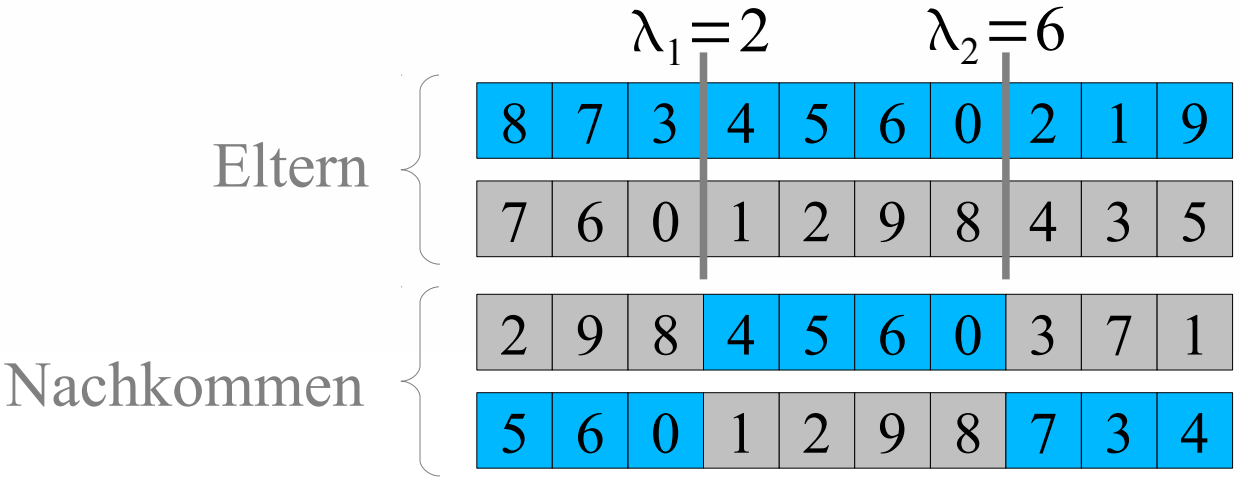
Kopiere die Allele, welche nicht Teil des Abschnitts sind, von dem zweiten Elter in den Nachkommen.

Wird das Ende des l-Tupels des zweiten Elters erreicht, so wird zyklisch zum Anfang des l-Tupels des zweiten Elters fortgeschritten.

Evolutionäre Algorithmen

Genetische Algorithmen

Order Crossover für Permutationen



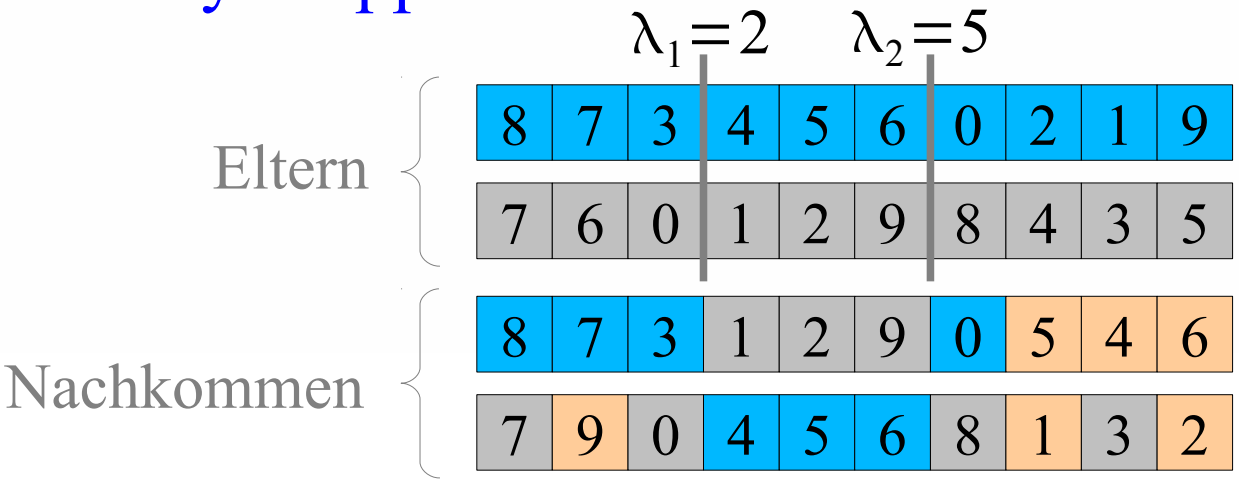
Zwei Schnittpunkte $\lambda_1 < \lambda_2$ werden zufällig aus der Menge $\lambda_1, \lambda_2 \in \{0, 1, \ldots, l-2\}$ gewählt.

Als Nachkomme wird mit gleicher Wahrscheinlichkeit einer der beiden Nachkommen gewählt.

Evolutionäre Algorithmen

Genetische Algorithmen

Partially Mapped Crossover für Permutationen



Zwei Schnittpunkte $\lambda_1 < \lambda_2$ werden zufällig aus der Menge $\lambda_1, \lambda_2 \in \{0, 1, \ldots, l-2\}$ gewählt.

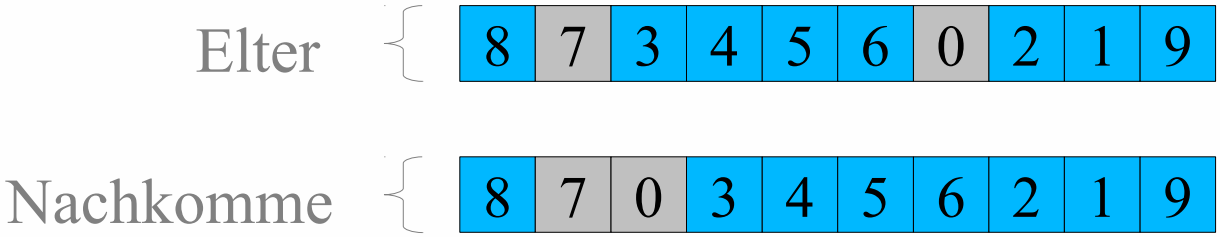
Die innerhalb des Kreuzungsbereiches auftretenden Paare geben die auszuführenden Vertauschungen an.

Als Nachkomme wird mit gleicher Wahrscheinlichkeit einer der beiden Nachkommen gewählt.

Evolutionäre Algorithmen

Genetische Algorithmen

Shift Mutation für Permutationen



Zwei Allele des *l*-Tupels des Elters werden zufällig gewählt.

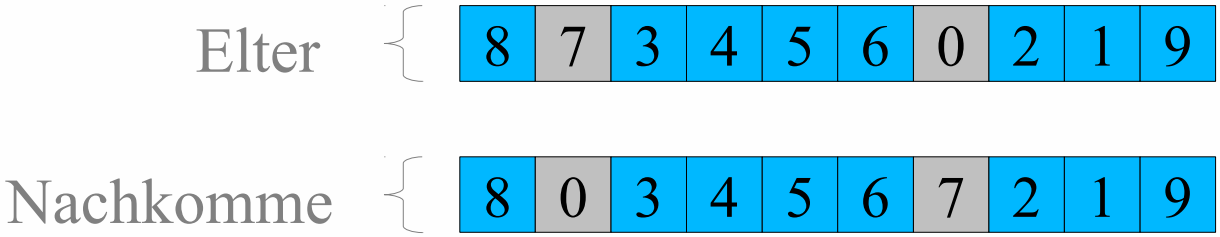
Das zweite Allel wird zu dem ersten Allel verschoben.

Die restlichen Allele werden in ihrer ursprünglichen Reihenfolge hinzugefügt.

Evolutionäre Algorithmen

Genetische Algorithmen

Swap Mutation für Permutationen



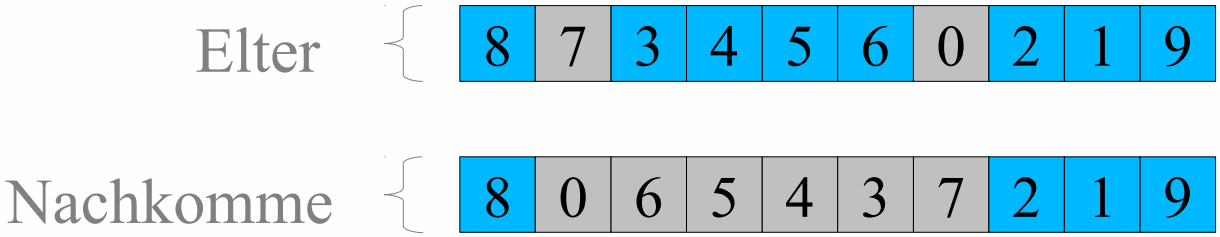
Zwei Allele des l-Tupels des Elters werden zufällig gewählt.

Die beiden Allele werden vertauscht.

Evolutionäre Algorithmen

Genetische Algorithmen

Inversion Mutation für Permutationen



Zwei Allele des l-Tupels des Elters werden zufällig gewählt.

Der Teil des l-Tupels zwischen den beiden Allelen wird invertiert.

Übung

Genetische Algorithmen

Implementieren Sie einen genetischen Algorithmus zur Lösung des TSP – Travelling Salesman Problem in MATLAB oder der Programmiersprache C.

Optimieren Sie die Rundreise für $l = 30$ Städte.

Erzeugen Sie zu diesem Zweck in dem Quadrat $[0,1]^2$ zufällig die zweidimensionalen Positionen der Städte.

Die Entfernung einer Rundreise ergibt sich aus der Summe der Euklidschen Abstände

$$\sqrt{\left(x_{\text{Münster}} - x_{\text{Düsseldorf}}\right)^2 + \left(y_{\text{Münster}} - y_{\text{Düsseldorf}}\right)^2}$$

der in der Rundreise benachbarten Städte „Münster“ und „Düsseldorf“.

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

SGA – Simple Genetic Algorithm

$t = 0;$
Initialisiere Population $P(t) = \{a_0(t), a_1(t), \ldots, a_{r-1}(t)\};$
while end of adaptation $\neq$ true ***do***
    for $(k = 0; k \leq r-1; k{+}{+})$
        Selektiere Eltern a_{Vater} und a_{Mutter} mit
            fitnessproportionaler Selektion;
        Bilde Nachkomme $a_k(t+1) = \mu_\Omega(\chi_\Omega(a_{\text{Vater}}, a_{\text{Mutter}}))$
            mit 1-Point Crossover χ_Ω und Mutation μ_Ω;
        Berechne Fitness von $a_k(t+1)$;
    end
    $t{+}{+};$
end

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Schema

$$H=(H_0, H_1, \ldots, H_{l-1})$$

Ein (binäres) l-Tupel $a(t)=(a_0, a_1, \ldots, a_{l-1})$ repräsentiert ein Schema $H=(H_0, H_1, \ldots, H_{l-1})$ mit $H_i \in \mathbb{Z}_2 \cup \{\#\}$, d.h.

$$a(t)=(a_0(t), a_1(t), \ldots, a_{l-1}(t)) \in H=(H_0, H_1, \ldots, H_{l-1})$$

wenn

$$a_i(t)=\begin{cases} H_i & , \text{ falls } H_i \neq \# \\ \text{beliebig} & , \text{ falls } H_i = \# \end{cases}$$

(# bedeutet ein *Don´t Care*)

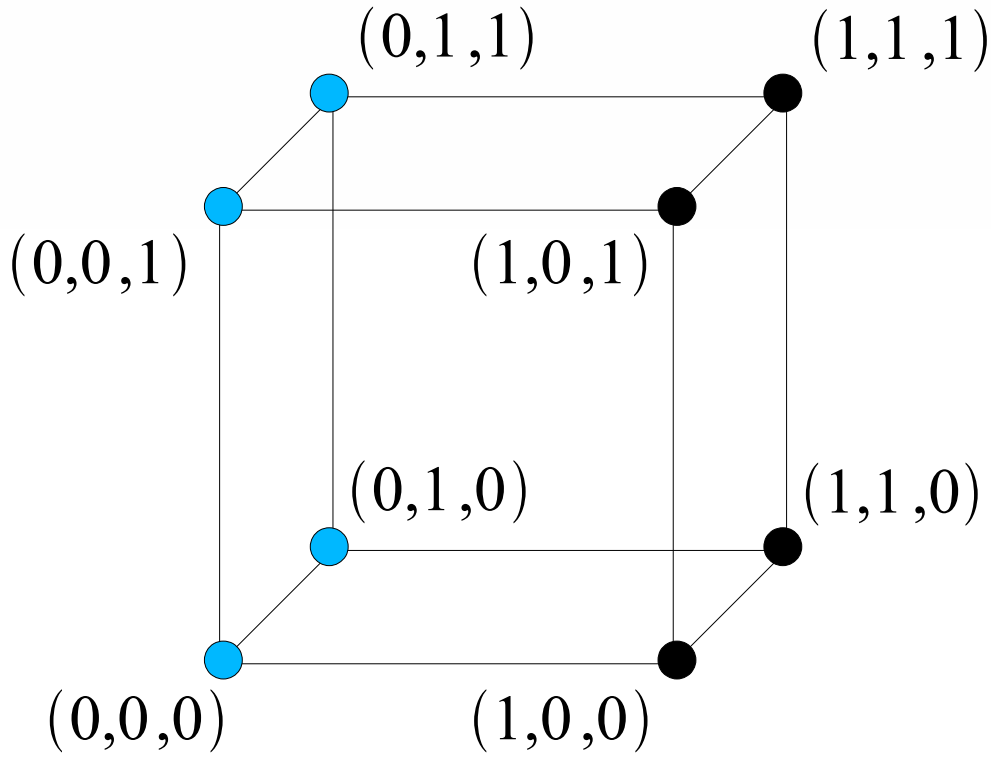
Ein Schema entspricht einer Hyperebene in A.

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Schema entspricht Hyperebene $H = (H_0, H_1, \ldots, H_{l-1})$

Beispiel für $l = 3$



$H = (0, \#, \#)$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Die mittlere Fitness einer Menge von r individuellen l-Tupeln $a_k(t)=(a_{k,0}(t), a_{k,1}(t), \ldots, a_{k,l-1}(t))$ lautet

$$\overline{f(t)}=\frac{1}{r}\sum_{k=0}^{r-1} f\left(a_k(t)\right)$$

Die mittlere Fitness der $r_H(t)$ l-Tupeln, die das Schema $H=(H_0, H_1, \ldots, H_{l-1})$ repräsentieren, lautet

$$\overline{f_H(t)}=\frac{1}{r_H(t)}\sum_{k=0, a_k(t)\in H}^{r-1} f\left(a_k(t)\right)$$

Ziel: Finde Kombinationen von Komponenten bzw. Schemata mit $\overline{f_H(t)} > \overline{f(t)}$, die zu Strukturen hoher Güte führen.

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Einführendes Beispiel (Fortsetzung)

SGA mit fitnessproportionaler Selektion und Populationsgröße $r = 4$ zur Optimierung der Funktion

$$f(x) = x^2$$

in dem Intervall

$$0 \leq x \leq 31$$

Binäre l-Tupel $a = (a_0, a_1, a_2, a_3, a_4)$ mit $l = 5$ und der Codierung

$$x = a_0 \cdot 2^4 + a_1 \cdot 2^3 + a_2 \cdot 2^2 + a_3 \cdot 2^1 + a_4 \cdot 2^0$$
$$= a_0 \cdot 16 + a_1 \cdot 8 + a_2 \cdot 4 + a_3 \cdot 2 + a_4$$

Quelle: D.E. Goldberg: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989

Theorie genetischer Algorithmen

Einführendes Beispiel (Fortsetzung)

Generation $t = 0$

k	a_k	x	$f(a_k)$	p_k	n_k	$\hat{n}_k$
0	01101	13	169	0,14	0,58	1
1	11000	24	576	0,49	1,97	2
2	01000	8	64	0,06	0,22	0
3	10011	19	361	0,31	1,23	1

Mittelwert	293
Maximum	576

H	$a_k \in H$	$\overline{f}_H$
1####	1 und 3	469
#10##	1 und 2	320
1###0	1	576

Theorie genetischer Algorithmen

Einführendes Beispiel (Fortsetzung)

Generation $t = 1$

k	a_k	x	$f(a_k)$	p_k	n_k	$\hat{n}_k$
0	01100	12	144	0,08	0,33	0
1	11001	25	625	0,31	1,22	1
2	11011	27	729	0,34	1,35	2
3	10000	16	256	0,18	0,72	1

Mittelwert	439
Maximum	729

H	$a_k \in H$	$\overline{f_H}$
1####	1, 2 und 3	537
#10##	1 und 2	677
1###0	3	256

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Ordnung o_H des Schemas H

Anzahl der definierenden Positionen eines Schemas H mit $H_i \neq \#$

Beispiel:

Schema $H = \#0\#\#1\#1\#$ mit der Ordnung $o_H = 3$

Länge l_H des Schemas H

Maximaler Abstand der definierenden Positionen eines Schemas H

Beispiel:

Schema $H = \#0\#\#1\#1\#$ mit der Länge $l_H = 5$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Schema-Theorem

Genetischer Algorithmus SGA mit fitnessproportionaler Selektion, 1-Point Crossover mit Crossoverwahrscheinlichkeit χ und Mutation mit Mutationswahrscheinlichkeit μ

Die erwartete Zahl $r_H(t)$ der das Schema H mit der Länge l_H und der Ordnung o_H repräsentierenden Individuen ändert sich in einer Generation zu

$$E\left\{r_H(t+1)\right\} \geq r_H(t) \cdot \frac{\overline{f_H(t)}}{\overline{f(t)}} \cdot \left(1 - \chi \cdot \frac{l_H}{l-1} \cdot \left[1 - \frac{r_H(t)}{r} \cdot \frac{\overline{f_H(t)}}{\overline{f(t)}}\right]\right) \cdot \left(1-\mu\right)^{o_H}$$

Quelle: A. Neubauer: *Adaptive Filter auf der Basis genetischer Algorithmen*, VDI-Verlag, 1997

Theorie genetischer Algorithmen

Schema-Theorem – Beweis

Wahrscheinlichkeit für die Selektion eines individuellen l-Tupels $a_k(t)$ bei der fitnessproportionalen Selektion

$$\Pr\left\{\text{Individuum } a_k(t) \text{ selektiert}\right\} = p_k(t) = \frac{f\left(a_k(t)\right)}{\sum_{j=0}^{r-1} f\left(a_j(t)\right)}$$

Das selektierte individuelle l-Tupel repräsentiert das Schema H mit der Wahrscheinlichkeit

$$\Pr\left\{\text{Schema } H \text{ selektiert}\right\} = \sum_{k=0,\, a_k(t)\in H}^{r-1} p_k(t) = \frac{\sum_{k=0,\, a_k(t)\in H}^{r-1} f\left(a_k(t)\right)}{\sum_{j=0}^{r-1} f\left(a_j(t)\right)}$$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Schema-Theorem – Beweis

Mit der mittleren Fitness des Schemas H

$$\overline{f_H(t)} = \frac{1}{r_H(t)} \sum_{k=0, a_k(t) \in H}^{r-1} f\left(a_k(t)\right)$$

und der mittleren Fitness der Population

$$\overline{f(t)} = \frac{1}{r} \sum_{k=0}^{r-1} f\left(a_k(t)\right)$$

folgt

$$\Pr\left\{\text{Schema } H \text{ selektiert}\right\} = \frac{r_H(t)}{r} \cdot \frac{\overline{f_H(t)}}{\overline{f(t)}}$$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Schema-Theorem – Beweis

Die Wahrscheinlichkeit, dass ein Schema *H* durch 1-Point Crossover zerstört wird, ist nach unten beschränkt durch das Produkt aus der Crossoverwahrscheinlichkeit χ und der Wahrscheinlichkeit $l_H/(l-1)$, dass der zufällig gewählte Schnittpunkt λ innerhalb der definierenden Positionen des Schemas liegt.

$\Rightarrow \quad \chi \cdot \frac{l_H}{l-1}$

Hinweis: Hierbei wird angenommen, dass nur ein Elter das Schema *H* repräsentiert.

Theorie genetischer Algorithmen

Schema-Theorem – Beweis

Wird ein das Schema H repräsentierendes l-Tupel mit einem zweiten selektierten l-Tupel mittels 1-Point Crossover mit der Wahrscheinlichkeit χ gekreuzt, so ist die Wahrscheinlichkeit dafür, dass das resultierende l-Tupel das Schema H repräsentiert, nach unten beschränkt durch

$$1-\chi\cdot\frac{l_H}{l-1}\cdot\left[1-\Pr\left\{\text{Schema } H \text{ selektiert}\right\}\right]$$

$$=1-\chi\cdot\frac{l_H}{l-1}\cdot\left[1-\frac{r_H(t)}{r}\cdot\frac{\overline{f_H(t)}}{\overline{f(t)}}\right]$$

Theorie genetischer Algorithmen

Schema-Theorem – Beweis

Wird ein das Schema *H* repräsentierendes *l*-Tupel mittels Mutation mit der Mutationswahrscheinlichkeit μ mutiert, so ist die Wahrscheinlichkeit dafür, dass das resultierende *l*-Tupel das Schema *H* repräsentiert, nach unten beschränkt durch

$$\left(1-\mu\right)^{o_H}$$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Schema-Theorem – Beweis

$$\Rightarrow \quad E\left\{\frac{r_H(t+1)}{r}\right\} \geq \Pr\{\text{Schema } H \text{ selektiert}\} \cdot \left(1-\chi\cdot\frac{l_H}{l-1}\cdot\left[1-\Pr\{\text{Schema } H \text{ selektiert}\}\right]\right)\cdot(1-\mu)^{o_H}$$

$$\Rightarrow \quad E\{r_H(t+1)\} \geq r_H(t)\cdot\frac{\overline{f_H(t)}}{\overline{f(t)}}\cdot\left(1-\chi\cdot\frac{l_H}{l-1}\cdot\left[1-\frac{r_H(t)}{r}\cdot\frac{\overline{f_H(t)}}{\overline{f(t)}}\right]\right)\cdot(1-\mu)^{o_H}$$

q.e.d.

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Schema-Theorem

Mit der Ungleichung

$$0 \leq \Pr\{\text{Schema } H \text{ selektiert}\} = \frac{r_H(t)}{r} \cdot \frac{\overline{f_H(t)}}{\overline{f(t)}} \leq 1$$

folgt die vereinfachte Ungleichung

$$E\{r_H(t+1)\} \geq r_H(t) \cdot \frac{\overline{f_H(t)}}{\overline{f(t)}} \cdot \left(1 - \chi \cdot \frac{l_H}{l-1}\right) \cdot (1-\mu)^{o_H}$$

Mit der Ungleichung von Bernoulli $(1-\mu)^{o_H} \geq 1 - \mu \cdot o_H$

folgt ferner

$$E\{r_H(t+1)\} \geq r_H(t) \cdot \frac{\overline{f_H(t)}}{\overline{f(t)}} \cdot \left(1 - \chi \cdot \frac{l_H}{l-1} - \mu \cdot o_H\right)$$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Building Block-Hypothese

Mit dem Verlustfaktor

$$\varepsilon_H = \chi \cdot \frac{l_H}{l-1} + \mu \cdot o_H$$

folgt für das Schema-Theorem

$$E\left\{ r_H(t+1) \right\} \geq r_H(t) \cdot \frac{\overline{f_H(t)}}{\overline{f(t)}} \cdot \left(1 - \varepsilon_H \right)$$

In der Population repräsentierte kurze Schemata H geringer Ordnung mit überdurchschnittlicher Fitness $\overline{f_H(t)} > \overline{f(t)}$ (*Building Blocks*) werden in der nächsten Generation von einer größeren erwarteten Zahl von Individuen repräsentiert (*Building Block*-Hypothese).

Übung

Genetische Algorithmen

Analysieren Sie für einen genetischen Algorithmus mit fitnessproportionaler Selektion, 1-Point Crossover und Mutation zur Adaption binärer *l*-Tupel das Schema-Theorem.

Simulieren Sie den genetischen Algorithmus zu diesem Zweck in MATLAB oder der Programmiersprache C und stellen Sie den Anteil der Schemata 0#0#...#, 0#1#...#, 1#0#...# und 1#1#...# grafisch über den Generationen dar.

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Das Schema-Theorem stellt die ursprünglich von J.H. Holland vorgeschlagene Beschreibung der Funktionsweise eines genetischen Algorithmus dar.

Kritik

- Analyse ausschließlich der destruktiven Wirkung genetischer Operatoren
- Geltung des Schema-Theorems nur für eine Generation
- Fitness eines Schemas abhängig von der aktuellen Population
- Schemata nicht unabhängig voneinander
- Begrenzte Aussagekraft hinsichtlich der Konvergenz bei Optimierungsproblemen

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Alternative theoretische Beschreibungen

Markoff-Ketten

Der genetische Algorithmus wird als Markoff-Prozess modelliert (→ Konvergenzverhalten, ...).

⇒ Dynamische Systeme

Der genetische Algorithmus wird als dynamisches System modelliert (→ Trajektorien, Fixpunkte, ...).

Statistische Mechanik-Approximation

Das Verhalten des genetischen Algorithmus wird ähnlich wie in der statistischen Mechanik durch geeignet definierte Mittelwerte approximiert.

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Jedes Individuum wird als binäres l-Tupel der Länge l aufgefasst und als Binärzahl $0 \le a \le 2^l - 1$ geschrieben.

Suchraum $\Omega = \mathbb{Z}_2^l = \{0,1\}^l$ der Kardinalität $|\Omega| = n = 2^l$

Werden die Individuen als Binärzahlen geschrieben, so gilt $\Omega = \{0, 1, \ldots, n-1\}$.

Elementweise Binäroperationen

modulo-2 Addition $a \oplus b$

Beispiel: $3 \oplus 5 = 011 \oplus 101 = 110 = 6$

modulo-2 Multiplikation $a \otimes b$

Beispiel: $6 \otimes 2 = 110 \otimes 010 = 010 = 2$

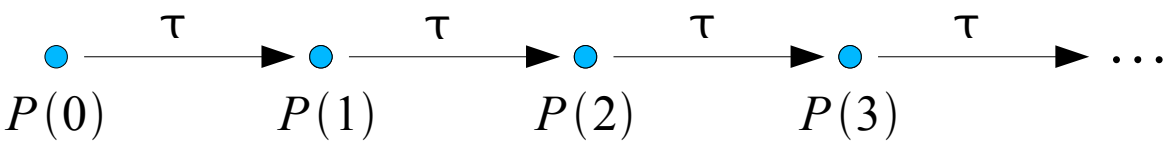
Komplement $\bar{a}$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Ein adaptives System auf Basis eines genetischen Algorithmus arbeitet auf einer Population mit r Individuen $P(t)=\{a_0(t), a_1(t), \ldots, a_{r-1}(t)\}$ unter Verwendung der individuellen Fitness $f\left(a_k(t)\right)$.

Algorithmus $P(t+1)=\tau\left(P(t)\right)$ it $\tau : \Omega^r \to \Omega^r$



Bemerkung:

Die Umgebungsinformation in Form des Fitness-Tupels $i_e(t)=\left\{f\left(a_0(t)\right), f\left(a_1(t)\right), \ldots, f\left(a_{r-1}(t)\right)\right\}$ wird der Übersichtlichkeit halber weggelassen.

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Da ein genetischer Algorithmus einen stochastischen Algorithmus darstellt, wird der Populationsvektor $\vec{p}(t)=\left(p_0(t), p_1(t), \ldots, p_{n-1}(t)\right)^T$ definiert.

Die Komponente $p_i(t)$ gibt den relativen Anteil des Individuums $i \in \Omega$ in der Population $P(t)$ an.
Dies entspricht der Wahrscheinlichkeit, mit der das Individuum $i \in \Omega$ in der Population $P(t)$ auftritt.

Schreibweise:
Mit der Indikatorfunktion $[i=j]=\delta_{i,j}=\begin{cases}1 & , \quad i=j \\ 0 & , \quad sonst\end{cases}$ gilt

$$p_i(t)=\frac{1}{r}\sum_{j \in P(t)}[j=i]$$

Evolutionäre Algorithmen

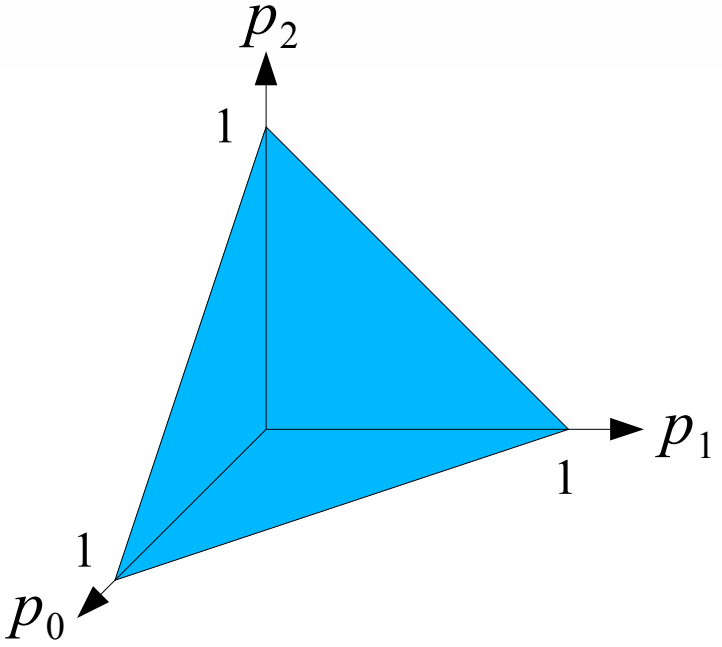
Theorie genetischer Algorithmen

Simplex

$$\Lambda = \left\{ \vec{p} = \left(p_0, p_1, \ldots, p_{n-1} \right) \;:\; p_i \geq 0 \;\wedge\; \sum_{i=0}^{n-1} p_i = 1 \right\}$$

Ein Populationsvektor $\vec{p}$ entspricht einem Wahrscheinlichkeitsvektor.

Beispiel für $n = 3$

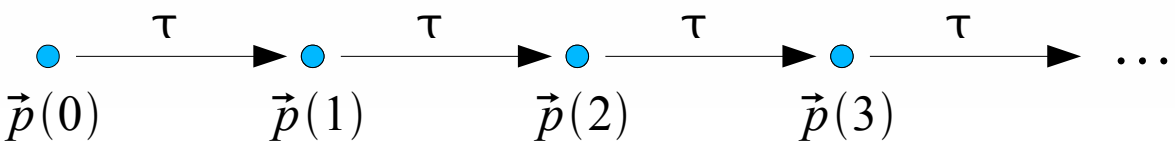


Evolutionäre Algorithmen

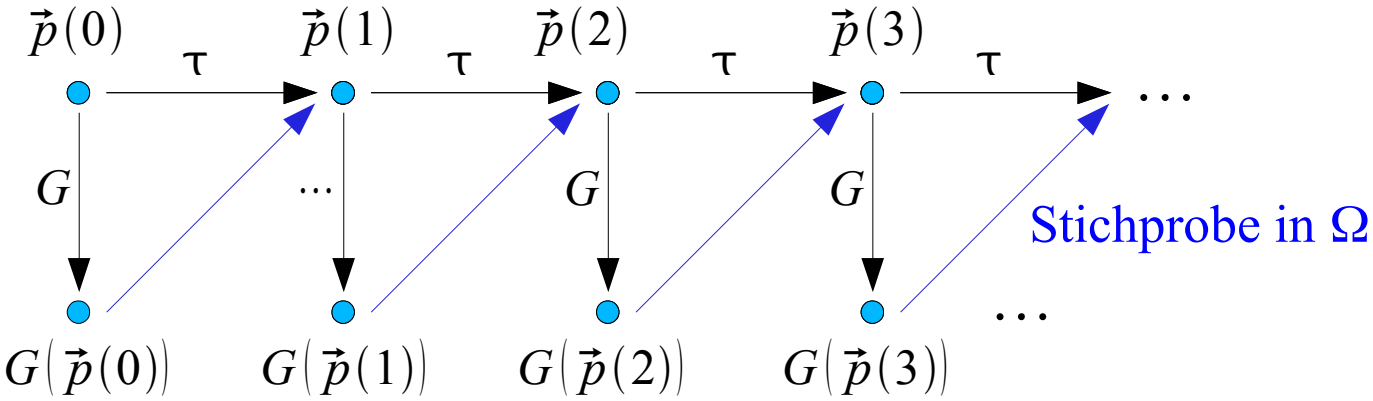
Theorie genetischer Algorithmen

Der stochastische Adaptionsalgorithmus wird auf Basis des Populationsvektors $\vec{p}$ definiert.

Algorithmus $\vec{p}(t+1)=\tau(\vec{p}(t))$ mit $\tau:\Lambda\to\Lambda$



Algorithmus mit heuristischer Funktion $G:\Lambda\to\Lambda$



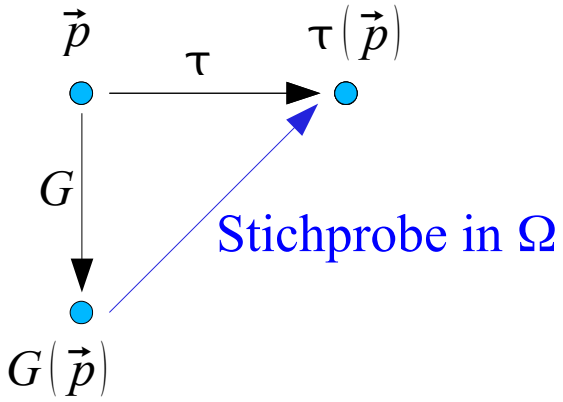
Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Heuristik

Die heuristische Funktion $G(\cdot)$ erzeugt auf Basis des Populationsvektors $\vec{p}$ den Wahrscheinlichkeitsvektor $G(\vec{p})$, entsprechend dem eine zufällige Stichprobe $\tau(\vec{p})$ in dem Suchraum Ω genommen wird.

Generation



Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Heuristik

$$G(\vec{p})_i = \Pr\{\text{Individuum } i \text{ ist in der Stichprobe von } \Omega\}$$

Mittelwert

$$E\{\tau(\vec{p})\} = G(\vec{p})$$

⇒ Im Mittel wird $G(\vec{p})$ als neuer Populationsvektor erhalten.

Varianz (= mittlere quadratische Abweichung)

$$E\{\|\tau(\vec{p}) - G(\vec{p})\|^2\} = \frac{1}{r} \cdot \left(1 - \|G(\vec{p})\|^2\right)$$

⇒ Für wachsende Populationsgrößen $r \to \infty$ wird der stochastische Adaptionsalgorithmus $\tau(\vec{p})$ mit steigender Genauigkeit durch die Heuristik $G(\vec{p})$ approximiert.

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Heuristik

Die Trajektorie

$\vec{p}, \tau(\vec{p}), \tau^2(\vec{p}), \tau^3(\vec{p}), \ldots$

folgt näherungsweise der Trajektorie

$\vec{p}, G(\vec{p}), G^2(\vec{p}), G^3(\vec{p}), \ldots$

Die Heuristik $G(\vec{p})$ zeigt punktiertes Equilibruum, d.h. Folgen von Perioden mit relativer Stabilität nahe eines Fixpunktes $\vec{\omega} = G(\vec{\omega})$ unterbrochen von plötzlichen Übergängen zu anderen dynamischen Gleichgewichten nahe eines anderen Fixpunktes.

Der Anteil der Zeit in der Nähe instabiler Fixpunkte geht für $r \to \infty$ gegen 0.

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Da eine Population aus r Individuen besteht, sind die relativen Häufigkeiten rationale Zahlen bzw. Brüche.

$$p_i = r^{-1} \sum_{j \in P} [j = i]$$

Die Menge aller Populationsvektoren entspricht einer diskreten Untermenge des Simplex Λ

$$\vec{p} \in \left\{ \frac{1}{r}\left(s_0, s_1, \ldots, s_{n-1}\right) \;:\; s_i \in \mathbb{N}_0 \;\wedge\; \sum_{i=0}^{n-1} s_i = r \right\} \subset \Lambda$$

mit der Kardinalität $\binom{n+r-1}{r}$.

Übergangswahrscheinlichkeiten

$$\Pr\left\{\tau(\vec{p}) = \vec{q}\right\} = \Pr\left\{\vec{q} \mid \vec{p}\right\} = r! \cdot \prod_{i=0}^{n-1} \frac{G(\vec{p})_i^{r q_i}}{(r q_i)!}$$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Diskrete Untermenge eines Simplex

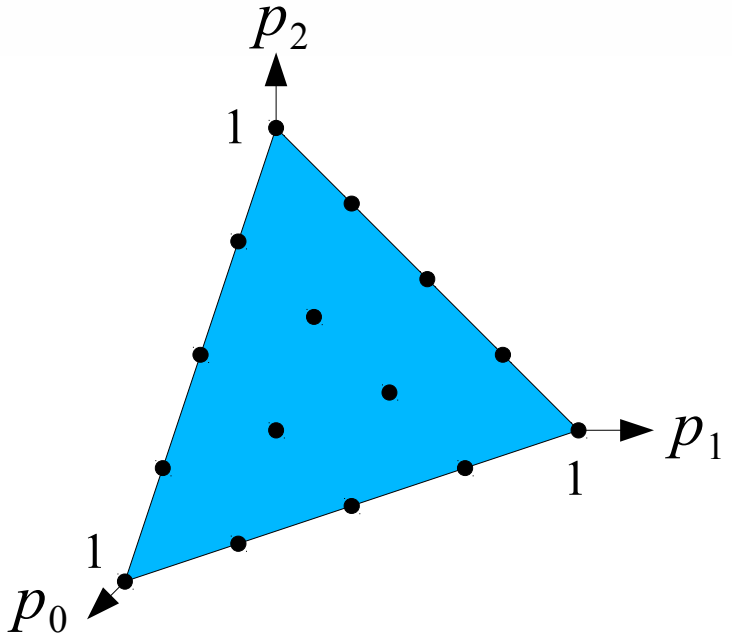
$$\left\{ \frac{1}{r}\left(s_0, s_1, \ldots, s_{n-1} \right) \;:\; s_i \in \mathbb{N}_0 \;\wedge\; \sum_{i=0}^{n-1} s_i = r \right\} \subset \Lambda$$

Ein Populationsvektor entspricht dem diskreten Vektor $\vec{p} = \frac{1}{r}\left(s_0, s_1, \ldots, s_{n-1} \right)$.

Beispiel für $n = 3$ und $r = 4$

mit der Kardinalität

$$\binom{n+r-1}{r} = \binom{3+4-1}{4} = 15$$



Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Selektion

$$S_\Omega(\vec{p})_i = \Pr\{\text{Individuum } i \text{ wird selektiert}\}$$

Die Selektion geschieht auf Basis der Fitness $f_i = f(i)$ der Individuen $i \in \Omega$.

Fitnessproportionale Selektion

$$S_\Omega(\vec{p})_i = \frac{f_i \cdot p_i}{\sum_{j=0}^{n-1} f_j \cdot p_j} = \frac{f_i \cdot p_i}{\vec{f}^T \cdot \vec{p}} \quad \Rightarrow \quad S_\Omega(\vec{p}) = \frac{1}{\vec{f}^T \cdot \vec{p}} \cdot \mathrm{diag}(\vec{f}) \cdot \vec{p}$$

mit dem Fitnessvektor $\vec{f} = (f_0, f_1, \ldots, f_{n-1})^T$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Mutation

Jedes Bit des binären l-Tupels $a \in \Omega$ wird mit der Mutationswahrscheinlichkeit μ mutiert.

Mutationsoperator $\mu_\Omega : \Omega \to \Omega$

$$\mu_\Omega(a) = a \oplus m$$

mit der Mutationsmaske $m \in \Omega$.

Die Mutationsmaske $m \in \Omega$ wird zufällig entsprechend dem Wahrscheinlichkeitsvektor $\vec{\mu} = \left(\mu_0, \mu_1, \ldots, \mu_{n-1}\right)^T$ gewählt mit der Wahrscheinlichkeit

$$\mu_m = \mu^{\mathbf{1}^T m} \cdot (1-\mu)^{l - \mathbf{1}^T m}$$

Hierbei bezeichnet $\mathbf{1}^T m$ die Anzahl der Einsen in m.

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Crossover

Die beiden binären l-Tupel $a \in \Omega$ und $b \in \Omega$ werden mit der Crossoverwahrscheinlichkeit χ gekreuzt.

Crossoveroperator $\chi_\Omega : \Omega \times \Omega \to \Omega$

$$\chi_\Omega(a, b) = \begin{cases} a \otimes m \oplus \bar{m} \otimes b & \text{mit der Wahrscheinlichkeit } 1/2 \\ a \otimes \bar{m} \oplus m \otimes b & \text{mit der Wahrscheinlichkeit } 1/2 \end{cases}$$

mit der Crossovermaske $m \in \Omega$.

Fachhochschule Münster University of Applied Sciences

Theorie genetischer Algorithmen

Uniform Crossover

Jedes Bit des Nachkommen $a \otimes m \oplus \bar{m} \otimes b$ wird mit der Wahrscheinlichkeit χ von dem Elter a und mit der Wahrscheinlichkeit $1-\chi$ von dem Elter b gewählt.

Die Crossovermaske $m \in \Omega$ wird zufällig mit der Wahrscheinlichkeit

$$\chi_m = \begin{cases} 2^{-l} \cdot \chi & , \quad \text{falls } m > 0 \\ 1 - \chi + 2^{-l} \cdot \chi & , \quad \text{falls } m = 0 \end{cases}$$

gewählt.

$\Rightarrow$ Wahrscheinlichkeitsvektor $\vec{\chi} = \left(\chi_0, \chi_1, \ldots, \chi_{n-1} \right)^T$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

1-Point Crossover

Mit einer zufällig gewählten Kreuzungsstelle $1 \le \lambda \le l-1$ stammt das linke Teiltupel des Nachkommen $a \otimes m \oplus \bar{m} \otimes b$ von dem Elter a und das rechte Teiltupel von dem Elter b (entsprechend für den Nachkommen $a \otimes \bar{m} \oplus m \otimes b$).
Die Kreuzung geschieht mit der Wahrscheinlichkeit χ.

Die Crossovermaske $m \in \Omega$ wird zufällig gewählt mit der Wahrscheinlichkeit

$$\chi_m = \begin{cases} \dfrac{\chi}{l-1} & , \text{ falls } m = 2^\lambda - 1 \text{ mit } 1 \le \lambda \le l-1 \\ 1-\chi & , \text{ falls } m = 0 \\ 0 & , \text{ sonst} \end{cases}$$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Kreuzung (Mixing)

$$\Pr\left\{c=\mu_\Omega(\chi_\Omega(a,b))\right\}$$

$$=\sum_{j=0}^{n-1}\mu_j\cdot\Pr\left\{\chi_\Omega(a,b)=c\oplus j\right\}$$

$$=\sum_{j=0}^{n-1}\mu_j\sum_{i=0}^{n-1}\chi_i\cdot\frac{[a\otimes i\oplus\bar{i}\otimes b=c\oplus j]+[a\otimes\bar{i}\oplus i\otimes b=c\oplus j]}{2}$$

$$=\sum_{j=0}^{n-1}\mu_j\sum_{i=0}^{n-1}\frac{\chi_i+\chi_{\bar{i}}}{2}\cdot[a\otimes i\oplus\bar{i}\otimes b=c\oplus j]$$

$\Rightarrow$

$$\Pr\left\{c=\mu_\Omega(\chi_\Omega(a,b))\right\}=\sum_{i,j\in\Omega}\mu_j\cdot\frac{\chi_i+\chi_{\bar{i}}}{2}\cdot[a\otimes i\oplus\bar{i}\otimes b=c\oplus j]$$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Kreuzung (Mixing)

Es gilt

$$\Pr\left\{\mu_\Omega(\chi_\Omega(a,b))=c\right\}=\Pr\left\{\mu_\Omega(\chi_\Omega(a\oplus c,b\oplus c))=0\right\}$$

$n\times n$ Mixing-Matrix $\quad M=\left(M_{i,j}\right)_{0\le i,j\le n-1}$

$$M_{i,j}=\Pr\left\{\mu_\Omega(\chi_\Omega(i,j))=0\right\}$$

mit

$$M_{i,j}=\sum_{u,v\in\Omega}\mu_v\cdot\frac{\chi_u+\chi_{\bar{u}}}{2}\cdot[i\otimes u\oplus\bar{u}\otimes j=v]$$

$\Rightarrow$ $\Pr\left\{\mu_\Omega(\chi_\Omega(a,b))=c\right\}=M_{a\oplus c,b\oplus c}$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Kreuzung (Mixing)

Spezialfall: nur Mutation, kein Crossover

$$\chi_m = [m = 0]$$

$$\Rightarrow M_{i,j} = \sum_{u,v \in \Omega} \mu_v \cdot \frac{\chi_u + \chi_{\bar{u}}}{2} \cdot [i \otimes u \oplus \bar{u} \otimes j = v]$$

$$= \sum_{v \in \Omega} \mu_v \cdot \frac{1}{2} \cdot \left([j = v] + [i = v] \right)$$

$$= \frac{\mu_i + \mu_j}{2}$$

$$\Rightarrow \boxed{M_{i,j} = \frac{\mu_i + \mu_j}{2}}$$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Heuristik

Wegen $G(\vec{p})_i = \Pr\{\text{Individuum } i \text{ ist in der Stichprobe von } \Omega\}$ gilt

$$
\begin{aligned}
& G(\vec{p})_i \\
&= \Pr\{i \text{ ist in der nächsten Generation}\} \\
&= \sum_{u,v\in\Omega} \Pr\{u \text{ selektiert}\} \cdot \Pr\{v \text{ selektiert}\} \cdot \Pr\{\mu_\Omega(\chi_\Omega(u,v)) = i\} \\
&= \sum_{u,v\in\Omega} S_\Omega(\vec{p})_u \cdot S_\Omega(\vec{p})_v \cdot \Pr\{\mu_\Omega(\chi_\Omega(u\oplus i, v\oplus i)) = 0\} \\
&= \sum_{u,v\in\Omega} S_\Omega(\vec{p})_u \cdot S_\Omega(\vec{p})_v \cdot M_{u\oplus i, v\oplus i} \\
&= \sum_{u,v\in\Omega} S_\Omega(\vec{p})_{u\oplus i} \cdot S_\Omega(\vec{p})_{v\oplus i} \cdot M_{u,v}
\end{aligned}
$$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Heuristik

Insgesamt folgt

$$G(\vec{p})_i = \sum_{u,v \in \Omega} S_\Omega(\vec{p})_{u \oplus i} \cdot S_\Omega(\vec{p})_{v \oplus i} \cdot M_{u,v}$$

Unter Verwendung der Kreuzung (Mixing) $M_\Omega(\vec{p})$ mit

$$M_\Omega(\vec{p})_i = \sum_{u,v \in \Omega} p_{u \oplus i} \cdot p_{v \oplus i} \cdot M_{u,v}$$

und der Selektion $S_\Omega(\vec{p})$ folgt $G(\vec{p}) = M_\Omega(S_\Omega(\vec{p}))$ bzw.

$$G = M_\Omega \circ S_\Omega$$

Übung

Genetische Algorithmen

Stellen Sie für einen genetischen Algorithmus SGA mit 1-Point Crossover und Mutation, der auf binären l-Tupeln mit $l = 3$ arbeitet, die Mixing-Matrix M auf.

Geben Sie für die fitnessproportionale Selektion und die ONEMAX-Funktion

$$f_i = \mathbf{1}^T i = \text{Anzahl der Einsen in } i$$

die Heuristik G an.

Simulieren Sie die Heuristik G und vergleichen Sie die Folge der Populationsvektoren mit der durch Simulation eines SGA erhaltenen Folge.

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Heuristik

Wegen $$G(\vec{p})_i = \sum_{u,v \in \Omega} S_\Omega(\vec{p})_{u \oplus i} \cdot S_\Omega(\vec{p})_{v \oplus i} \cdot M_{u,v}$$

ist die $n \times n = 2^l \times 2^l$-Mixing-Matrix mit $|\Omega| = n = 2^l$ zum Verständnis des genetischen Algorithmus wichtig.

Zur effizienten Berechnung kann die Walsh-Hadamard-Transformation verwendet werden.

Vektoren $\hat{\vec{x}} = W \cdot \vec{x}$ und $\vec{x} = W \cdot \hat{\vec{x}}$

Matrizen $\hat{A} = W \cdot A \cdot W$ und $A = W \cdot \hat{A} \cdot W$

mit der $n \times n$-Transformationsmatrix $W = \left(W_{i,j} \right)_{0 \le i,j \le n-1}$

$$W_{i,j} = n^{-1/2} \cdot (-1)^{i^T j}$$ mit $W^{-1} = W = W^T$

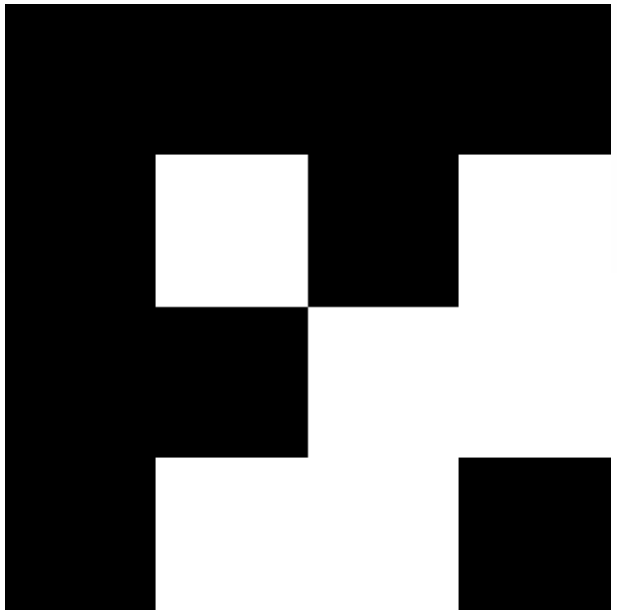
Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Walsh-Hadamard-Transformation

$$W_{i,j} = n^{-1/2} \cdot (-1)^{i^T j}$$

$$n = 2^1 = 2$$



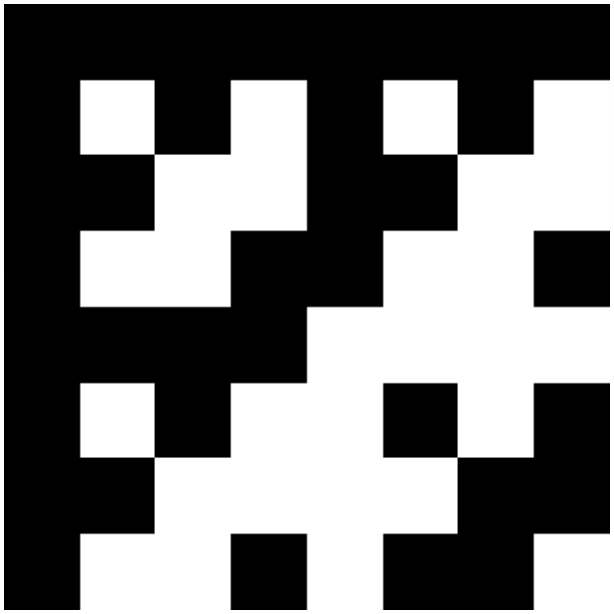
$$n = 2^2 = 4$$

Evolutionäre Algorithmen

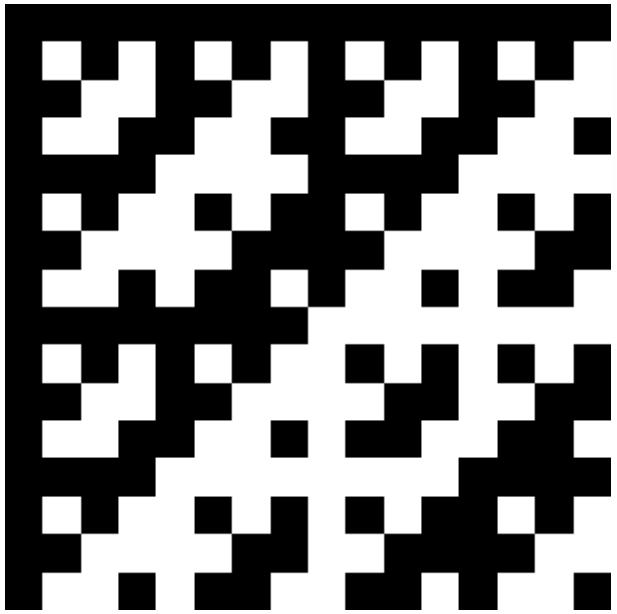
Theorie genetischer Algorithmen

Walsh-Hadamard-Transformation

$$W_{i,j} = n^{-1/2} \cdot (-1)^{i^T j}$$



$n = 2^3 = 8$



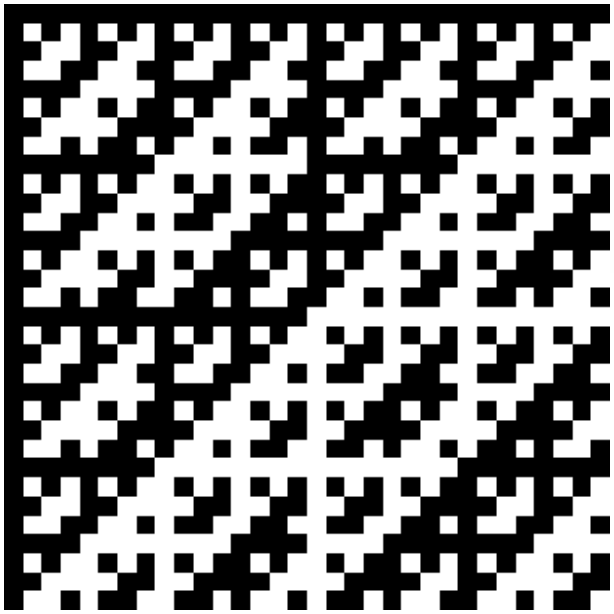
$n = 2^4 = 16$

Evolutionäre Algorithmen

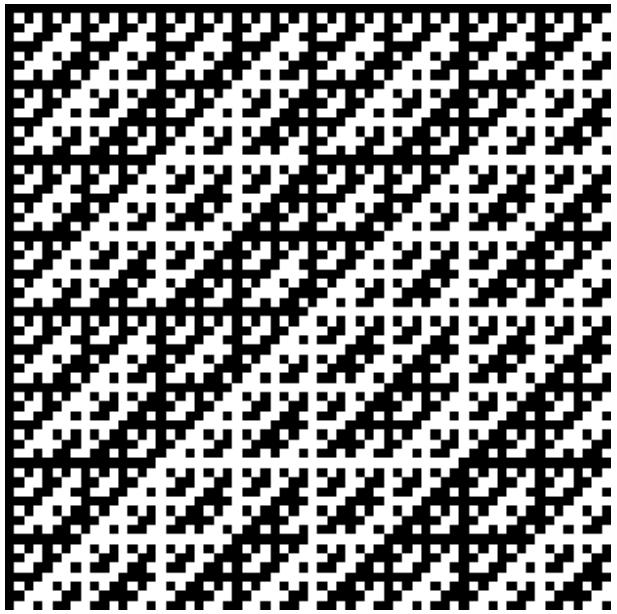
Theorie genetischer Algorithmen

Walsh-Hadamard-Transformation

$$W_{i,j} = n^{-1/2} \cdot (-1)^{i^T j}$$



$$n = 2^5 = 32$$



$$n = 2^6 = 64$$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Walsh-Hadamard-Transformierte der Mixing-Matrix

$$\widehat{M}_{i,j} = n^{1/2} \cdot [i^T j = 0] \cdot \hat{\mu}_{i \oplus j} \cdot \sum_{k \in \Omega_{\bar{i} \otimes \bar{j}}} \left(\chi_{k \oplus i} + \chi_{k \oplus j} \right)$$

mit $\Omega_k = \left\{ i \in \Omega \;:\; i \otimes \bar{k} = 0 \right\}$

Mit der Walsh-Hadamard-Transformierten

$$\hat{\mu}_i = n^{-1/2} \cdot \left(1 - 2\mu \right)^{\mathbf{1}^T i}$$

des Mutations-Wahrscheinlichkeitsvektors $\vec{\mu}$ folgt

$$\widehat{M}_{i,j} = [i^T j = 0] \cdot \left(1 - 2\mu \right)^{\mathbf{1}^T (i \oplus j)} \sum_{k \in \Omega_{\bar{i} \otimes \bar{j}}} \left(\chi_{k \oplus i} + \chi_{k \oplus j} \right)$$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Walsh-Hadamard-Transformierte der Mixing-Matrix

Spezialfall: nur Mutation, kein Crossover

$$M_{i,j} = \frac{\mu_i + \mu_j}{2}$$

$$\Rightarrow \quad \hat{M}_{i,j} = \sum_{u,v} W_{i,u} \cdot M_{u,v} \cdot W_{v,j}$$

$$= \sum_{u,v} W_{i,u} \cdot \frac{\mu_u + \mu_v}{2} \cdot W_{v,j}$$

$$= \frac{1}{2} \sum_{v} W_{v,j} \underbrace{\sum_{u} W_{i,u}}_{=\hat{\mu}_i} \cdot \mu_u + \frac{1}{2} \sum_{u} W_{i,u} \underbrace{\sum_{v} W_{j,v}}_{=\hat{\mu}_j} \cdot \mu_v$$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Walsh-Hadamard-Transformierte der Mixing-Matrix

Spezialfall: nur Mutation, kein Crossover

$$\Rightarrow \quad \widehat{M}_{i,j} = \frac{1}{2} \underbrace{\sum_{v} W_{v,j}}_{=n^{1/2}\cdot[j=0]} \cdot \hat{\mu}_i + \frac{1}{2} \underbrace{\sum_{u} W_{i,u}}_{=n^{1/2}\cdot[i=0]} \cdot \hat{\mu}_j$$

$$= \frac{n^{1/2}}{2} \cdot \left(\hat{\mu}_i \cdot [j=0] + \hat{\mu}_j \cdot [i=0] \right)$$

$$\Rightarrow \quad \boxed{\widehat{M}_{i,j} = \frac{n^{1/2}}{2} \cdot \left(\hat{\mu}_i \cdot [j=0] + \hat{\mu}_j \cdot [i=0] \right)}$$

Matrix:

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Transformierte Mixing-Matrix für 1-Point Crossover

$$\widehat{M}_{i,j} = [i \otimes j = 0] \cdot (1-2\mu)^{\mathbf{1}^T i + \mathbf{1}^T j} \cdot \left((1-\chi) \cdot \frac{[i=0]+[j=0]}{2} + \chi \cdot \frac{(\mathrm{lo}(j)-\mathrm{hi}(i))^+ + (\mathrm{lo}(i)-\mathrm{hi}(j))^+}{2(l-1)} \right)$$

$$\text{mit } \mathrm{lo}(a) = \begin{cases} l-1 & , \quad a=0 \\ \min\{b : 2^b \otimes a > 0\} & , \quad \text{sonst} \end{cases}$$

$$\mathrm{hi}(a) = \begin{cases} 0 & , \quad a=0 \\ \max\{b : 2^b \otimes a > 0\} & , \quad \text{sonst} \end{cases}$$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Modelle

Äquivalenzrelation $i \equiv j$ mit $i, j \in \Omega$

Äquivalenzklassen $[i] = \{ j \in \Omega \ : \ j \equiv i \}$

Äquivalenzabbildung Ξ mit der Matrix

$$\Xi_{[i],j} = [i \equiv j] = \begin{cases} 1 & , \quad j \in [i] \\ 0 & , \quad j \notin [i] \end{cases} \quad \Rightarrow \quad i \equiv j \quad \Leftrightarrow \quad \Xi_{[i],j} = 1$$

Äquivalente Populationsvektoren

$$\vec{p} \equiv \vec{q} \quad \Leftrightarrow \quad \forall i \in \Omega \ : \ \sum_{j \in \Omega} [i \equiv j] \cdot p_j = \sum_{j \in \Omega} [i \equiv j] \cdot q_j$$

Für äquivalente Populationsvektoren $\vec{p}$ und $\vec{q}$ sind die Anteile innerhalb der Äquivalenzklassen $[i]$ identisch.

Theorie genetischer Algorithmen

Modelle

Populationsvektoren $\vec{p}, \vec{q} \in \Lambda$

Mit

$$\Xi_{[i],j} = [i \equiv j]$$

folgt

$$\sum_{j \in \Omega} [i \equiv j] \cdot p_j = \sum_{j \in \Omega} \Xi_{[i],j} \cdot p_j = \left(\Xi \cdot \vec{p} \right)_{[i]}$$

und somit

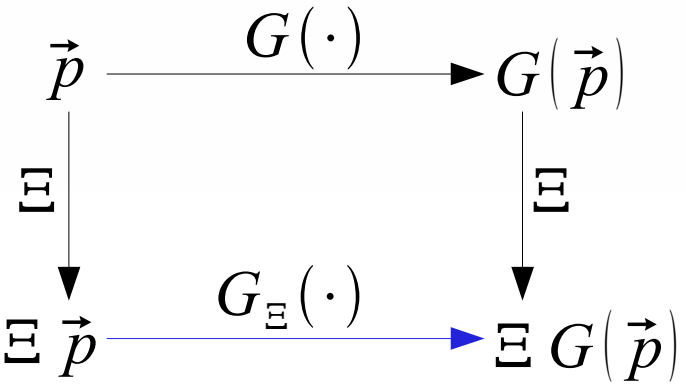
$$\vec{p} \equiv \vec{q} \quad \Leftrightarrow \quad \Xi \, \vec{p} = \Xi \, \vec{q}$$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Modelle

Kommutativität



$$\Xi\, G(\vec{p}) = G_\Xi(\Xi\, \vec{p})$$

Für ein konsistentes Modell muss das obige Diagramm kommutativ sein.

$$\vec{p} \equiv \vec{q} \quad \Rightarrow \quad G(\vec{p}) \equiv G(\vec{q})$$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Schemata als Äquivalenzklassen

$|\Omega_\xi| \times |\Omega|$ Matrix $\quad \Xi = \left(\Xi_{[i],j} \right)_{i \in \Omega_\xi,\, j \in \Omega}$

$$\Xi_{[i],j} = [\, j \otimes \xi = i \,]$$

mit Schemata-Familie $\xi \in \Omega$ und $\Omega_\xi = \left\{ i \in \Omega \;:\; i \otimes \bar{\xi} = 0 \right\}$

Anteil der Population in Schema (Äquivalenzklasse) $\quad i \in \Omega_\xi$

$$\left(\Xi \, \vec{p} \right)_{[i]} = \sum_{j \in \Omega} \Xi_{[i],j} \cdot p_j = \sum_{j \in \Omega} [\, j \otimes \xi = i \,] \cdot p_j = \sum_{j \in \Omega_{\bar{\xi}}} p_{i \oplus j}$$

$$\left(\Xi \, \vec{p} \right)_{[i]} = \sum_{j \in \Omega_{\bar{\xi}}} p_{i \oplus j}$$

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Schemata als Äquivalenzklassen

Äquivalenzrelation

$$i \equiv j \quad \Leftrightarrow \quad i \otimes \xi = j \otimes \xi$$

Zwei Individuen i und j sind äquivalent, wenn sie an den definierenden Positionen der Schemata-Familie ξ übereinstimmen.

Anzahl $\mathbf{1}^T \xi$ der definierenden Positionen von ξ

Kardinalität $|\Omega_\xi| = 2^{\mathbf{1}^T \xi}$ von $\Omega_\xi = \left\{ i \in \Omega \;:\; i \otimes \bar{\xi} = 0 \right\}$

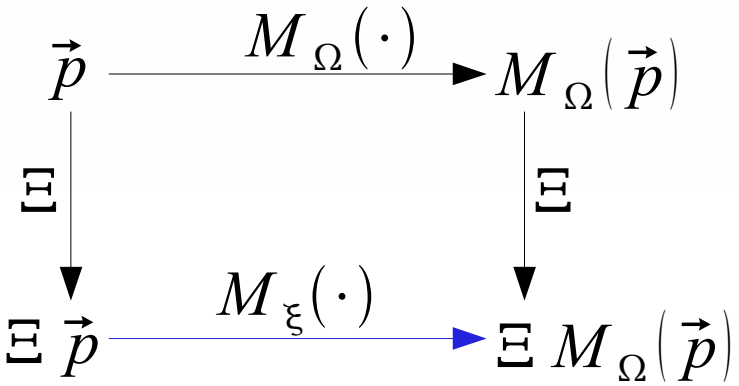
Mit der Kardinalität des Suchraums $|\Omega| = 2^l$ ist Ξ eine $2^{\mathbf{1}^T \xi} \times 2^l$-Matrix.

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Schemata als Äquivalenzklassen

Mixing M_Ω kommutiert.



$$\Xi\, M_\Omega(\vec{p}) = M_\xi(\Xi\, \vec{p})$$

Problem: Selektion S_Ω kommutiert nicht.

$\Rightarrow$ Heuristik $G = M_\Omega \circ S_\Omega$ kommutiert nicht.

Evolutionäre Algorithmen

Theorie genetischer Algorithmen

Die theoretische Beschreibung der Funktionsweise eines einfachen genetischen Algorithmus SGA auf Basis eines dynamischen Systems von M.D. Vose stellt die derzeit am besten ausgearbeitete Theorie dar.

Kritik

- Die Dimension der Mixing-Matrix M wächst exponentiell mit der Länge l der binären l-Tupel, so dass realistische Werte für l nicht mehr handhabbar werden.
- Die zu optimierende Fitness beeinflusst auf komplizierte Weise die Heuristik G mit ihren Fixpunkten und Trajektorien; hierdurch wird die Beurteilung des Konvergenzverhaltens des SGA bei der gegebenen Fitnessfunktion erschwert.