

# Fun mit Antennen und genetischen Algorithmen

Dr. Ralf Schlatterbeck  
Open Source Consulting

Email: [office@runtux.com](mailto:office@runtux.com)  
Web: <http://www.runtux.com>  
Tel. +43/650/621 40 17



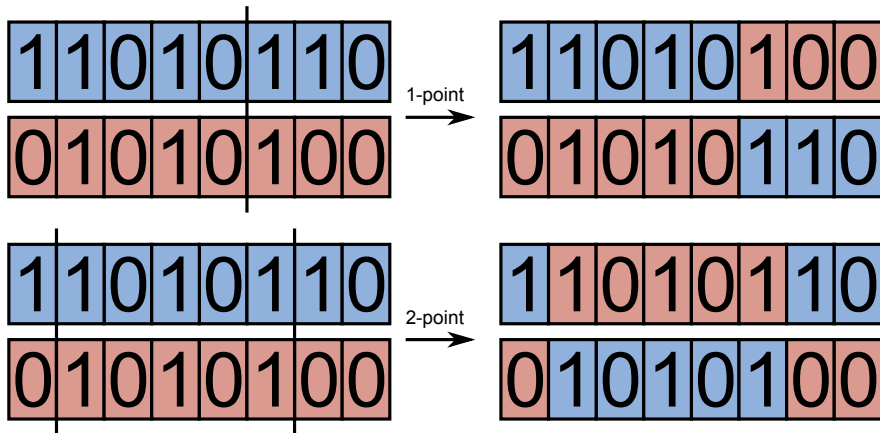
- Suche einer optimalen Lösung für ein Problem
- Inspiriert von der Natur
- Gene sind Sequenzen von Zahlen oder Symbolen (binär, Integer, Floating-Point)
- Aus einem Gen wird das Problem aufgebaut:  
Phänotyp
- Algorithmus arbeitet mit Population einer bestimmten Größe
- Fitness: Bewertungsfunktion wird optimiert
- Maximierung oder Minimierung möglich
- Selektion, Crossover, Mutation



- Selektion der besten Individuen
- Mehrere Auswahlverfahren möglich
- Deterministisch oder Stochastisch
- Proportional zur Fitness: Roulette-Rad mit Selektion proportional zur Fitness: Zu hoher Selektionsdruck am Anfang, zu niedrig am Schluss
- Truncation Selection (die K besten)
- Turnier mit N Teilnehmern  
z. B. 2 Teilnehmer: ziehe zwei Individuen, besseres gewinnt



# Crossover





# Mutation

1	1	0	1	0	1	1	0
---	---	---	---	---	---	---	---



1	1	0	1	1	1	1	0
---	---	---	---	---	---	---	---



# Aus der Sicht des GA: Maximiere (Blinde Suche!)

<u>String</u>	<u>Wert</u>
10111	10
01000	5
11010	3
00011	20

Was tun? [Gol02, S. 61]



## Schema Theorem

- Pattern-Matching 0, 1, \* (don't care)
- \*1010\* matcht 010100, 010101, 110100, 110101
- \* steht für *eine* Stelle
- Ein konkreter String steht für *viele* Schemata
- → Parallele Suche!
- Länge: vom ersten 0/1 bis zum letzten 0/1
- \*1010\* hat Länge 4
- Wenn Schemata lang werden
- ... werden sie mit hoher Wahrscheinlichkeit
- ... beim Crossover zersägt



## Schema Theorem

- Dinge die zusammengehören nahe im Gen
- ... leider weiß man das nicht immer
- Uniform Crossover: Für jedes Bit wird entschieden welcher Elternteil
- Two-Point oder One-Point Crossover besser als Uniform Crossover
- Moderne Algorithmen lernen Gen-Abhängigkeiten
- “Linkage Learning”
- z.B. Bayesian Optimization Algorithm (BOA)
- Schema Theorem John Holland 1975 [[Hol75](#)]





# Magisches Quadrat

- Zahlen von 1 bis  $n^2$
- Zeilensummen, Spaltensummen, Diagonalsumme
- ... sollen immer  $n * (n^2 + 1)/2$  sein
- Bewertungsfunktion:  
Betrag der Differenz zwischen soll und ist
- ... aufsummiert über alle möglichen Summen
- Aber: Mutation und Crossover zerstören Gültigkeit
- ... wenn wir naive Darstellung  $(1..n^2)$  wählen



## Magisches Quadrat

- Lösung: Spezielle Crossover und Mutations Operatoren
- ... dazu gibts *viel* Literatur wie man das machen kann
- z. B. Reparatur nach Crossover/Mutation
- Oder Prioritäten: Für jede Zahl von  $1..n^2$
- Priorität gibt Position im Quadrat an
- Prioritäten können Integer oder Floating-Point sein
- Literatur nennt diese Prios Zufallszahlen[Bea94]



- Zahlen von 0 bis 9, 0 heißt *leer*
- Bewertungsfunktion
  - bei zwei widersprüchlichen Sudokus gewinnt das mit *weniger* Zahlen
  - bei zwei mehrdeutigen Sudokus gewinnt das mit *mehr* Zahlen
  - bei zwei eindeutig lösbaeren Sudokus gewinnt das mit *weniger* Zahlen
  - widersprüchlich < mehrdeutig < eindeutig
- Bewertungsfunktion macht depth-first search
- ... und versucht das Sudoku zu lösen



- Simulation mit NEC (numerical electromagnetics code)
- Stammt aus den 80ern mit Lochkarten
- Rewrite in C/C++ „nec2c“
- nec2c: Übersetzt ein NEC file in Tabellen-Ausgabe
- xnec2: Nimmt ein NEC-File und zeigt Grafik
- xnecview nimmt nec2c Ausgabe und zeigt Grafik
- ... alle in Debian gepackaged



## Antennen-Parameter: Stehwellenverhältnis

- Stehwellenverhältnis VSWR  
(Voltage Standing Wave Ratio)
- Wenn eine Antenne gut angepasst ist wird alles abgestrahlt
- Wenn nicht wird ein Teil reflektiert

→ Stehwelle

$$\text{SWR} = \frac{|U_V| + |U_R|}{|U_V| - |U_R|}$$

- Ist 1 wenn die Antenne gut angepasst ist
- Gute Werte  $< 2$

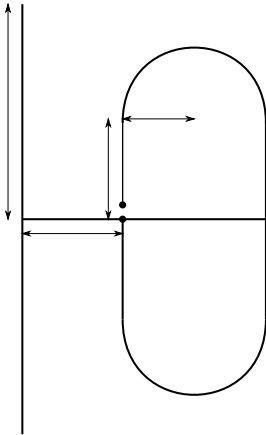


## Antennen-Parameter: Gewinn

- Gewinn gegenüber einem idealen Kugelstrahler
- “Isotropic Radiator”
- Kann man nicht bauen aber ist gut als Referenz
- Einheit: dBi (Dezi-Bel über isotropic radiator)
- Eine Antenne mit Richtwirkung hat Gewinn  $> 0\text{dB}$
- z.B. ein  $\lambda/2$ -Dipol hat 2.15 dBi Gewinn
- Gewinn gegenüber Dipol: dBd
- Umrechnung dBi  $\rightarrow$  dBd: subtrahiere 2.15
- X dB entspricht einem Faktor  $10^{X/10}$
- 3dB entspricht Faktor 2
- 10dB entspricht Faktor 10



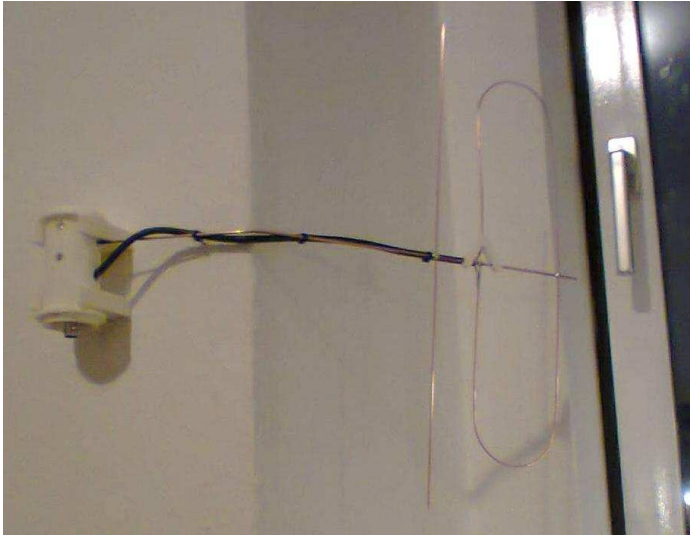
## 2-Elemente mit Faltdipol



- 4 Parameter
  - Coding als Floating-Point oder Binär
- Binär als Bits



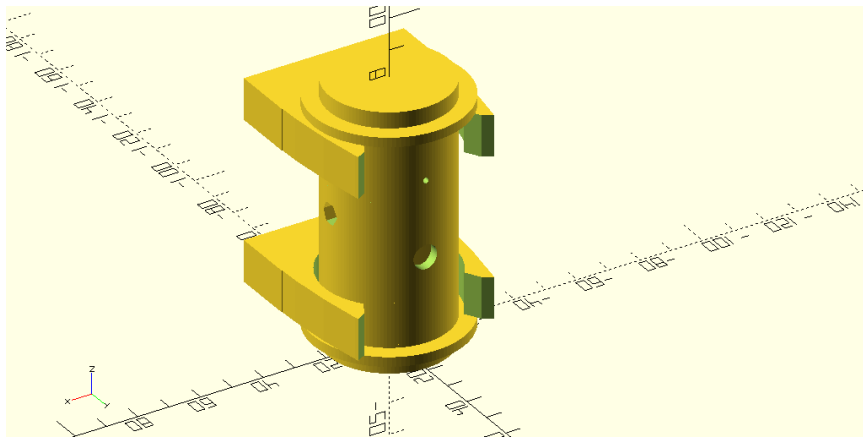
# Antenne montiert





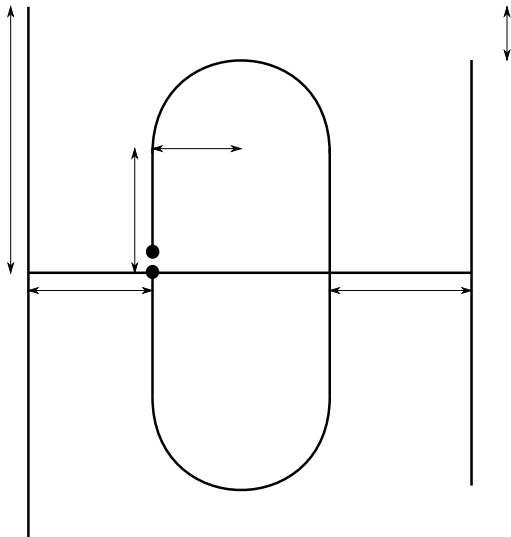


# Antennenmontage





# 3-Elemente mit Faltdipol





- **PGApack:**  
`https://github.com/schlatterbeck/pgapack`
- **PGApy:**  
`https://github.com/schlatterbeck/pgapy`
- **PyNEC:**  
`https://pypi.org/project/PyNEC/`
- **sudokumaker:**  
`https://pypi.org/project/sudokumaker/`



- [Bea94] James C. Bean. Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, 6(2):154–160, 1994.
- [Gol02] David E. Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Genetic Algorithms and Evolutionary Computation. Springer, August 2002.



[Hol75] John Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, Michigan, 1975.