



# 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



## Genetische Algorithmen

- 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 eine potentielle Lösung für das Problem aufgebaut: Phänotyp
- Algorithmus arbeitet mit Population einer bestimmten Größe
- Fitness: Bewertungsfunktion wird optimiert
- Maximierung oder Minimierung möglich
- *Individuum*: Genotyp + Phänotyp



## Skizze Algorithmus

1. Zufällige Initialisierung der Individuen
2. Bewertung der Individuen
3. Loop: Erzeugen neue Generation:
  - (a) Selektion: Auswahl für nächsten Schritt
  - (b) Anwendung genetische Operatoren auf selektierte Individuen
  - (c) Bewertung der neuen Individuen
  - (d) Erzeugen nächste Generation aus neuen und alten Individuen



## Selektion

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



## Selektionsdruck

Ohne Selektionsdruck haben wir keine Suche sondern eine zufällige Bewegung (random Walk) im Lösungsraum – Varianten:

- Selektion im ersten Schritt der Loop *Selektion*
- Selektion im letzten Schritt der Loop *nächste Generation*

Verschiedene Varianten von genetischen Algorithmen mit unterschiedlichen Strategien

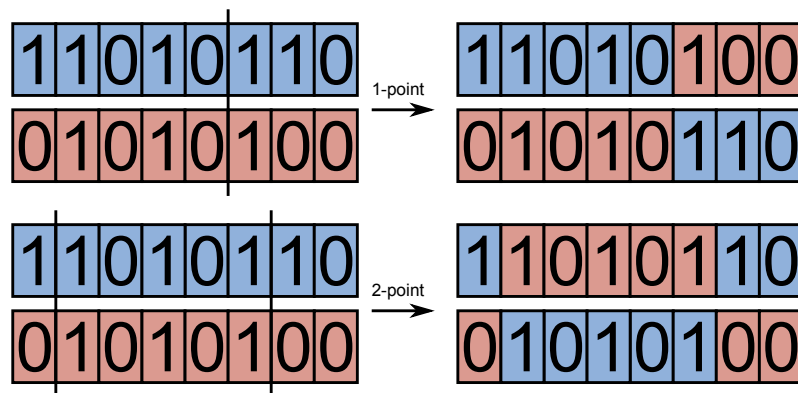


## Algorithmus klassisch

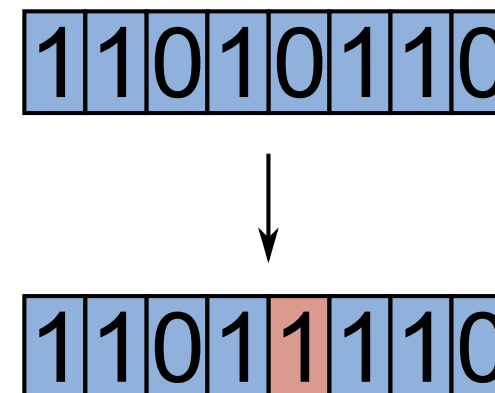
1. Zufällige Initialisierung der Individuen
2. Bewertung der Individuen
3. Loop: Erzeugen neue Generation:
  - (a) Selektion: *Tournament*
  - (b) *Crossover, Mutation* auf selektierte Individuen
  - (c) Bewertung der neuen Individuen
  - (d) nächste Generation: Übernahme der  $k$  besten aus der letzten Generation plus neu erzeugte Individuen



## Crossover



## Mutation





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

String	Wert
10111	10
01000	5
11010	3
00011	20

Was tun? [Gol02, S. 61]



## Variante Restricted Tournament Replacement

- Ersetzt letzten Schritt (erzeugen neue Generation)
- Für jedes neu erzeugte Individuum:
  - Wählt *window size* Individuen aus alter Generation
  - Suche das genetisch ähnlichste Individuum
  - Wenn neu erzeugtes Individuum besser: Ersetze das alte (genetisch ähnlichste) Individuum

Diese Variante hat Selektionsdruck!  
Und erhält die genetische Vielfalt länger!

[Har94, Har95]



## Variante Differential Evolution

Nur für Floating-Point Gene! [SP97, PSL05]

- Selektion: Nimm der Reihe nach jedes Individuum  $k$
- Wähle zufällig drei Individuen, alle verschieden und verschieden von  $k$
- Bilde Vektor-Differenz aus zwei dieser Individuen und addiere Differenz zum dritten
- Optional: Crossover des Resultats mit  $k$
- nächste Generation: Das neue Individuum ersetzt das alte wenn es besser ist

Numerische Optimierung – auch für Antennen!



## Implementation in PGPack

- PGPack: Parallel Genetic Algorithm Package
- Open Source genetic algorithm implementation
- Originally from David Levine, Argonne National Laboratory
- Parallel execution via MPI (Message Passing Interface)
- Construction kit for your experiments
- I'm maintaining a fork since 2017
- Current version implements all variants above and more
- PGPpy is a python-wrapper (since 2005)



## 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) [Pel05]
- 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 Reihenfolge! Auch bei anderen Problemen wie „Geschäftsreisende“ (Travelling Salesperson)



## 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 (random numbers) [Bea94]



## Sudoku

- 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 lösbaren Sudokus gewinnt das mit *weniger* Zahlen
  - widersprüchlich < mehrdeutig < eindeutig
- Bewertungsfunktion macht depth-first search
- ... und versucht das Sudoku zu lösen



## Antennen

- 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-Optimierung mit genetischen Algorithmen ist schon in den 90er Jahren publiziert [Lin97] aber vielleicht nicht genutzt weil bis 2015 patentiert? [AL95]



## Antennen und Genetische Algorithmen

- Breitband Antennen mit gedrehter Polarisation und Hemisphären-Abdeckung [AL97a, AL97b]
- Speziell interessant die *Crooked Wire Antenna*
- Zusammengefasste Ergebnisse von Linden (Diss.) [Lin97]
- Interessante Ergebnisse der NASA für Satelliten mit Einschränkungen (Platz in Radom) [HGLL06, HLL11]
- Gedrehte Yagi-Uda Antenne für gedrehte Polarisation (!) *Twisted Yagi* [Lin03]



## 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

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

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

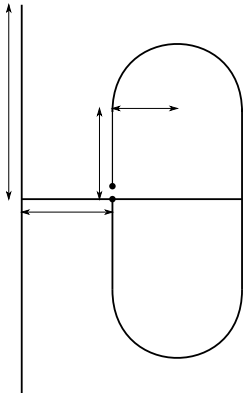


## Antennen-Parameter: Gewinn

- 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 > 0dB
- z.B. ein  $\lambda/2$ -Dipol hat 2.15 dBi Gewinn
- Gewinn gegenüber Dipol: dBd
- Umrechnung dBi → dBd: subtrahiere 2.15
- X dB entspricht einem Faktor  $10^{X/10}$
- 3dB entspricht Faktor 2
- 10dB entspricht Faktor 10



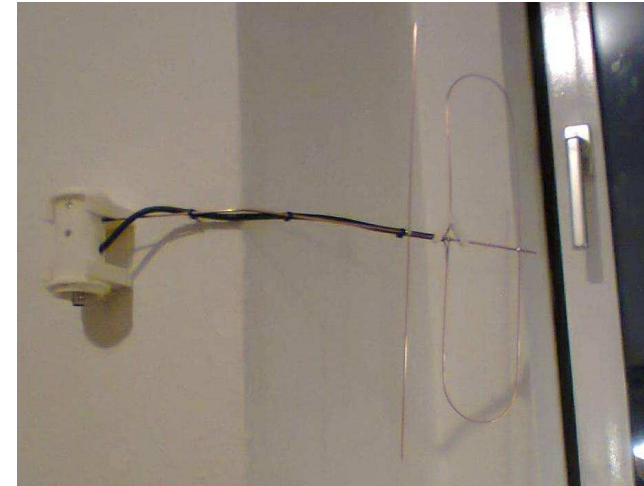
## 2-Elemente mit Faltdipol



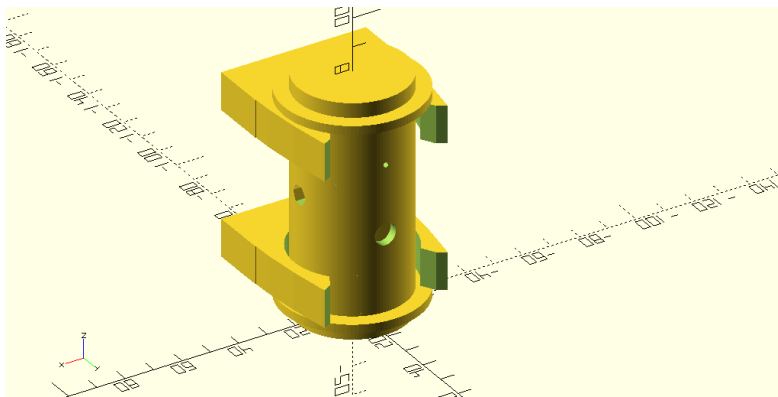
- 4 Parameter
- Coding als Floating-Point oder Binär
- Binär als Bits



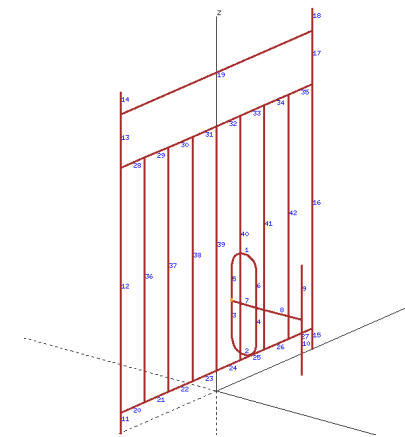
## Antenne montiert



## Antennenmontage



## Ausblick: Einbezug der Umgebung





## Software

- PGApack:  
[github.com/schlatterbeck/pgapack](https://github.com/schlatterbeck/pgapack)
- PGAPy:  
[github.com/schlatterbeck/pgapy](https://github.com/schlatterbeck/pgapy)
- PyNEC:  
[pypi.org/project/PyNEC/](https://pypi.org/project/PyNEC/)
- sudokumaker:  
[pypi.org/project/sudokumaker/](https://pypi.org/project/sudokumaker/)
- Antenna-Optimizer:  
[github.com/schlatterbeck/antenna-optimizer](https://github.com/schlatterbeck/antenna-optimizer)



## Bibliography

- [AL95] Edward E. Altshuler and Derek S. Linden. Process for the design of antennas using genetic algorithms. US Patent US5719794A, July 1995.
- [AL97a] Edward E. Altshuler and Derek S. Linden. Design of a loaded monopole having hemispherical coverage using a genetic algorithm. *IEEE Transactions on Antennas and Propagation*, 45(1):1–4, January 1997.



## Bibliography

- [AL97b] Edward E. Altshuler and Derek S. Linden. Wire-antenna designs using genetic algorithms. *IEEE Antennas and Propagation Magazine*, 39(2):33–43, April 1997.
- [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 Compentent*



## Bibliography

- Genetic Algorithms*. Genetic Algorithms and Evolutionary Computation. Springer, August 2002.
- [Har94] Georges Harik. Finding multiple solutions in problems of bounded difficulty. IlliGAL Report 94002, [Illinois Genetic Algorithm Lab](http://illigal.org), May 1994.
- [Har95] Georges R. Harik. Finding multimodal solutions using restricted tournament selection. In Larry J. Eshelman, editor, *Procee-*



## Bibliography

*dings of the International Conference on Genetic Algorithms (ICGA)*, pages 24–31. Morgan Kaufmann, July 1995.

[HGLL06] Gregory S. Hornby, Al Globus, Derek S. Linden, and Jason D. Lohn. Automated antenna design with evolutionary algorithms. In *AAIA Space Forum*, San Jose, California, September 2006. **American Institute of Aeronautics and Astronautics**.

[HLL11] Gregory S. Hornby, Jason D. Lohn, and De-



## Bibliography

rek S. Linden. Computer-automated evolution of an x-band antenna for NASA's space technology 5 mission. *Evolutionary Computation*, 19(1):1–23, 2011.

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

[Lin97] Derek S. Linden. *Automated Design and Optimization of Wire Antennas Using Genetic Algorithms*. Dissertation, Massachu-



## Bibliography

setts Institute of Technology, September 1997.

[Lin03] Derek S. Linden. The twisted yagi antenna optimized with a genetic algorithm: A potential alternative to the helix. In *IEEE Antennas and Propagation Society International Symposium*, volume 1, pages 153–156, Columbus, OH, 2003.

[Pel05] Martin Pelikan. *Hierarchical Bayesian Optimization Algorithm: Toward a New Gene-*



## Bibliography

*ration of Evolutionary Algorithms*, volume 170 of *Studies in Fuzziness and Soft Computing*. Springer, 2005.

[PSL05] Kenneth V. Price, Rainer M. Storn, and Jouni A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Springer, Berlin, Heidelberg, 2005.

[SP97] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous





spaces. *Global Optimization*, 11(4):341–359, December 1997.