

# Optimierung von Fließkomma-Problemen mit 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 („Mixing“)
  - (c) Bewertung der neuen Individuen
  - (d) Erzeugen nächste Generation aus neuen und alten Individuen („Replacement“)



## 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

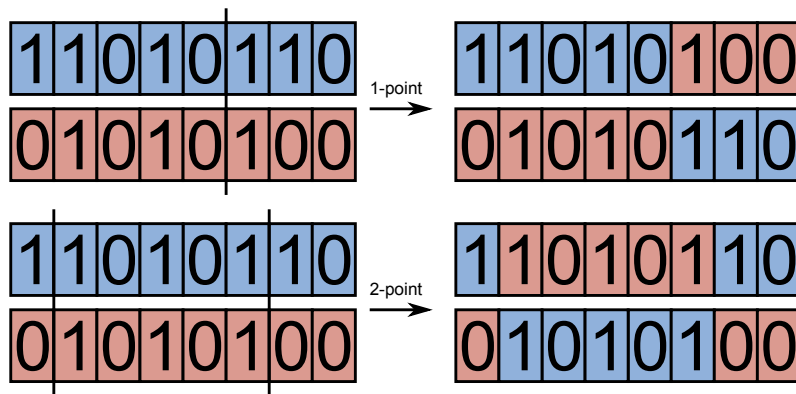


## Algorithmus klassisch

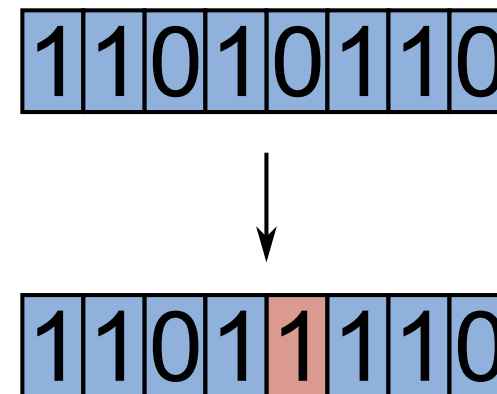
1. Zufällige Initialisierung der Individuen
2. Bewertung der Individuen
3. Loop: Erzeugen neue Generation:
  - (a) Selektion: *Tournament*
  - (b) *Crossover, Mutation* auf selektierte Individuen (niedrige Mutation!)
  - (c) Bewertung der neuen Individuen
  - (d) nächste Generation: Übernahme der  $k$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



## Crossover



## Mutation





## Fließkomma vs. Klassisch

- Zu wenig „Mixing“ mit klassischem Ansatz
  - Crossover soll Eigenschaften beider Vorgänger verwenden
  - ... ohne diese einfach zu kopieren
  - funktioniert mit Fließkomma nur ungenügend
- Anderer Ansatz nötig  
→ Differential Evolution



## Differential Evolution

- Gen ist ein Vektor von Fließkommazahlen
- Für alle Gene:
- Addiere gewichtete *Differenz* von zwei Genen zu drittem Gen
- Crossover mit aktuellem Gen
- Crossover verwendet mindestens eine „Allele“ des neuen Gens
- Crossover-Wahrscheinlichkeit je nach Problem klein (0.2) oder groß (0.9 – 1.0)
- Beispiel 2-Dimensionaler Fall:  $(x, y)$



## Differential Evolution

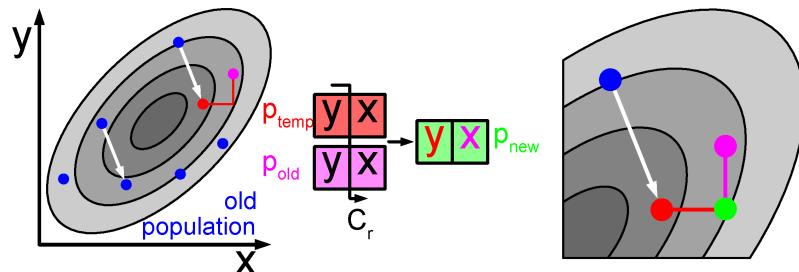


Abbildung 1: Differential Evolution Crossover [Sch23]



## 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 gewichtete Vektor-Differenz aus zwei dieser Individuen und addiere zum dritten
- Optional: Crossover des Resultats mit  $k$
- nächste Generation: Das neue Individuum ersetzt das alte wenn es besser ist

Numerische Optimierung!



## Differential Evolution Visualization

[Parcours for Differential Evolution]  
[Example with animated GIF]



## Differential Evolution Visualization

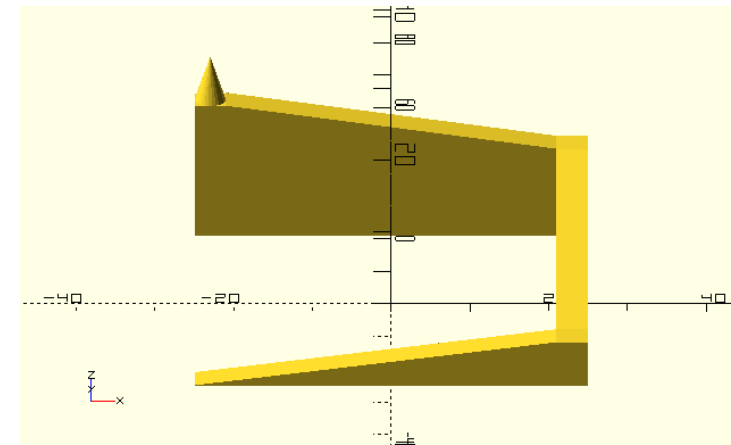


Abbildung 2: Parcours for Differential Evolution (with OpenS-CAD)



## Implementation in PGApack

- PGApack: 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 differential evolution and other variants
- PGAPy is a python-wrapper (since 2005)



## Randbedingungen

- Einfache „Box Constraints“
- Minimum und Maximum von jedem Parameter (=Gen)
- z.B. bei Längen: Sind nicht negativ
- ... und meist ist Größenordnung bekannt (mm, m, oder km)
- Von praktisch allen genetischen Algorithmen unterstützt
- Komplizierte Bedingungen: mehrere Zielfunktionen
- Eval-Funktion hat mehrere Rückgabewerte



## Mehrere Zielfunktionen

- Traditionell: Nur eine Zielfunktion (objective function)
  - Wird maximiert oder minimiert
  - Mehrere Zielfunktion:
    - für Randbedingungen (Constraints)
    - mehrere gleichzeitige Ziele
- Randbedingungen erschweren das Optimierungsproblem
- Ziele können sich widersprechen!
- Pareto-Optimierung



## Waveform Design for Inkjet

- Piezo-Electric Inkjet
- Nozzle width 23µm
- Droplet size around 6pL (picoliter)
- Waveform determines droplet quality
- Print quality: Speed, satellites (small drops after main drop), droplet size, stable at high jet-rate, . . .
- Analytic physical/mathematical solution is complex with many simplifications (to be able to compute it at all)
- Printhead dimensions are not given by manufacturer



## Dropwatcher

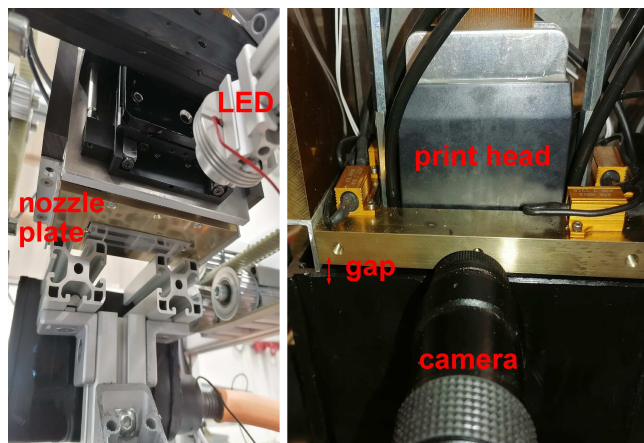


Figure 3: Dropwatcher Hardware [Sch23]



## Print Head

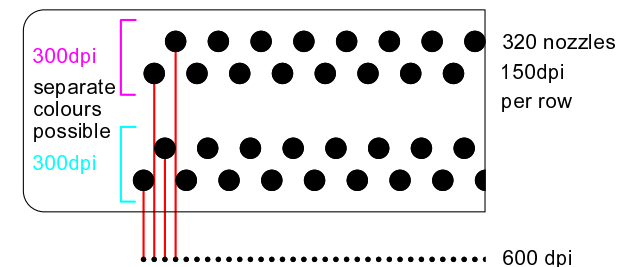


Figure 4: Print Head [Sch23]



## Image Recognition

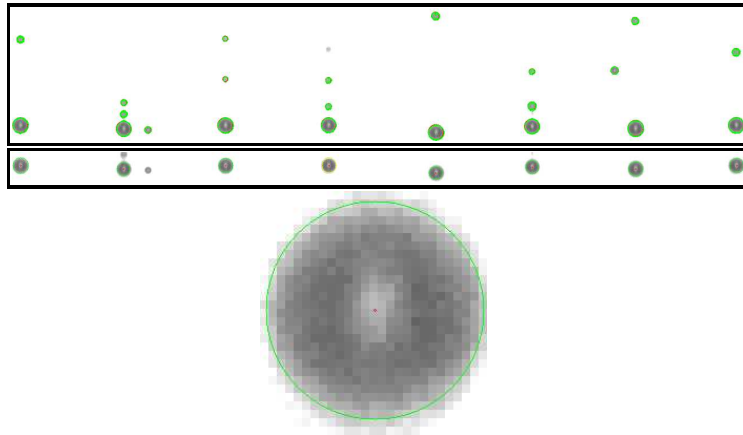


Figure 5: Blob- and Circle Detection [Sch23]



## Optimization with Differential Evolution

- Waveform parameters set by genetic algorithm
- Fitness determined from image recognition
- e.g. minimum speed, size of drop, number of satellites ...

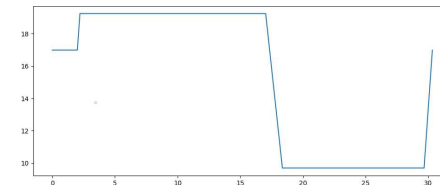


Figure 6: Typical waveform [Sch23]



## Drop Fronts

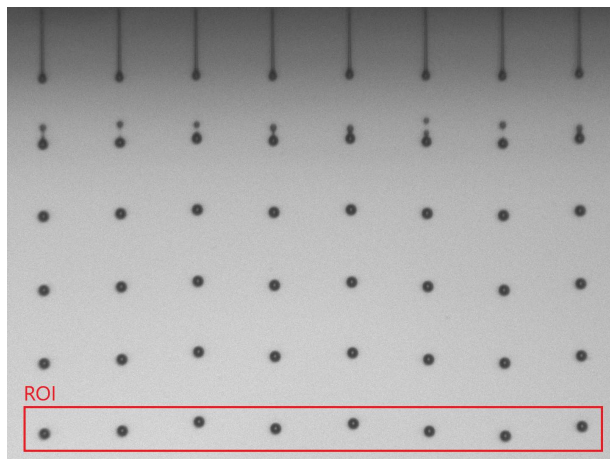


Figure 7: Drop Fronts Jetting at 32kHz [Sch23]



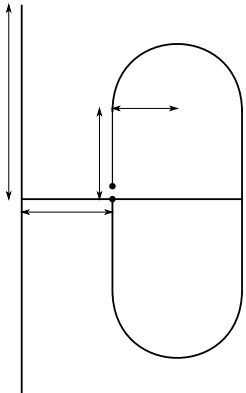
## Beispiel: Antennen

- Simulation mit NEC (numerical electromagnetics code)
- Stammt aus den 80ern mit Lochkarten
- Rewrite in C/C++ „nec2c“ Open Source
- 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]



## 2-Elemente mit Faltdipol



- 4 Parameter
  - Coding als Floating-Point
- Differential Evolution

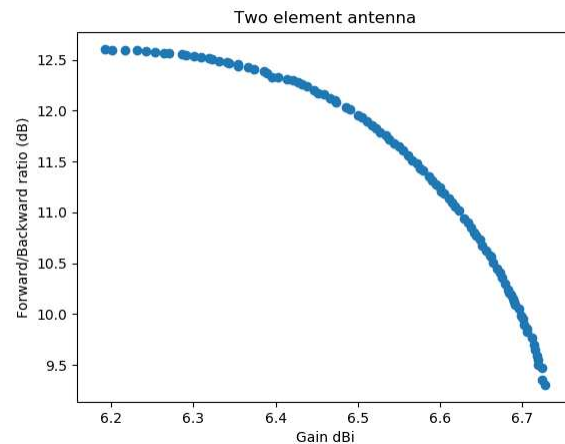


## Antennen-Optimierung mit NSGA-II

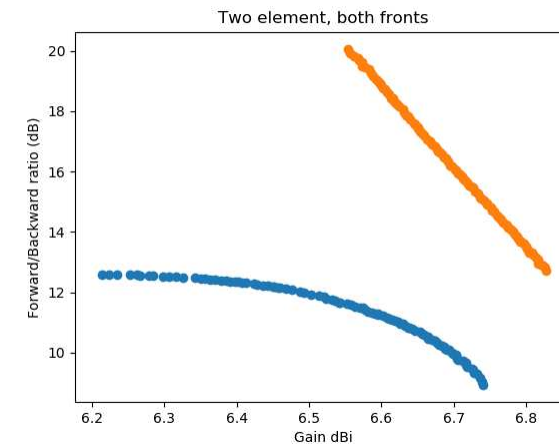
- Eine Randbedingung:  $VSWR < 1.8$
- Zwei Zielfunktionen: Gewinn und Vorwärts/Rückwärts
- Gen der Länge 4:
- Länge des Falt-Dipols
- Breite des Falt-Dipols
- Länge des zweiten Elements
- Abstand der Elemente
- Box-Constraints für alle Parameter



## Pareto Front optimierte Antenne

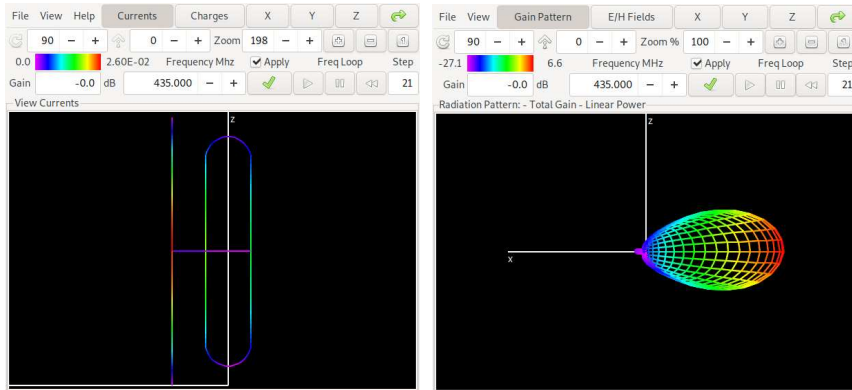


## ➤ Pareto Front anderer Optimierungslauf

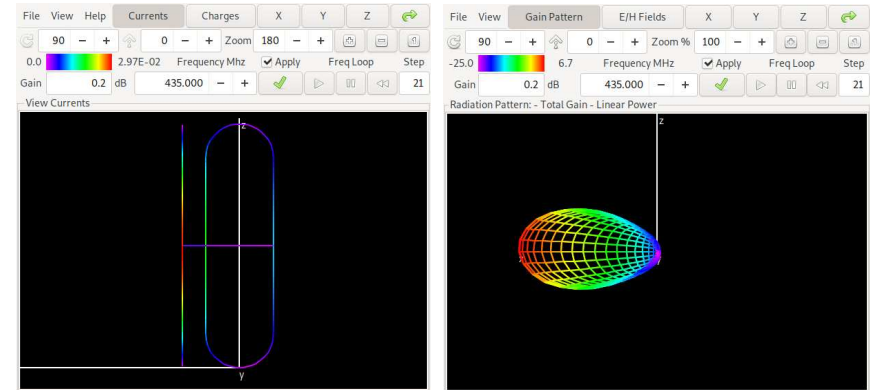




## Antenne Untere Front



## Antenne Obere Front



## 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/)
- Antenna-Optimizer:  
[github.com/schlatterbeck/antenna-optimizer](https://github.com/schlatterbeck/antenna-optimizer)
- Gearbox:  
[github.com/schlatterbeck/gearbox](https://github.com/schlatterbeck/gearbox)



## Bibliography

- [AL95] Edward E. Altshuler and Derek S. Linden. Process for the design of antennas using genetic algorithms. US Patent US5719794A, July 1995.
- [Lin97] Derek S. Linden. *Automated Design and Optimization of Wire Antennas Using Genetic Algorithms*. Dissertation, Massachusetts Institute of Technology, September 1997.



- [PSL05] Kenneth V. Price, Rainer M. Storn, and Joni A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Springer, Berlin, Heidelberg, 2005.
- [Sch23] Maximilian Schlatterbeck. Actuation waveform design for piezo electric inkjet applications using an evolutionary algorithm. Bachelor thesis, FH Technikum Wien, September 2023.
- [SP97] Rainer Storn and Kenneth Price. Differential



evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, December 1997.