Building Self-Optimizing SDRs Using Python DEAP

Zubair Shaik, André Puschmann

FOSDEM SDR devroom
January 31, 2016
Agenda

1. Introduction
2. Approach
3. Prototype
4. Conclusion
Motivation - Setting Up a Communication Link

Steps:
1. Get it to work somehow
2. Improve performance somehow (e.g., high spectral efficiency)
3. Obtain (sub-)optimal performance

Challenges:
- Huge amount of radio parameters
- Too many possible solutions
- Exhaustive search not tractable
Solution Space Example

- Tx power (multiple gain stages)
  - RF1: range(30, 70, 1) \(\times 40\)
  - RF2: range(0, 20, 1) \(\times 20\)
  - Soft: range(-15, 0, 1) \(\times 15\)

- Bandwidth \(\times 4\)
- Frequency \(\times 4\)
- Modulation (bpsk, qpsk, qam16, ..) \(\times 10\)
- Coding scheme (h128, ..) \(\times 10\)
- Subcarrier allocation \(\times 4\)
- Frame size range(100, 1500, 100) \(\times 14\)

- Number of combinations: 1,075,200,000
- Time for exhaustive search (1 try per second): 12,444d or 34y
Towards A Possible Solution

Desired goals/features:

- Find tractable solutions for optimizing radio link (multiple objectives)
- Applicable for over-the-air experiments
- Use available open-source solutions

Genetic Algorithms:

- Heuristic search inspired by biological organism
  ⇒ Optimization as evolutionary process
- Chromosome including genes that represents a trait
- Fitness function to select survivor of generation
- Good for large solution space (not optimal necessarily)

Source: Wikipedia on GA
Optimization Process Flow

- Initialization
- Parent selection
- Parent selection
- Population
- Survivor selection
- Parents
- Mutation/Crossover
- Offspring

Adopted from D. Kozel [1]
System Components

- Iris SDR framework
  - Exposes SDR as virtual network device via TUN/TAP
- Optimizer
  - Controller
  - Client
  - ZeroMQ/Protobuf-IPC for reconfiguring SDR
  - Distributed Evolutionary Algorithms in Python (DEAP)
- nuttcp
  - Generate UDP traffic
  - Third-party-mode
Controller Protocol

- Runs between controller and client over reliable connection
- FLEX_FRAME_CLI_CONFIG
  - Configure IPs for control and data connection
- FLEX_FRAME_TRY
  - Configure new radio parameters
- FLEX_FRAME_START_TRAFFIC
- FLEX_FRAME_RESULTS
GUI Screenshots
Video Demo
Conclusion

Currently:

- Extend evaluation and run more experiments
- Bug-fixes and code cleanup

What’s next?

- Add more optimization criteria (e.g., minimize EVM, energy)
- Add GNU Radio support (maybe through GSoC)

https://github.com/andrepuschmann/
Thanks!

andre.puschmann@tu-ilmenau.de
Ilmenau University of Technology, Germany
Visit us: http://www.tu-ilmenau.de/ics
References


https://github.com/andrepuschmann/OSPECORR.

https://github.com/DEAP.

http://nuttcp.net/nuttcp/.

André Puschmann
Integrated Communication Systems Group
http://www.tu-ilmenau.de/ics

Self-Optimizing SDRs
January 31, 2016