CMOS Inverter Optimisation using NSGA-II

Author: Kris Seunarine
Date: 2025-10-02

Overview

This repository contains a Python-based framework for **multi-objective optimisation** of a CMOS inverter using **NSGA-II**. The project focuses on balancing two critical performance metrics:

- **Propagation delay (`tp`)** – a measure of speed
- **Average current (`iavg`)** – a measure of power efficiency

The framework leverages **LTspice** simulations in batch mode for automated circuit evaluation and incorporates a **Differential Evolution (DE)** solution as a seed to accelerate convergence.

Key Features

- **Multi-Objective Optimisation:** Uses NSGA-II to explore the trade-off (Pareto front) between delay and current.
- **DE Seeding:** Integrates previously obtained DE-optimal solutions for faster and more reliable convergence.
- **Batch LTspice Simulations:** Runs simulations invisibly without GUI, enabling automated, large-scale evaluations.
- **Automatic Cleanup:** Temporary netlists and LTspice output files are deleted after each evaluation to keep the workspace tidy.
- **Pareto Front Results:** Outputs CSV file containing optimal transistor width (W) and length (L) along with evaluated `tp` and `iavg`.
- **Scalable Across Industries:** Though demonstrated for CMOS inverters, the methodology can be applied to optimise designs in **finance, manufacturing, energy systems, or any multi-objective engineering problem**.

Installation

1. Install Python 3.10+ and required libraries:

```
pip install deap
```

2. Ensure **LTspice** is installed and the path is correctly set in `LTSPICE_EXE`.

3. Place your LTspice circuit file in a known location and update `CIRCUIT_FILE` in the script.

Usage

Run the optimisation script:

```
python cmos_inverter_optimisation.py
```

The script will:

1. Generate temporary netlists for each evaluation.
2. Run LTspice in batch mode to extract `tp` and `iavg`.
3. Apply NSGA-II to evolve a population towards Pareto-optimal solutions.
4. Clean up temporary files automatically.
5. Save the results to `pareto_front_results.csv`.

Example Output

```
Evaluated W=8.0162 µm, L=0.4964 µm -> tp=7.284e-09, iavg=8.009e-07
Evaluated W=5.1666 µm, L=1.5337 µm -> tp=4.281e-09, iavg=1.364e-08
...
Top solutions in Hall of Fame:
W=11.2935, L=2.0224, tp=1.099e-09, iavg=5.205e-08
```

You can visualise the **Pareto Front** to analyse the trade-off between speed and power consumption.

Benefits

- **For Engineers:** Quickly identify optimal transistor dimensions without manual trial and error.
- **For CTOs & Product Teams:** Supports data-driven decision-making to reduce cost and maximise performance.
- **Cross-Industry Applicability:** Optimisation framework suitable for multi-objective problems in finance (risk vs return), manufacturing (throughput vs cost), energy (efficiency vs reliability), and more.

Limitations

- Small transistor sizes may trigger LTspice warnings.

- Simulation failures are penalised in optimisation but do not halt the script.
- Extraction of `tp` and `iavg` is currently simulated with placeholder values; integration with a **PyLTSpice parser** is recommended for real circuits.

Licence