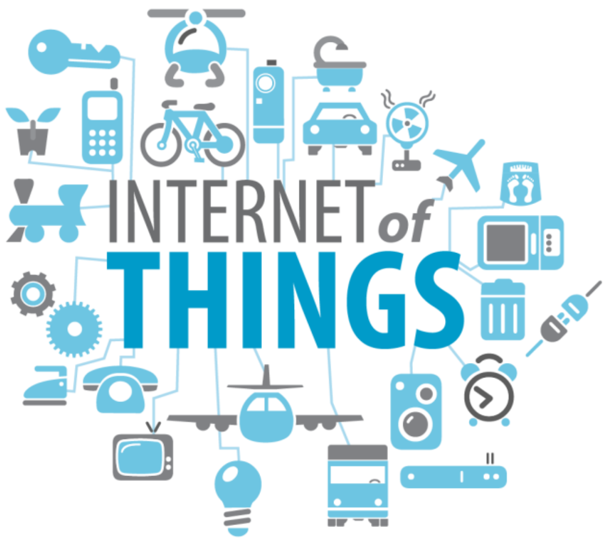
APOSTLE: Asynchronously Parallel Optimization Scheme for Sizing Analog Circuits using DNN Learning

Ahmet F. Budak[1], David Smart[2], Brian Swahn[2], David Z. Pan[1]

[1] ECE Department, UT Austin
[2] Analog Devices Inc.

Analog ICs: Introduction

- Sensor related applications and real-world interfaces require analog circuits.
- Increasing market demand: Internet of Things (IoT), autonomous and electric vehicles, communication and 5G networks…



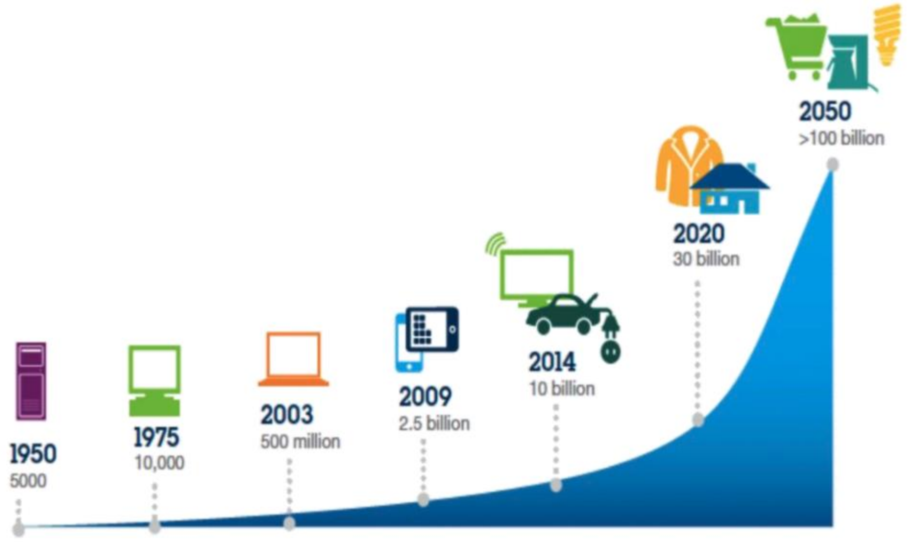
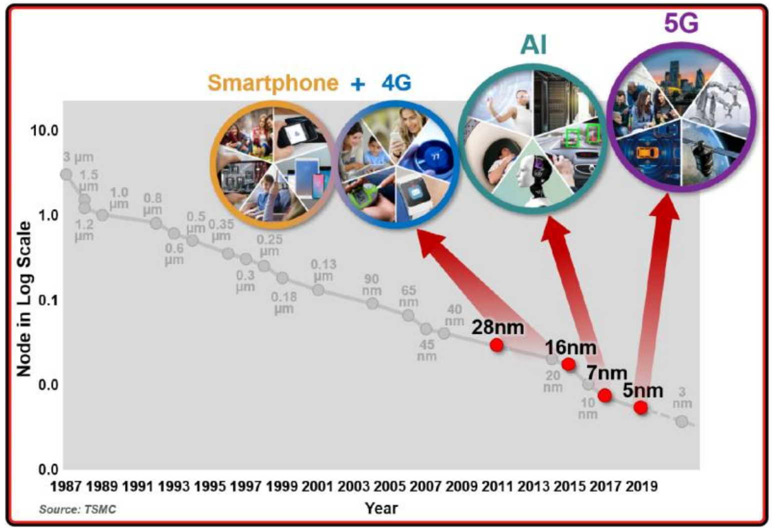
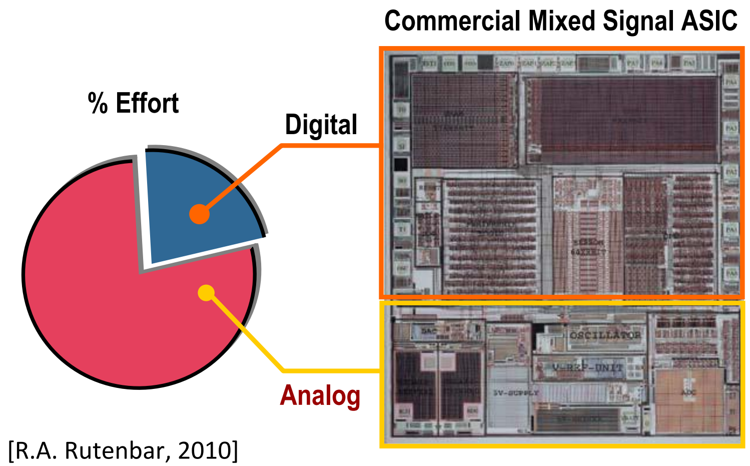


Image Sources: IBM, Ansys, public technology

Motivations for Analog Automation

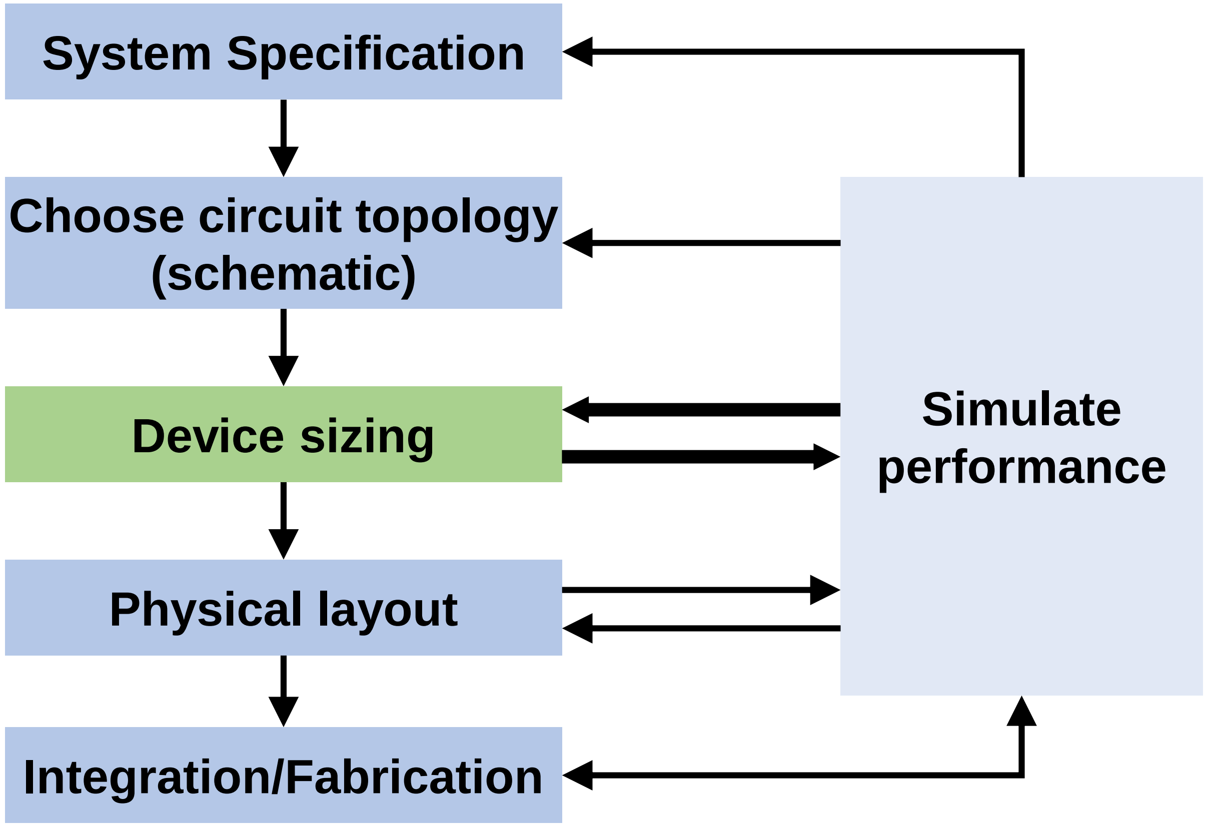




[R.A. Rutenbar, 2010]

- There exist repeating tasks:
  - Design is carried from one process fap to another.
  - Same design needs to be altered for a new set of performance specifications.
- Analog is here to stay:
  - Not all analog blocks can be converted to digital
  - Converting everything to digital and exploiting the existed automation is not a viable option.
- Better community & computers

Analog Design Challenge



- Simulation is involved everywhere, and it is sometimes very costly.
- Manual and iterative process.
- Sizing/resizing is required.

Analog Sizing Task

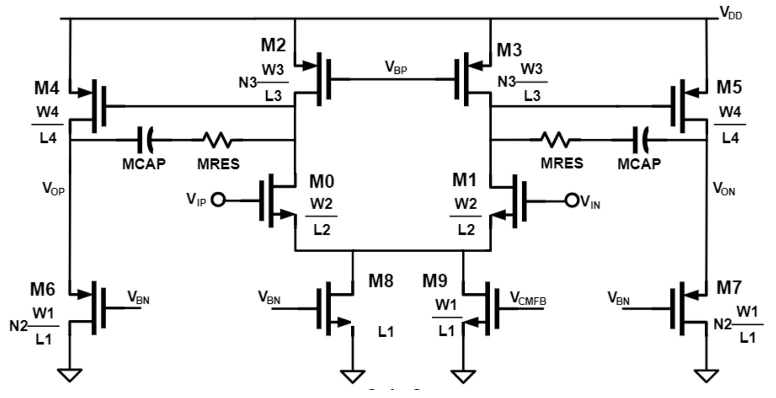
specifications

$$
\begin{aligned}
&\text{minimize } Power \\
&\text{s.t. } \text{DC Gain} > 60 \text{ dB} \\
&\quad \text{CMRR} > 80 \text{ dB} \\
&\quad \text{PSRR} > 80 \text{ dB} \\
&\quad \text{Output Swing} > 2.4 \text{ V} \\
&\quad \text{Output Noise} < 3 \times 10^{-4} \text{ V}_{\text{rms}} \\
&\quad \text{Phase Margin} > 60 \text{ deg} \\
&\quad \text{Unity Gain Frequency} > 40 \text{ MHz} \\
&\quad \text{Settling Time} < 3 \times 10^{-8} \text{ s} \\
&\quad \text{Static error} < 0.1 \\
&\quad \text{Saturation Margin} > 50 \text{ mV}
\end{aligned}
$$

design parameters & ranges

Parameters	LB	UB
L1 (μm)	0.18	2
L2 (μm)	0.18	2
⋮	⋮	⋮
W1 (μm)	0.22	150
W2(μm)	0.22	150
⋮	⋮	⋮
N3(integer)	1	20
N4(integer)	1	20

Topology



Miller OTA

What is the optimal sizing?

APOSTLE

APOSTLE leverages two aspects of sizing automation task to gain real-time advantage.

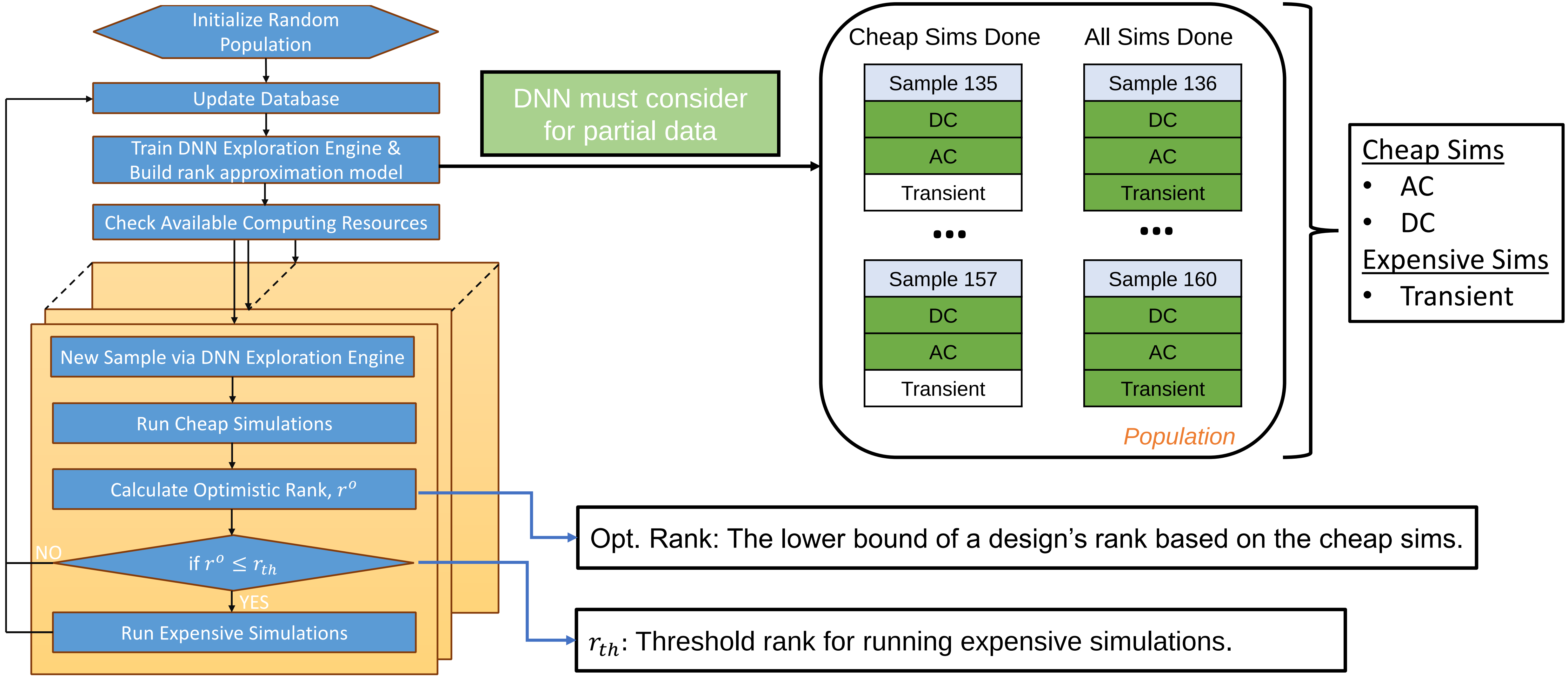
1) Analog optimization requires evaluation of different type of simulations such as ac, dc, transient, noise… and these have different computation cost.

   Idea: Evaluate cheap simulations first to make intermediate decisions.

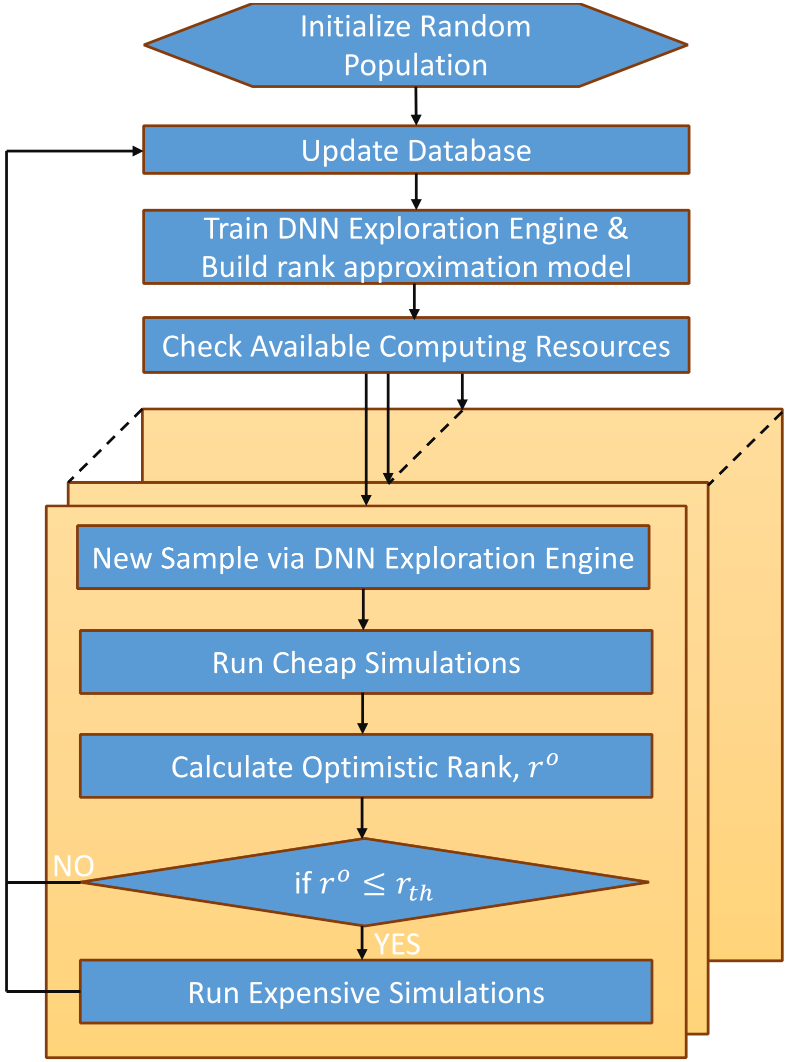
2) There could be more than one machine available at a time.

   Idea: Adapts a parallel framework instead of serial.

APOSTLE Framework

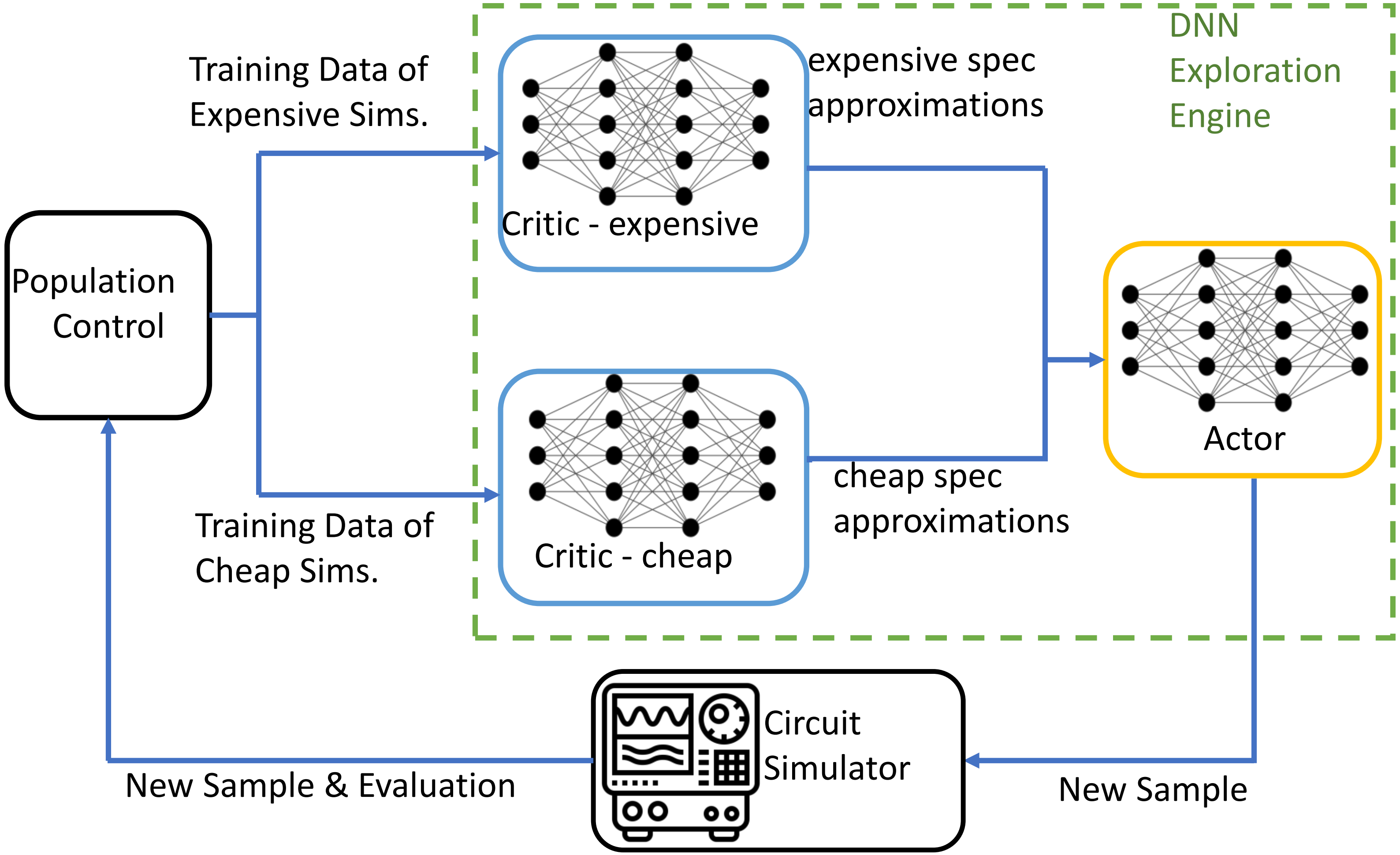


APOSTLE Contributions

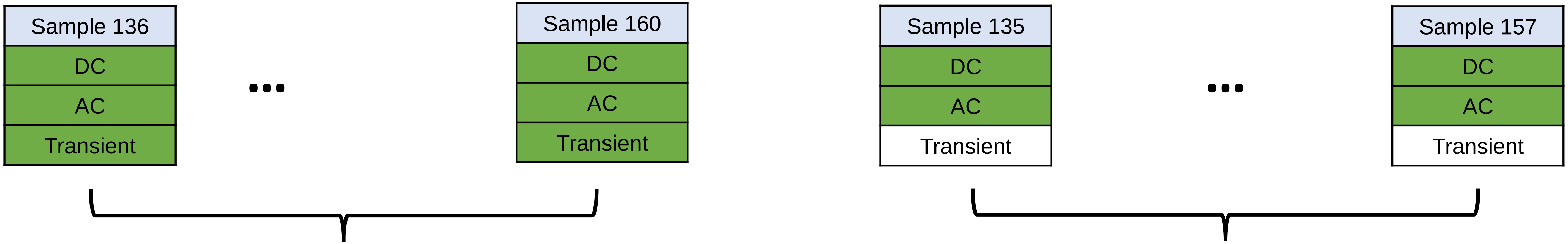


- DNN Exploration Engine with Missing Data
- Ranking Approximation Method
- Theory for Finding Threshold Rank
- Batch Optimization Algorithm

DNN Exploration Engine with Missing Data



Rank Approximation Method



Build regression model using fully evaluated samples and their ranks

Model should predict ranks based on the cheap evaluations.

Use Gaussian Process as the regression model. Input is the cheap specs, the output is the rank based on the full specs.

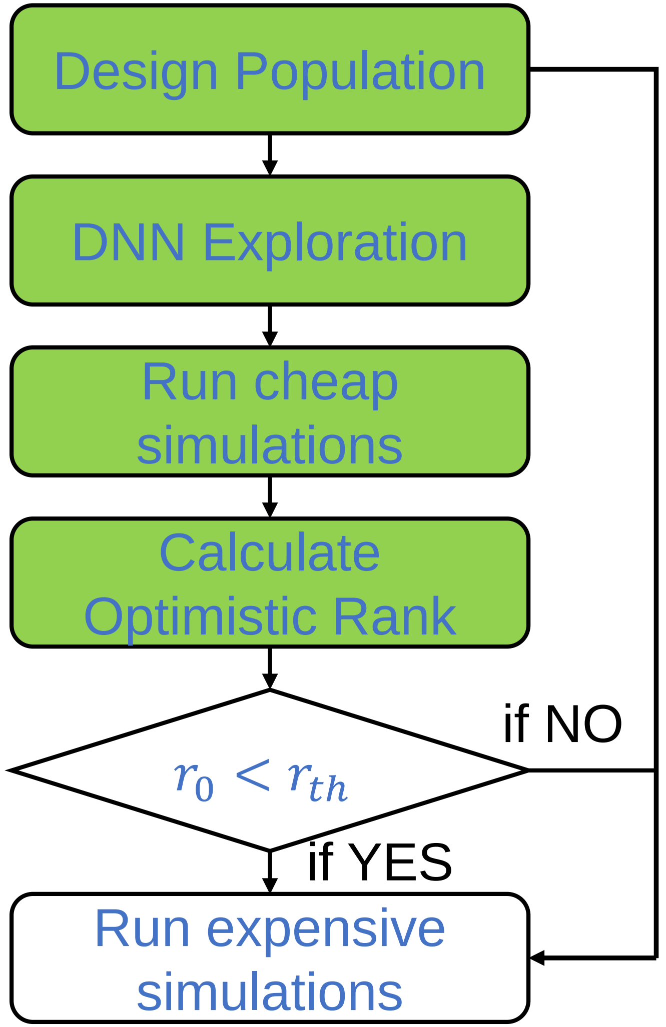
Build a model using only the information that will be available during the inference time.

hyperparameter

$$\mathbf{r}_i^{o} = \mu_i(y_{i,c}) - \alpha * \sigma(y_{i,c})$$

Optimistic rank   GP pred. mean   GP pred. std

Theory: Finding Threshold Rank



- What rank is worth to invest time for running expensive simulations.
  - r_{th} should be dependent on the cost of the expensive simulation: If not expensive at all, it should be relaxed; if too expensive, it should get lower, i.e., higher expectations before running expensive.
  - r_{th} should also be dependent on the average quality of new samples proposed by the DNN exploration strategy.

Theory: Finding Threshold Rank

Optimization Impact (OI)

$OI(r) = (\max(0,\ N_{es} - r))^{\beta}$

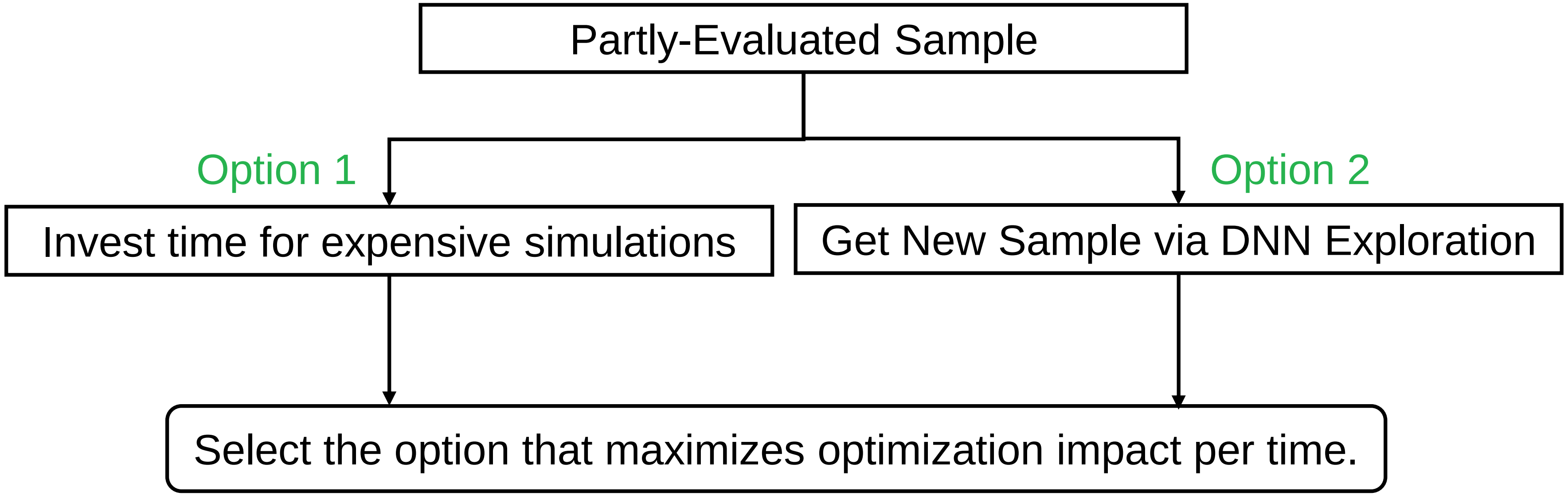
r: rank of a design

N_{es}: Population size

OI is larger if r is smaller

T_c and T_e: time cost of cheap & expensive simulations

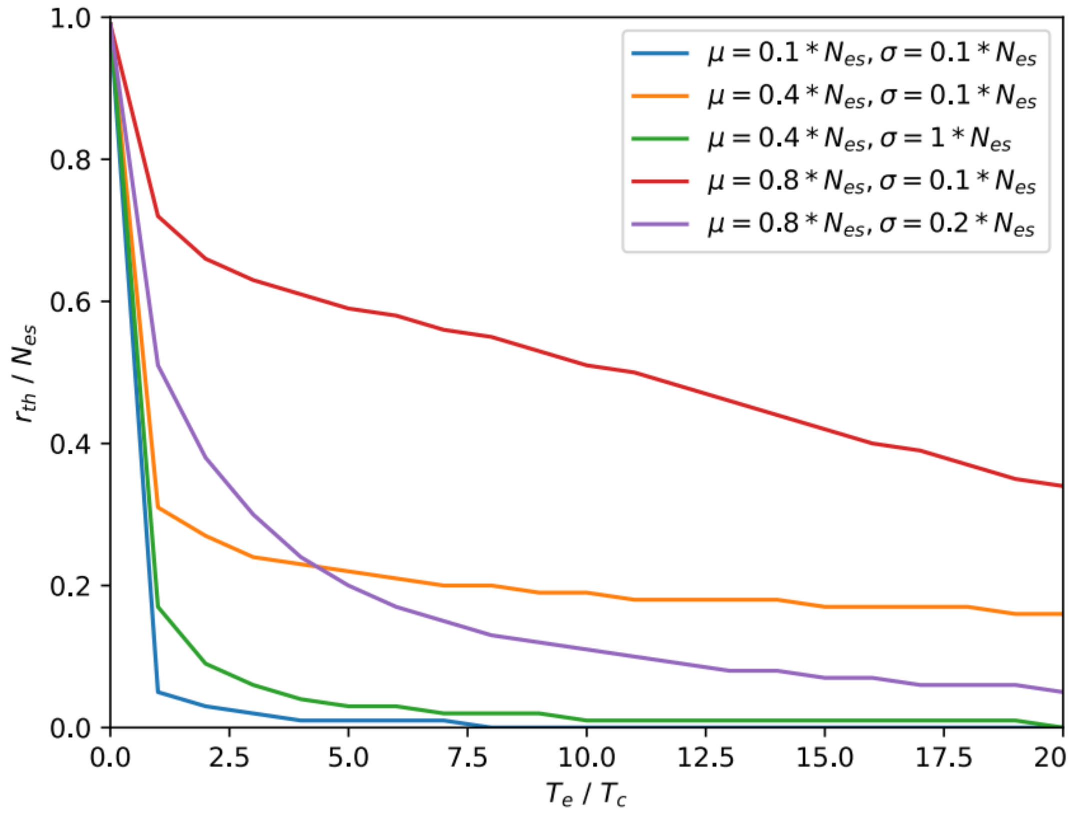
$T^{\star}$: expected time to draw a design with rank better than r*



$$r_{th} = \min_{r^\star} \left\{ r^\star \in \left(\frac{OI(r=r^\star)}{T_e} - \frac{OI(r|r \le r^\star)}{T^\star} \ge 0 \right) \right\}$$

if $r^o < r_{th}$ → Running expensive has greater expected OI per time.
if $r^o \ge r_{th}$ → Exploring a new sample has greater expected OI per time.

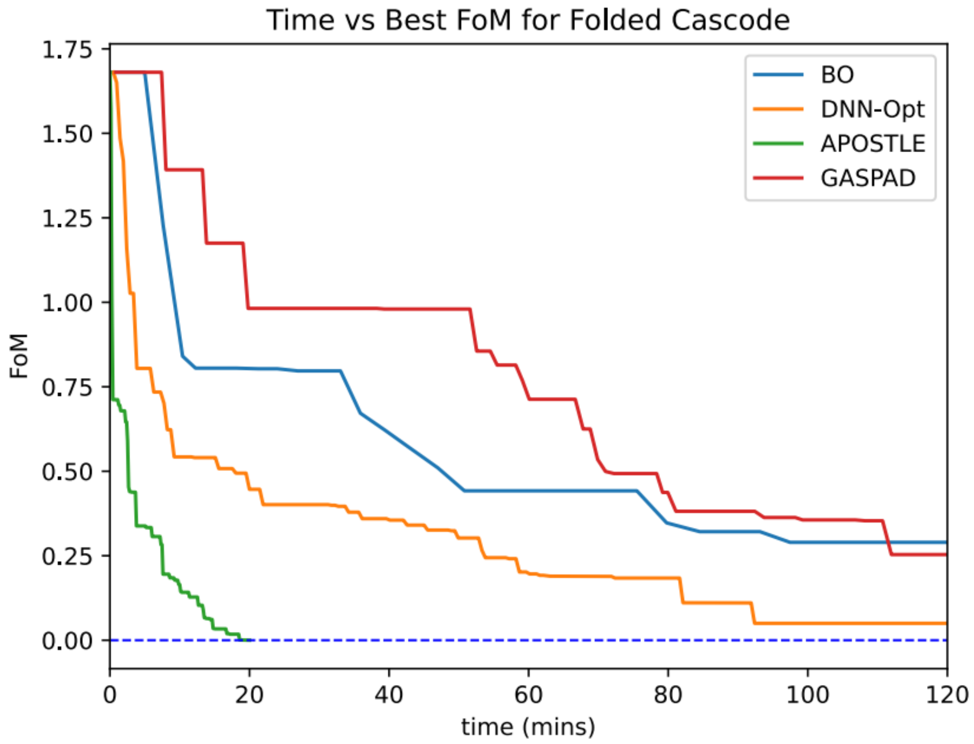
Theory: Finding Threshold Rank



- If expensive simulations comes for free, they are always run.
- As the cost of expensive simulations goes large, rth approximates to zero.
- If μ is small, rth goes small→ raised expectations for running expensive simulations.

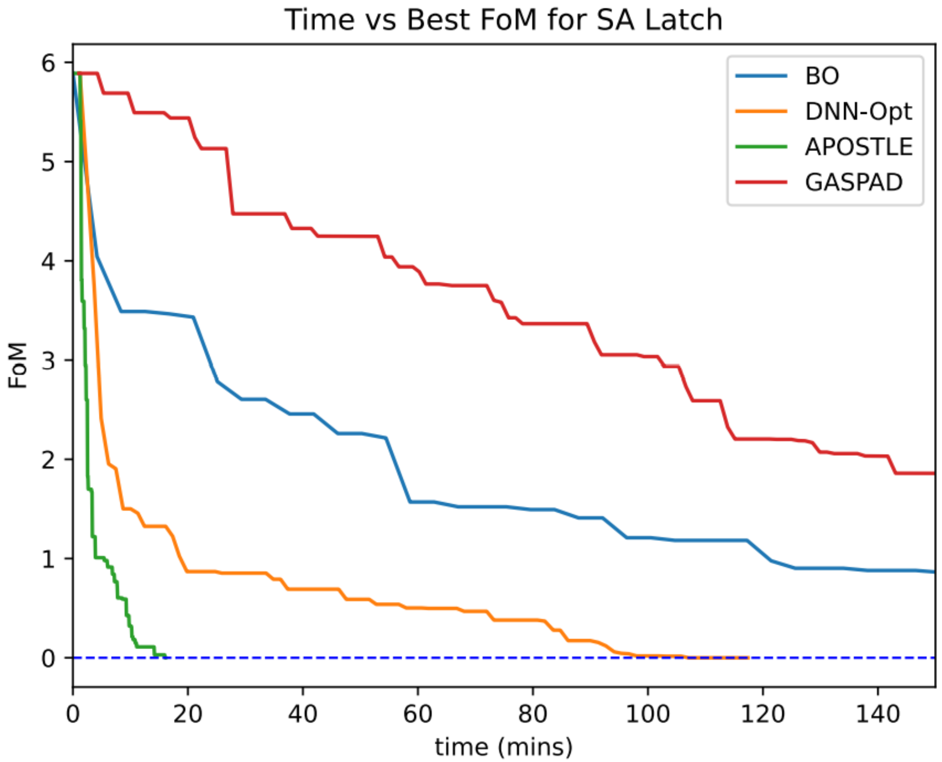
Experiments

- ▶ Folded Cascode OTA
  - ❑ 20 variables, 28 spec requirements



The average FoM (lower is better) curve w.r.t time in minutes

- ▶ Strong-Arm Latch Comparator
  - ❑ 13 variables, 10 spec requirements



The average FoM (lower is better) curve w.r.t time in minutes

Experiments: APOSTLE vs DNN-Opt

Testcase	FC OTA	SA Comp	PGA
# of samples APOSTLE	121	93	223
# of samples DNN-Opt	162	80	216
total time APOSTLE	18mins	14mins	3.7hrs
total time DNN-Opt	120mins	125mins	27hrs
objective val. APOSTLE	0.67 mW	2.6 μW	NA
objective val. DNN-Opt	1.3 mW	3.3 μW	NA
T_e/T_c	1.3	5	10
# of bypassed ES	11.8	32.3	54
% of bypassed ES	9.8%	31.6%	24.2%
Tot. CPU units APOSTLE	263	396	1913
Tot. CPU units DNN-Opt	372	480	2376

- The percentage overhead of visited designs by APOSTLE compared to DNN-Opt is between [−25%, +16%].
- APOSTLE's time efficiency varies between [6.7x to 9x], given Bmax = 8.
- APOSTLE reaches better objectives when given the same time.
- APOSTLE provided [17% to 30%] reduction in total CPU time which proves its efficiency only due to simulation skipping strategy.

THANK YOU!