Analog Circuit Generator based on Deep Neural Network enhanced Combinatorial Optimization

Conference Paper · June 2019

DOI: 10.1145/3316781.3322468

4 authors, including:

Kourosh Hakhamaneshi
University of California, Berkeley
7 PUBLICATIONS **21** CITATIONS
SEE PROFILE

Nick Werblun
University of California, Berkeley
3 PUBLICATIONS **13** CITATIONS
SEE PROFILE

Pieter Abbeel
University of California, Berkeley
385 PUBLICATIONS **27,484** CITATIONS
SEE PROFILE

Late Breaking Results: Analog Circuit Generator based on Deep Neural Network enhanced Combinatorial Optimization

Kourosh Hakhamaneshi, Nick Werblun, Pieter Abbeel, Vladimir Stojanović
kourosh_hakhamaneshi,nwerblun,vlada@berkeley.edu
pabbeel@cs.berkeley.edu
University of California Berkeley, USA

ABSTRACT

A deep neural network (DNN) based stochastic combinatorial optimization framework is presented that can find the optimal sizing of circuits in a sample-efficient manner. This sample efficiency allows us to unify this framework with generator-based tools like Berkeley Analog Generator (BAG) [1] to directly optimize layout, given the high level circuit specifications. We use this tool to design an optical link receiver layout, satisfying high-level design specifications, using post-layout simulations of only 348 design instances. Compared to an evolutionary algorithm without our DNN-based discriminator, our framework improves the sample efficiency and run time by more than 200x.

1 INTRODUCTION

Analog circuit design automation (ACDA) has always been a challenge, mostly because the analog design heavily depends on human expertise and intuition to guide the exploration of a complex multi-dimensional design space. Lack of cleanly labeled data sets prevents us from repeating the successes accomplished with supervised deep learning methods. Generating the relevant post-layout data is also expensive in terms of simulation time. Deep Reinforcement Learning (DRL) on the other hand, has been able to solve many human decision making problems [3] in an unsupervised manner, which creates new opportunities for ACDA as discussed in [4]. However, a lot of progress needs be made to make these DRL approaches scalable to real analog design problems. On the other hand, population-based circuit optimization approaches [2] have been demonstrated as flexible global optimization frameworks, but confined to small-size designs due to their sample inefficiency. They are not suitable for layout-level optimization, which requires long simulation time even for relatively small circuits.

In advanced technology nodes where wire capacitance dominates the circuit capacitive loads, the gap between schematic and post-layout circuit simulation results widens, forcing designers toward layout-aware design methodologies like BAG [1]. In BAG, designers formulate their design strategy as a parameterized layout of the circuit cells in a technology-agnostic way, allowing efficient and accurate layout generation and design-space exploration. The complexity of the post-layout simulations, however, creates additional challenges for the optimization tools that can be developed to automate the design process.

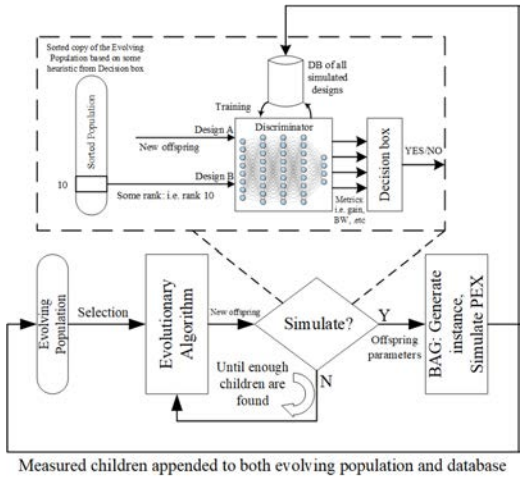

Figure 1: Outline of framework. Note that training is done with all possible $\{(d1, d2)|d1, d2 \in DB\}$ as inputs and their corresponding comparison in each metric as output labels.

In this paper, we tackle the sample efficiency problem of population-based methods using DNNs. We present a framework that is built on top of an evolutionary algorithm (EA) and leverages a DNN trained as a discriminator to prune out "bad" children to reduce the evaluation time spent in post-layout simulation. Utilizing the example of a full optical receiver front-end designed in GF14nm process we demonstrate that this method is effective on a relatively-large design space and can handle state-of-the-art designs within the layout constraints of an advanced process node.

2 FRAMEWORK IMPLEMENTATION

Figure 1 shows the structure of the proposed framework. We define a cost function for each design as:

$$cost(x) = \Sigma_i w_i p_i(x) \quad (1)$$

where $p_i(x) = \frac{|c_i - c_i^*|}{c_i + c_i^*}$ (normalized specification error) for a design that does not satisfy constraint c_i^*, and zero if it does. The goal of the optimization is to minimize this cost function toward zero cost (feasibility).

In each iteration we utilize the EA to create population candidates (new offspring). We then use a DNN to predict the quality of these new offspring and only simulate the samples that are classified as good design points. The DNN takes two designs as input and

DAC '19, June 2–6, 2019, Las Vegas, NV, USA

ACM ISBN 978-1-4503-6725-7/19/06…$15.00
https://doi.org/10.1145/3316781.3322468

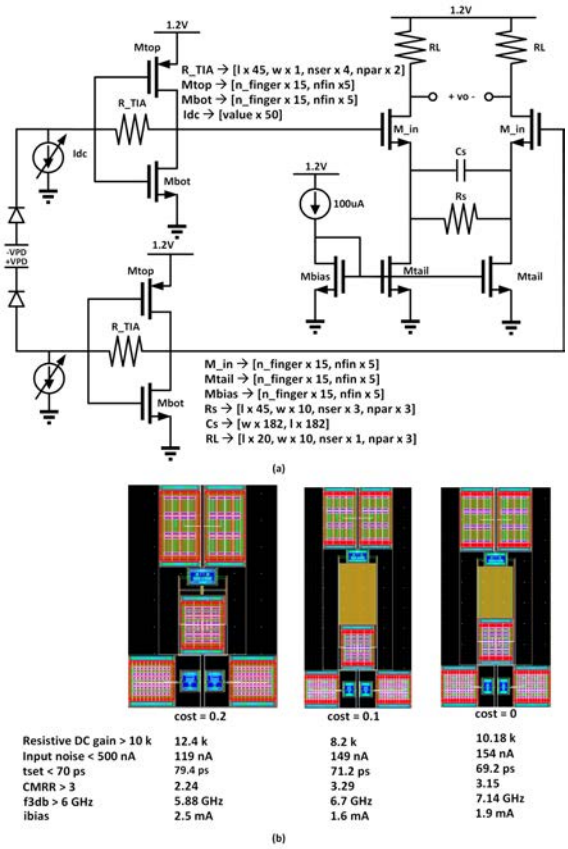

Figure 2: (a) Schematic and design parameters with the search range dimension for each (b) The evolution of layouts over time and their performance

predicts whether Design A is better than Design B in each individual metric (i.e. gain, bandwidth, etc.). A reference design is chosen by a ranking heuristic (imposed by Decision box) from the previous population and is used as Design B to the input of DNN. Each new offspring will be fed in as Design A. Based on the DNN classification outputs the Decision box will decide whether to simulate or ignore Design A, utilizing a heuristic, that determines which metrics to prioritize first.

During training, the DNN is continuously trained on all pairs of previously simulated designs for which we have the ground truths. At initialization, a set of random design samples are generated and simulated after layout extraction and the DNN is trained on all tuples from that sample set. The initial population is also used as the initial evolving population for the EA. The evolving population will go through some evolutionary and selection operations with the children curated by DNN. The optimization stops when a predetermined maximum number of evolution steps is reached. In the next section we will apply this approach to design an optimized optical link receiver layout instance.

3 EXPERIMENTS

We conducted an experiment of an end-to-end realization of a differential optical link receiver front-end operating in GF14nm technology. The layout, schematic and testbench generators were prepared using BAG. The framework was then used to find the design instance that satisfies some difficult specifications, like those shown in Figure 2. In terms of layout generator search space, each resistance drawn in Figure 2 has unit length, unit width, number of series units, and number of parallel units. The CTLE's capacitor has width and length, and each transistor has number of fins and number of fingers that need to be determined. The cardinality of each design parameters is annotated in Figure 2(a). For example Mtop, has 15 values of n_fingers and 5 values for nfins considered. Overall, the design example has a 21 dimensional exploration space with size of 1.7×10^{22}. Each evaluation includes a layout and schematic instance generation, LVS, RC extraction, testbench generation and characterization of the circuit, which takes roughly 78 seconds per design instance on a state-of-the-art compute server.

Table 1: Runtime breakdown.

	Our work		Just Evolution	
	n	time [min]	n	time [min]
Init pop generation	100	130	100	130
Training the DNN	50	14.9	-	-
Queries to the DNN	75031	256	-	-
Simulation	248	**295**	75031	**62 days**
Total	-	**7.1 hrs**	-	**62 days**

Figure 2(b) shows three distinct layouts with decreasing cost progression from 0.2 to 0, where all specs are satisfied. The performance metrics are displayed for each layout accordingly. We first generate 100 samples randomly. The minimum cost in this initial set is 0.52. For 50 iterations we train the DNN, generate the children, query the DNN and run simulations for those which were classified as better designs compared to the rank 10 design in the previous population in terms of the heuristic from Decision Box. The accumulated time (minutes), and number of samples (n) in each part is shown in Table 1. If we had just ran the evolution (without the DNN discriminator) we would have had to simulate a total of 75031 designs (equivalent to 62 days of runtime). Using our approach the design with zero cost was achieved using only 338 post extracted evaluations (7.1 hours on the same system - a more than a 200x runtime improvement). Through this example, the proposed approach demonstrates the potential to tackle complex analog and mixed-signal design problems.

REFERENCES

[1] Chang, E., Han, J., Bae, W., Wang, Z., Narevsky, N., NikoliC, B., and Alon, E. BAG2: A process-portable framework for generator-based AMS circuit design. In *2018 IEEE Custom Integrated Circuits Conference (CICC)* (San Diego, CA, Apr. 2018), IEEE, pp. 1–8.

[2] Prajapati, P. P., and Shah, M. V. Two stage CMOS operational amplifier design using particle swarm optimization algorithm. IEEE, pp. 1–5.

[3] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. Mastering the game of Go without human knowledge. *Nature 550*, 7676 (Oct. 2017), 354–359.

[4] Wang, H., Yang, J., Lee, H.-S., and Han, S. Learning to Design Circuits. *arXiv:1812.02734 [cs]* (Dec. 2018). arXiv: 1812.02734.