Transformer-Guided Mutation in Cartesian Genetic Programming for Approximate Multiplier Design

Author: Ondřej Galeta

Supervisor: prof. Ing. Lukáš Sekanina, Ph.D.

Standard Approach & Motivation

- Cartesian Genetic Programming (CGP) is an evolutionary algorithm used to automatically evolve combinational circuits.
- Each iteration (generation) follows 3 stages: evaluation, selection, and mutation.
- Evaluation is made according to fitness function represented by Eq. 1.
- (λ + 1) strategy: start with 1 parent, generate λ offspring, and pick the best from parent + offspring.
- The mutation operator performs small randomized changes in the chromosome that lead to the creation of a new one.

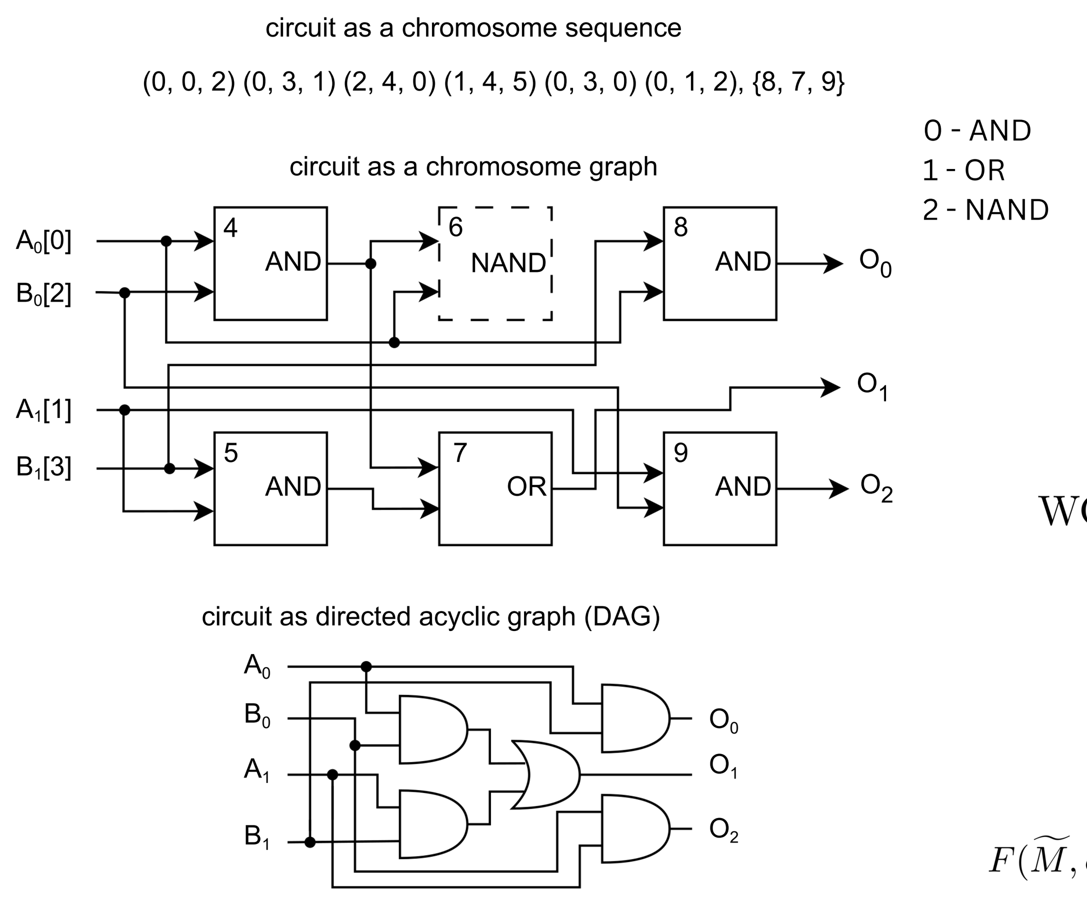


Fig. 1: Different representations of the same circuit in Cartesian Genetic Programming (CGP).

$$\mathrm{WCE} = \max_{\forall i} \left| O^{(i)}_{\mathrm{approx}} - O^{(i)}_{\mathrm{orig}} \right|$$

Eq. 3: Worst Case Error.

$$F(\widetilde{M}, \epsilon) = \begin{cases} cost(\widetilde{M}) & \text{if } \mathrm{WCE}(\widetilde{M}) \leq \epsilon \ \wedge \\ & \mathrm{WCE}_{zr}(\widetilde{M}) = 0 \\ \infty & \text{otherwise.} \end{cases}$$

Eq. 1: Fitness function [2].

Works well but has scalability issues, as it converges slowly to the optimum.

Proposed Model

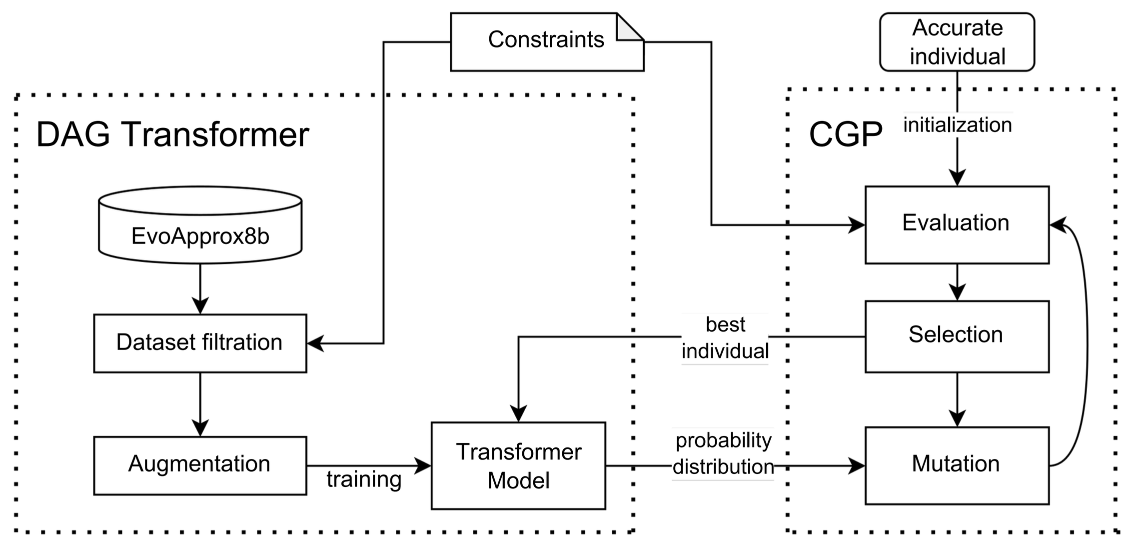
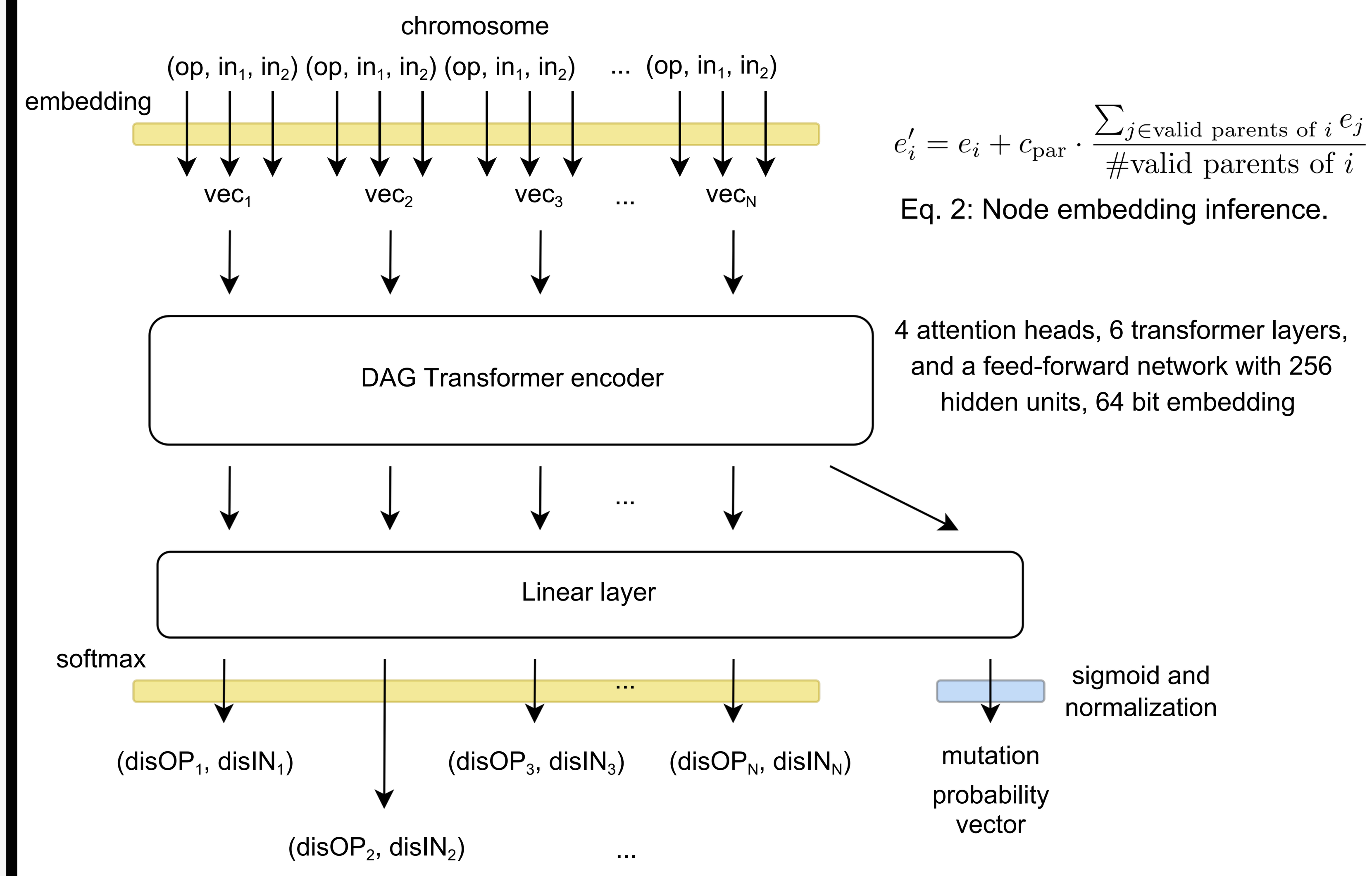


Fig. 3: Transformer training and incorporating it to the CGP loop.

- Created BERT-based transformer mutation operator.
- Model provides a probability distribution that is followed by a mutation operator.
- Developed algorithm that switches proposed and standard mutation operator to increase stability and performance.



$$e'_i = e_i + c_{\mathrm{par}} \cdot \frac{\sum_{j \in \text{valid parents of } i} e_j}{\#\text{valid parents of } i}$$

Eq. 2: Node embedding inference.

4 attention heads, 6 transformer layers, and a feed-forward network with 256 hidden units, 64 bit embedding

Fig. 2: Proposed transformer-based mutation operator model.

Model Training

- Dataset of approximate multipliers from EvoApproxLib [1]
- From the training set are filtered all multipliers that have a fitness value of infinity according to Eq. 1
- Constructed a Pareto curve by optimizing with respect to WCE and area.
- The distance from the Pareto curve influences the learning process by determining the importance of each multiplier in the dataset.

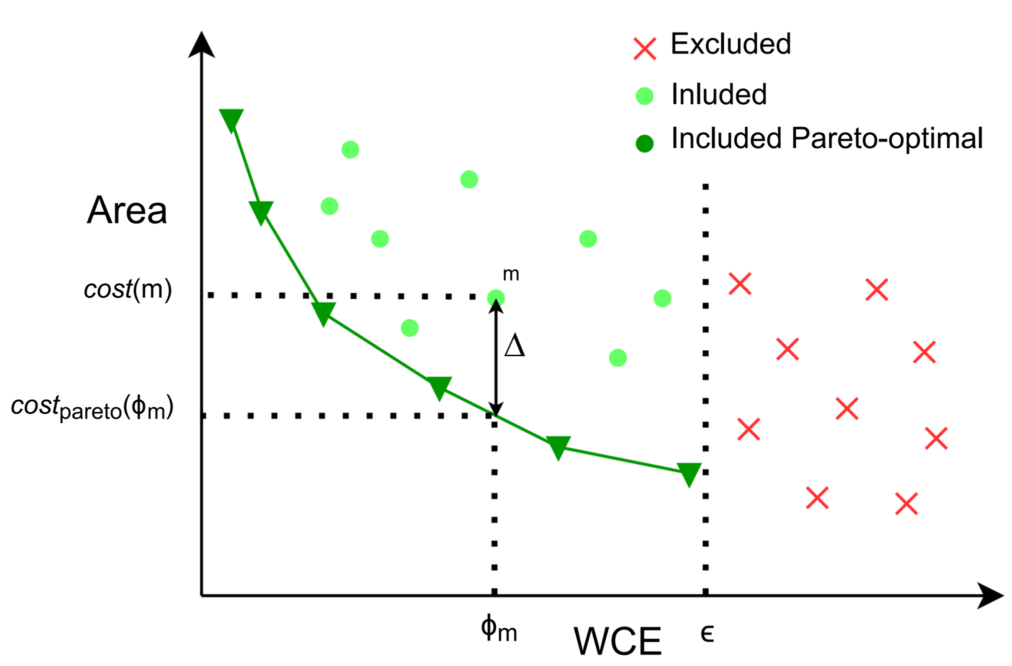
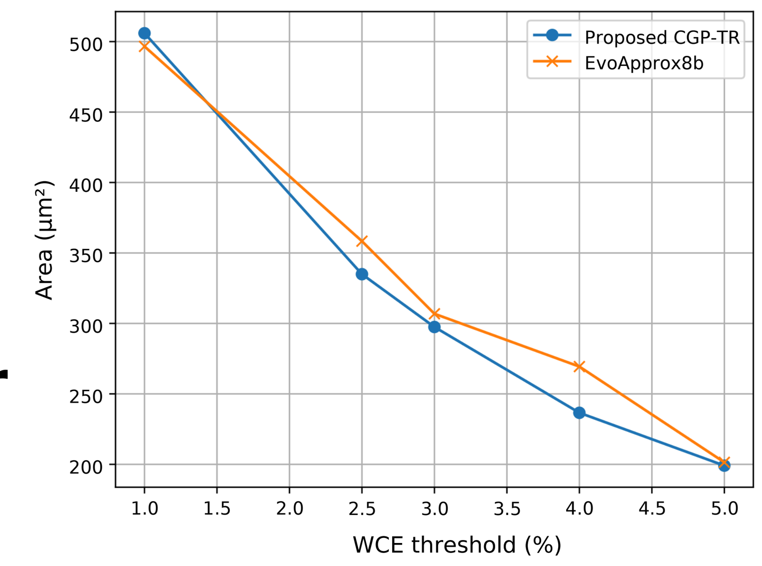


Fig. 4: The principle of multiplier selection for transformer model training.

Resulting Approximate Multipliers

Fig. 5: WCE and area of the best approximate 8-bit multipliers from EvoApproxLib [1] and those evolved using CGP-TR.



A smart mutation operator makes CGP converge faster and finds new, potentially better designs.

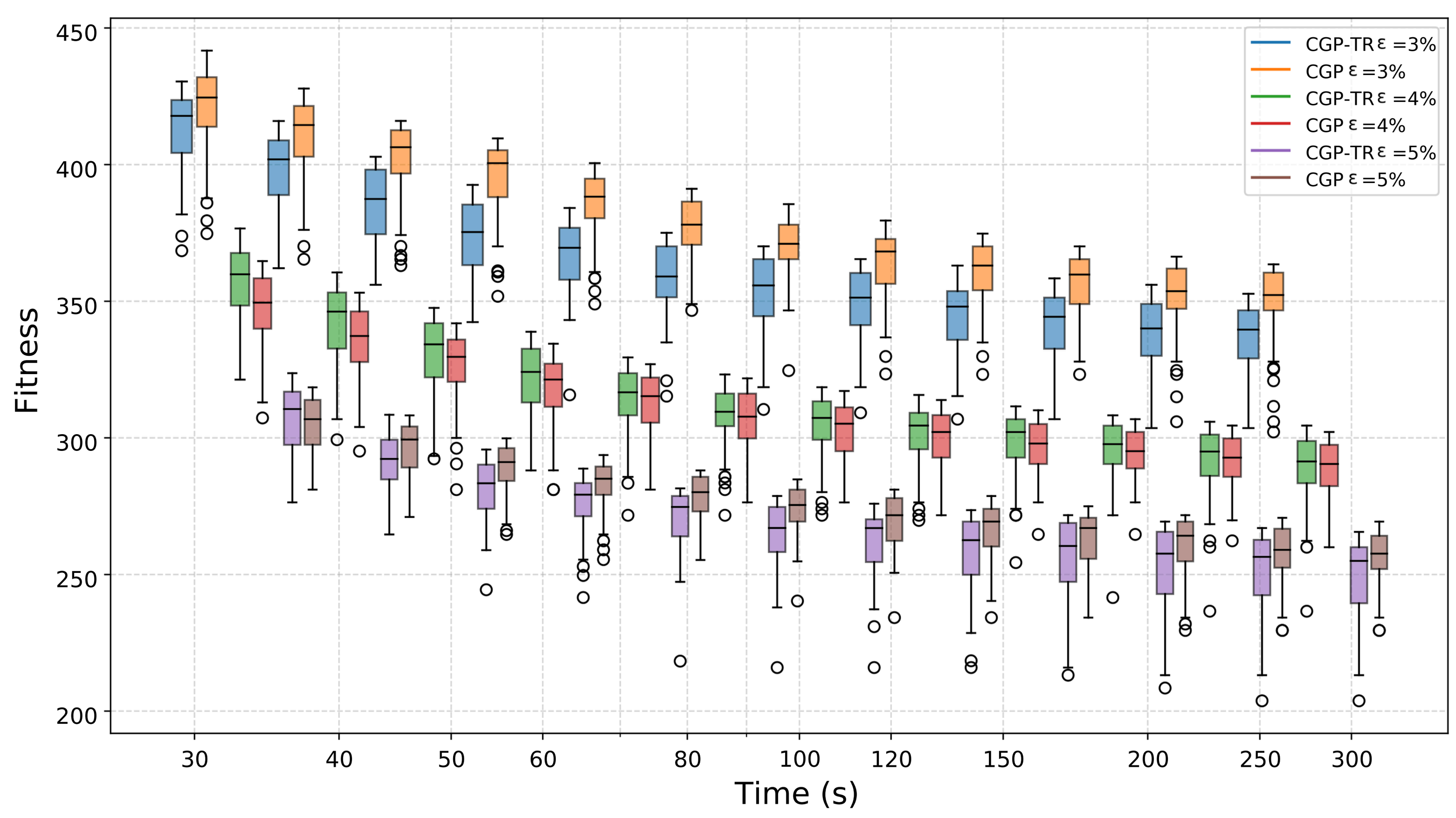


Fig. 6: Fitness progress in CGP utilizing standard mutation (CGP) and CGP with the transformer based mutation operator (CGP TR) for three different WCE thresholds ϵ.

[1] V. Mrazek, R. Hrbacek, et al. Evoapprox8b: Library of approximate adders and multipliers for circuit design and benchmarking of approximation methods. In Proc. of DATE'17, pages 258–261, 2017.
[2] Vojtech Mrazek, Lukas Sekanina, and Zdenek Vasicek. Libraries of approximate circuits: Automated design and application in cnn accelerators. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 10(4):406–418, 2020.

Excel@FIT 2026

BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY