

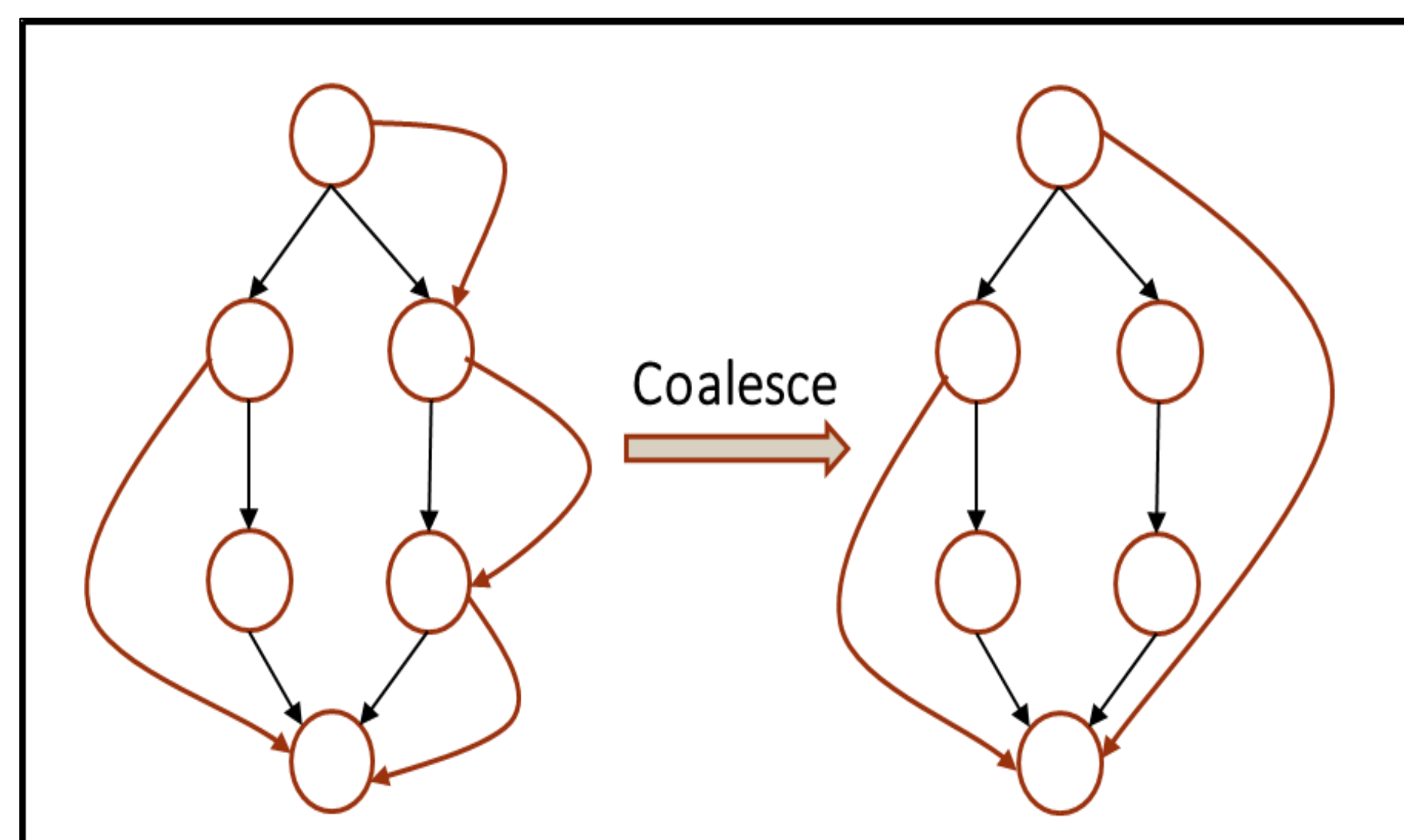
# Towards Inductive Synthesis of Compiler Heuristics: A Case Study with Register Allocation

Mohammad Ali  
Texas State University  
San Marcos, TX  
mohammad.ali@txstate.edu

Apan Qasem  
Texas State University  
San Marcos, TX  
apan@txstate.edu

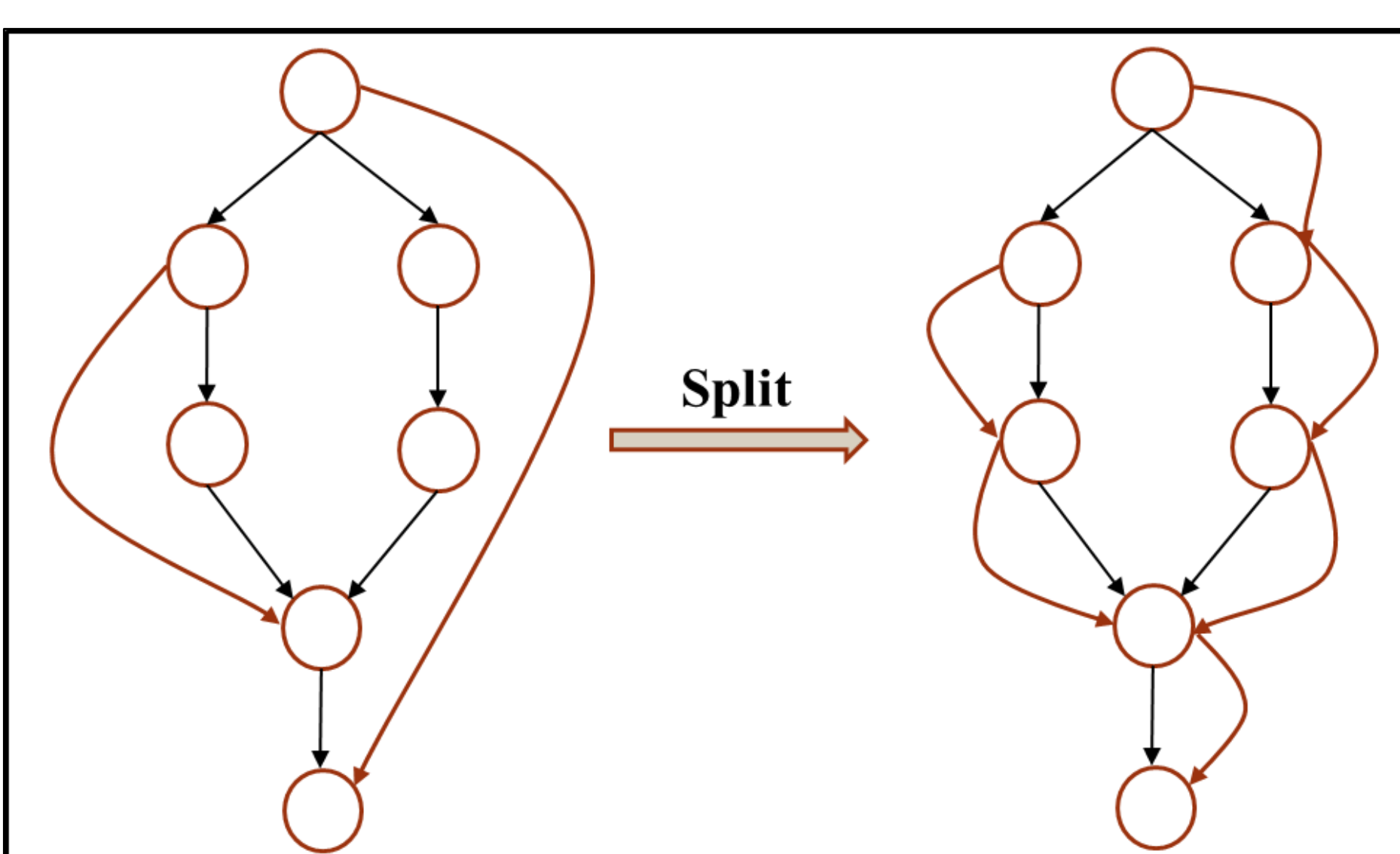
## Motivation

- In the compiler world, many graph-based algorithms are used for data-flow analysis and code optimization
- Most of the heuristics designed to solve the data-flow analysis and code optimization problems are focused on specific architectures and application domains
- In this work, we are trying to develop an automated method for creating optimal heuristics using Inductive Program Synthesis
- Given a program representation and a specification of an optimization task as input, generate an algorithm for that optimization that maximizes performance



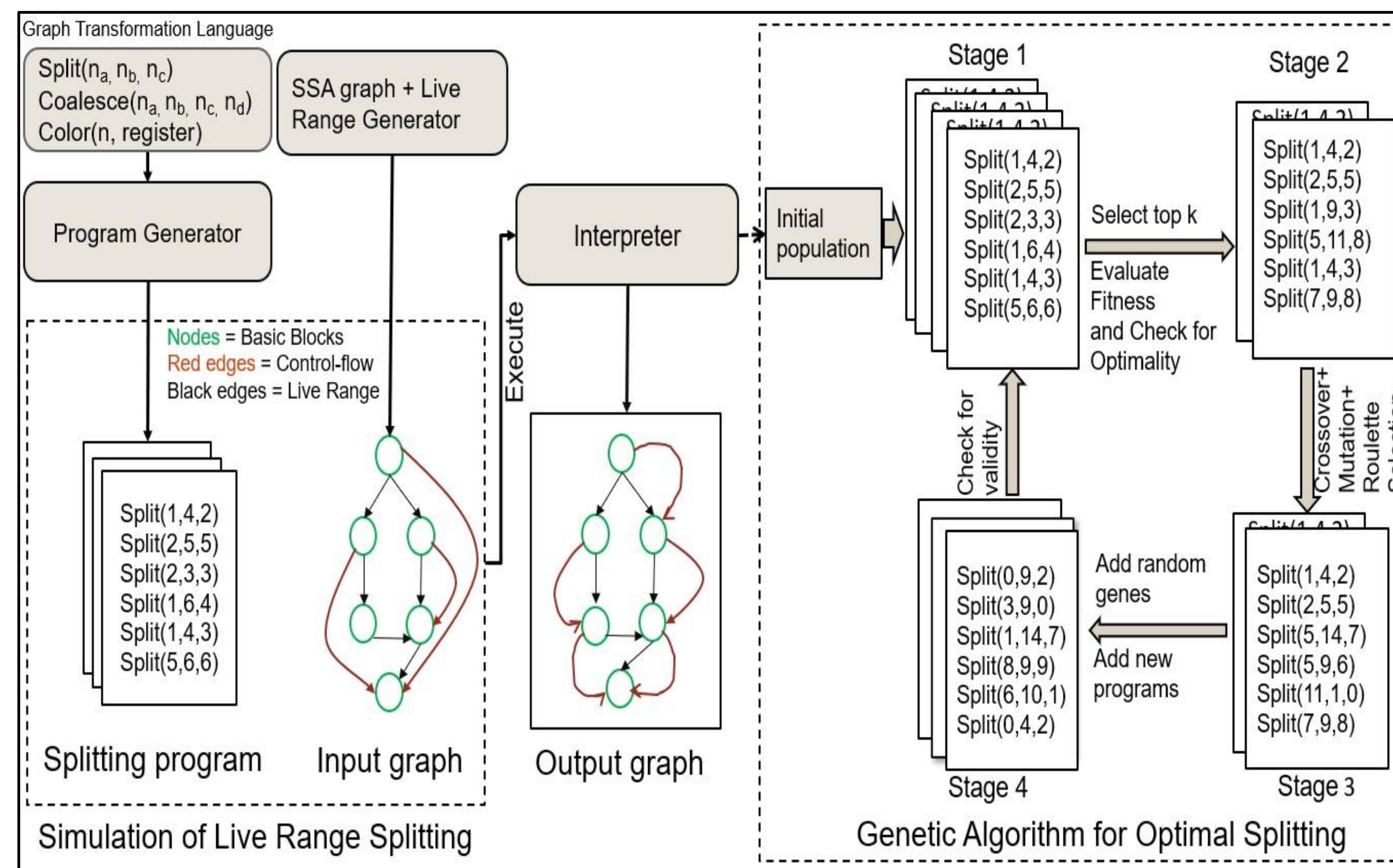
## Background

- This work focuses on the register allocation problem which employs two graph transformation algorithms
  - Splitting and coalescing of the SSA Live Range Graph
  - Coloring of the interference graph



## Methodology

- Input Generator: generates random SSA Graphs with different Live Ranges
- GTL: a language to express graph transformations for Live Range analysis and graph coloring; for this study, only the split operation is used
- Program Generator: generates random programs from GTL
- Interpreter: interprets programs written in GTL

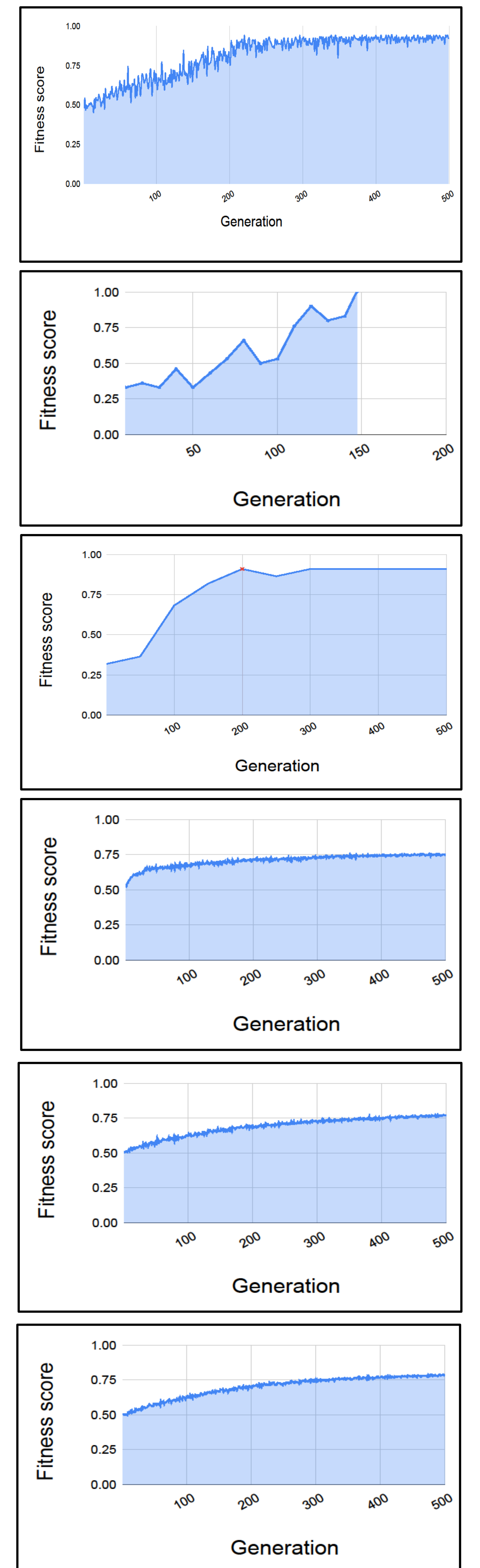


- We use genetic algorithms to produce a graph transformation algorithm that minimizes the amount of spill code in an input graph
- Cross-over, mutation, and roulette selection are applied to genomes which consist of a sequence of split operations
- The fitness score is evaluated to evolve the initial batch of genomes to the optimal solution

## Conclusion and Future Work

- Inductive Program Synthesis can be a viable solution to challenging problems in the compiler domain
- The structure of the input Live Range graph can have a significant impact on convergence time
- Due to the time-intensive nature of the Genetic Algorithm, we are considering the replacement of the Genetic Algorithm with more sophisticated techniques such as the Multi-Armed Bandit

## Experimental Results



## Acknowledgment

This work was supported by the National Science Foundation through award OAC-1829644, and equipment grants by AMD and NVIDIA