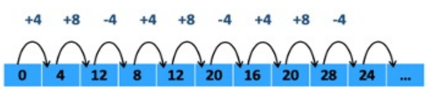
EECS 592: Memory Access Pattern Recognition Using Search Algorithms

ELECTRICAL ENGINEERING & COMPUTER SCIENCE
UNIVERSITY OF MICHIGAN

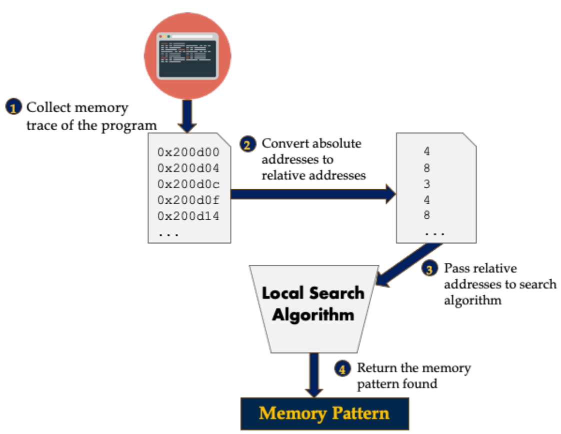
Nada Abdalgawad, Tersiteab Adem, Meron Zerihun Demissie

Introduction

- Memory access latency hinders system performance.
- Identifying the memory access pattern can help:
  - Prefetch data ahead of time
  - Identify memory faults before execution
- Assumption: the pattern is repetitive and continuous
- Implements search algorithms to identify these patterns
- Focus on relative memory access rather than absolute



High-level Approach



Methodology

Data Preprocessing

- **Synthetic Dataset**
  - Created for evaluating the algorithms
- **Real world Benchmarks**
  - Programs with **static** dataflow
  - VIP-Bench: Bubble sort and Flood fill
  - Memory trace collected using Intel Pin and translated to relative memory addresses

Objective Function

Mean Absolute Error (MAE)
Average difference between a pattern and a memory trace

Mean Misprediction
Average number of mismatches between the pattern and memory trace

Unmatched Entries Count
Total number of unmatched memory entries

Best-fit pattern: no redundancy, objective function evaluates to **zero**

Hill Climbing

Steepest Descent: minimize the objective function (find a pattern that produces the **smallest difference (MAE)** over a memory trace)

A) Initial State Instantiation

Initial state: **half** of the memory entries

Helped **explore more states** (more **down-hill** moves) but returned a pattern with redundant entries

Solution: Random-restart variant to explore more states and eventually find the best-fit pattern

Drawback: Takes a long time to find the best-fit pattern

B) Objective Function Exploration

Find an objective function that **strictly decreases** until the best-fit pattern is found

Solution: Count the number of **unmatched entries**

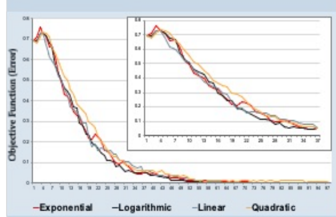
Very fast: function returns as soon as it finds entries that don't match

Takes **less steps** to find the best-fit pattern

Simulated Annealing

- Steepest Descent while **allowing** uphill movement with gradually **diminishing probability**

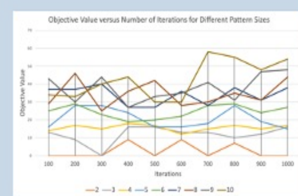
- Parameters to tune:
  - **Cooling schedule**: exploration vs exploitation
    - *Exponential*
    - *Logarithmic*
    - *Linear*
    - *Quadratic*
  - **Maximum Number of iterations**



Genetic Algorithm

- At each iteration:
  - A pattern size is randomly picked.
  - Population is initialized by randomly
  - In each step:
    - Pick 2 individuals randomly based on fitness function
    - Crossover the individuals → child
    - Child is mutated

- Parameters to tune:
  - **Maximum number of iterations**
  - **Number of individuals in the population**
  - **Number of steps**

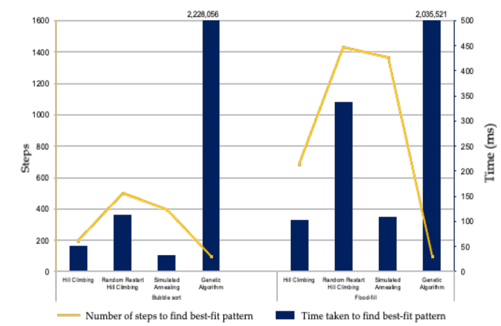


Results

Evaluation Workloads

Benchmark	Input Size	Pattern size
Bubble-sort	19603	198
Flood-fill	16392	683

	Hill Climbing	Simulated Annealing	Genetic Algorithm
Finds correct solution pattern	Yes	Yes	No
Is smallest pattern?	Yes	Yes	-
Time Complexity	O(n)	O(n log(n))	O(gnm)
Space Complexity	O(n)	O(n)	O(nm)



Conclusion and Future work

- **Conclusion**
  - Hill Climbing (HC) and Simulated Annealing (SA) algorithms were able to find best-fit patterns
  - Genetic Algorithm is not fit for this problem
  - For bubble-sort, SA was 1.6x faster than HC but took 2x more steps.
  - For flood-fill, HC was 1.06x faster than SA and took 2x fewer steps. Simulated Annealing took 2x more steps and was 1.6x faster than Hill Climbing for Bubble-sort
- **Future Work**
  - Improve Objective function
  - Handle traces with multiple patterns
  - Consider different ways of representing memory traces
  - Implement more search algorithms
  - Expand the read dataset

Significant References

- Peter Braun and Heiner Litz. Understanding memory access patterns for prefetching. 2019.
- Lauren Biernacki, Meron Zerihun Demissie, et al. Vip-bench: A benchmark suite for evaluating privacy-enhanced computation frameworks. In 2021 International Symposium on Secure and Private Execution Environment Design (SEED), pages 139–149, 2021

COLLEGE OF ENGINEERING
ELECTRICAL ENGINEERING & COMPUTER SCIENCE
UNIVERSITY OF MICHIGAN