TESTING VISION-BASED CONTROL SYSTEMS USING LEARNABLE EVOLUTIONARY ALGORITHMS

AUTHORS: RAJA BEN ABDESSALEM, SHIVA NEJATI, LIONEL C. BRIAND

PRESENTER: KELVIN MOCK

DATE: 24TH OCT 2025

AGENDA

Introduction

- Motivation
- Problem Statement
- Research Questions

Methodologies

- Algorithms
- Experiment Setup & Metrics
- Evaluation Outcomes

Critique

- Soundness
- Significance
- Novelty
- Verifiability & Transparency

Conclusion + Future Work

INTRODUCTION

WHAT IS THE PROBLEM? WHY IS IT SIGNIFICANT?

MOTIVATION

- AI-Based Systems are becoming popular!
- They take over systems that control your safety
  - Critical Infrastructure
  - Healthcare Systems
- Computer Vision is one of their capabilities!
- This Study focuses on **Vision-Based Control Systems** → autonomous vehicles
- Automated Emergency Braking (AEB) System
- Efficient Testing using Evolutionary Search + ML

PROBLEM STATEMENT

Representation

- Advanced Driver Assistance Systems (ADAS)
- Simulated Environment: $\langle S, O, I, D, C \rangle$
- **Static** Objects (×4) – road types, weather types
- **Dynamic** Objects (×5) – pedestrians, other vehicles
- **Initial** States of Mobile Objects – velocity / position
- **Domain** – what are there at the start?
- Boolean **Constraints** – Level of Fog, Visibility Range

Genetic Operators

- Phases: Selection, Crossover, Mutation

Fitness

- Multi-Objective
- Based on ADAS Simulation Outputs (for testing)
  - Positions: Vector<Vehicles, Pedestrians>
  - Time-To-Collision: $\mathbb{R}$
  - Certainty Of Detection: $\mathbb{R}$
  - Braking: Boolean
- Metric: Euclidean Distance
  - Minimize distance = Pedestrian $\leftrightarrow$ Vehicle
  - Maximize Current Vehicle's Speed
  - Maximize Certainty of Detection: seeing a pedestrian

RESEARCH QUESTIONS

1. How is ML being used in the loop to guide the evolutionary search? Is it effective enough?

2. Does it help characterize and converge towards homogeneous critical regions (high-risk scenarios)?

METHODOLOGIES

HOW DOES THE PAPER ADDRESS THE PROBLEM? EFFICIENT?

ALGORITHMS OVERVIEW

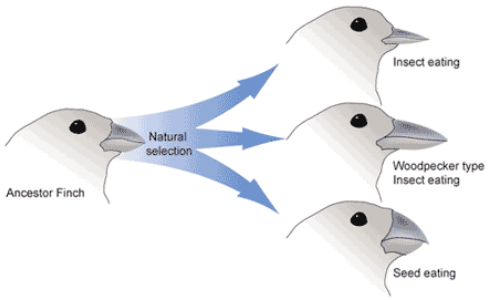
- Starts with a **Non-Dominated Sorting Genetic Algorithm** version 2 (NSGA-II) as the baseline
- Generates an initial population with the PLEDGE tool *(Christopher et al, 2013)*
  - t-wise combinatorial testing *(Kuhn et al., 2004)*
  - Maximizing the pairwise coverage of static variables
- Generates a set of solutions → Pareto nondominant-front
- Identifies the best solution based on the best Pareto front rank
- Guided by a **decision tree** classifier
  - Learns from past runs where failures produced

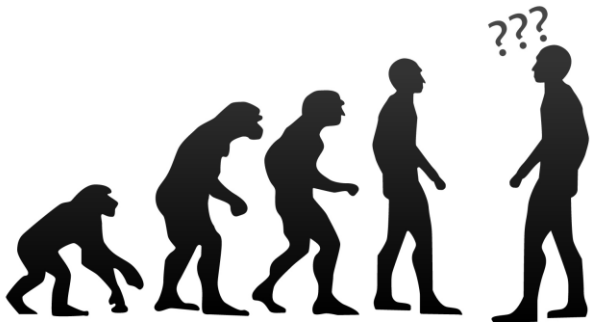
[DEFINITION]
A **dominance relation** over solutions is defined as follows: A solution x dominates another solution y if x is not worse than y in all fitness values, and x is strictly better than y in at least one fitness value.

ALGORITHMS – OPERATORS

Selection

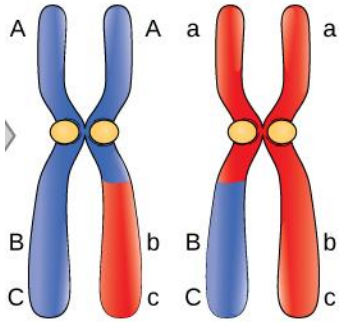
- Binary Tournament
- Random Selection With Replacement
- Forms a standard in NSGA-II algorithm





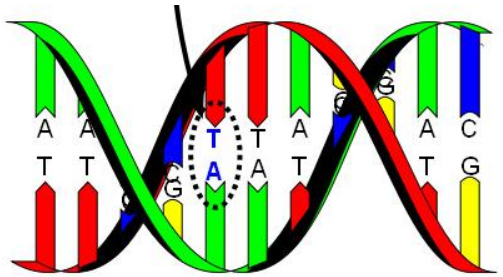
Crossover

- Simulated Binary Crossover (SBX)
- Applies only to dynamic variables (e.g., their speed)



Mutation

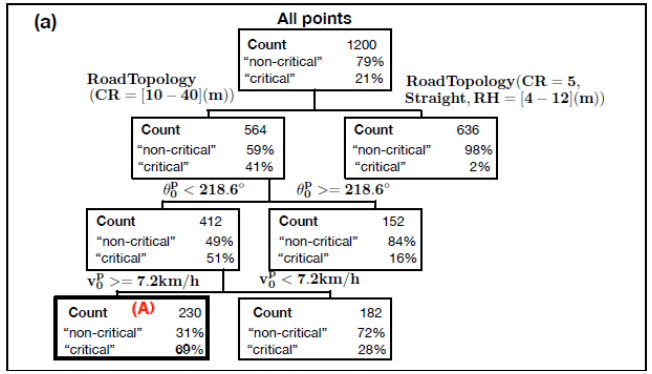
- Random Mutation on ALL static + dynamic variables
- Mutate Under Constraints – the Boolean set



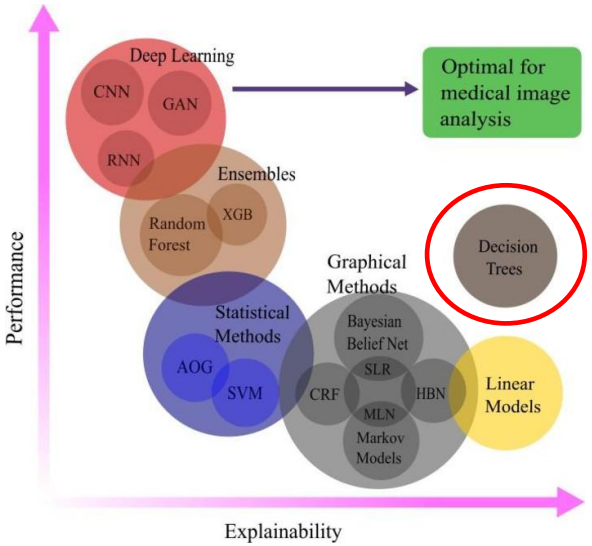
ALGORITHMS – DECISION TREE

- Supervised Binary Classification
- Boolean Predicate Functions to label ADAS scenarios – critical (in danger) or not

$$CB(U,V) = (F_1(u_1,u_2) < 50cm) \bigwedge (v_2 > 0.5) \bigwedge (F_2(u_1,u_2) > 30\ km\ h^{-1})$$



- Partitioning a set of **labelled** test scenarios in a **stepwise** manner
- Enhancement to characterize $CB(U,V)$ as a Real-Valued Function → Regression Tree
- ✓ Understandable (Explainable) by practitioners *(Tribikram Dhar et al, 2023)*
- ✓ Not only shows exact criticality, but also their associated road conditions!
- Generalization Issue – Overfitting: **Stopping Criterion** to control tree expansion
- Efficiency: Reuse Simulations given expensive computations



EXPERIMENT SETUP

Parameter	Value
Population Size	100
Crossover Rate	0.6
Chromosome Size	9
Mutation Rate = $\frac{1}{Chromosome\ Size}$	$0.\dot{1}$
Search Time	24 Hours (record results every 4 hours starting at 2h)
Minimum Split Parameter	10%
Search Iterations	22

- Goal: Stabilize and Reach the Search Plateau
- Combining Classification and Regression Trees *(Leo B. et al, 1984)*
- Understandable Binary Outcome + a Real-Valued Score
- Calculates the Score from a vector of combined vectors with complex relationships – that results in a high-risk scenario

EXPERIMENTAL METRICS

- Region Size (%) measures the size of the critical regions
- Lower the better

$$R_i = \prod_{j=1}^{n} \frac{|d_j|}{|D_j|} \times \prod_{j=1}^{m} \frac{\max(d'_j) - \min(d'_j)}{\max(D'_j) - \min(D'_j)}$$

- Goodness of Fit measures how well the trees are generated (DT)
- i.e., Classification Accuracy = Ratio of Correctly Classified Labels regardless of +ve or -ve
- Higher the better

- Number of distinct critical test scenarios

STATISTICAL METRICS

- Hypervolume measures **convergence** and **diversity** of the Pareto Front. *(Zitzler et al, 1998)*
- Higher HV = Larger Covered Area = Close to True Pareto Front

$$HV(S) = vol(\bigcup_{x \in S} [f_1(x), r_1] \times [f_2(x), r_2] \times \cdots \times [f_m(x), r_m])$$

- Generational Distance measures **convergence accuracy**. *(Van Veldhuizen, 1999)*
- Lower the better = Close to True Pareto Front (Optimality)

$$GD(S) = \frac{1}{|S|} \sqrt{\sum_{x \in S} d(x)^2}$$

- Spread measures **diversity / uniformity** of the distribution of solutions along the Pareto front. *(Deb et al, 2002)*
- Lower the better = more evenly spaced and well-distributed solutions

$$SP(S) = \frac{d_f + d_l + \sum_{i=1}^{|S|-1} |d_i - \bar{d}|}{d_f + d_l + (|S| - 1)\bar{d}}$$

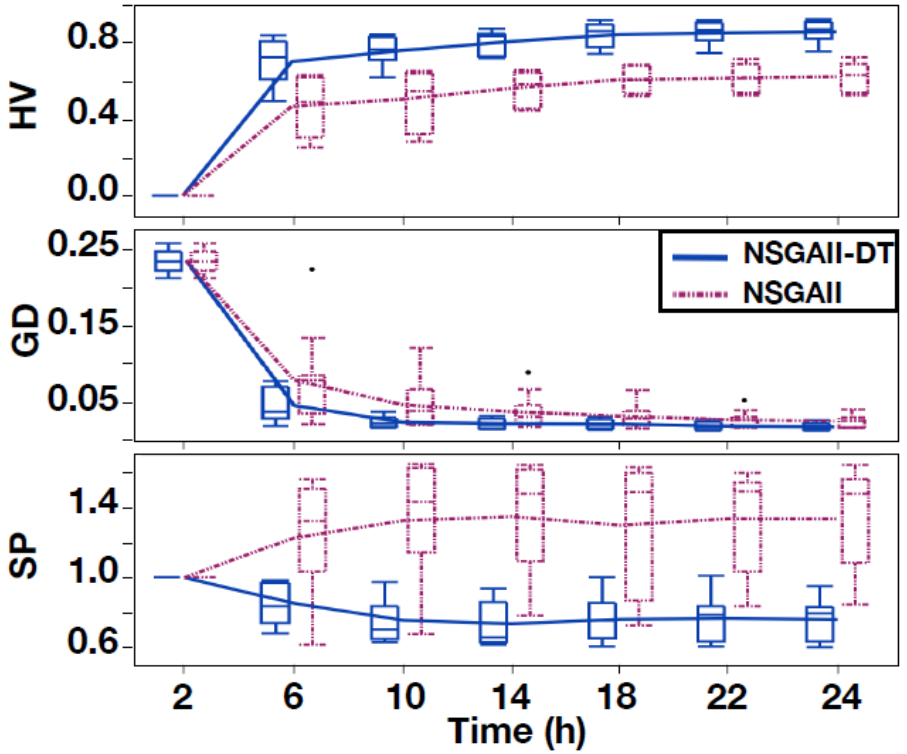
- Statistical Significance: **non-parametric pairwise Wilcoxon rank sum test** and **Vargha-Delaney's** $\widehat{A}_{12}$ effect size.
- Level of Significance $\alpha = 0.05$

EVALUATION OUTCOMES

- NSGAII-DT is consistently better!
- Converges better!
  - To a higher HV
  - To lower GD and SP values eventually
- Statistical Test is conducted at 24H:

NSGAII-DT → NSGAII

Metric	p-value	$\widehat{A}_{12}$
HV	0.01	0.9
GD	0.07	0.3
SP	0.01	0.1



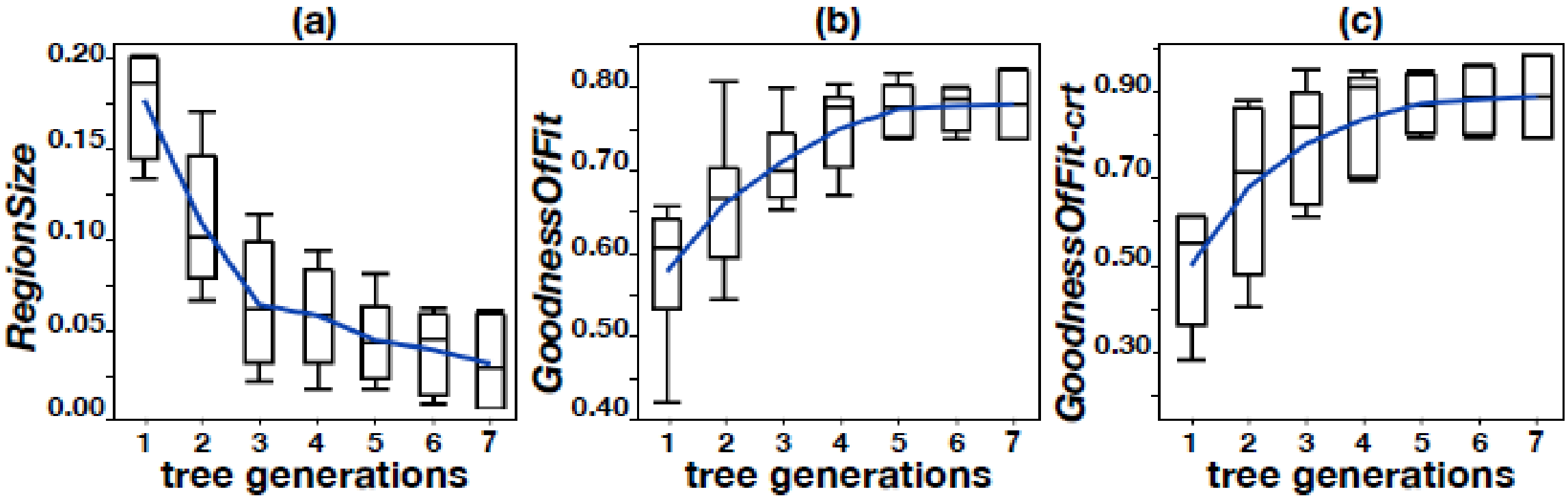
EVALUATION OUTCOMES

Most Importantly!!!

- Number of distinct critical test scenarios (in 15 runs)

	NSGAII	NSGAII-DT
Number of Critical Scenarios	411 (58.05%)	731 (69.95%)
Number of Non-Critical Scenarios	297	314
Total	**708**	**1045**

EVALUATION OUTCOMES



CRITIQUE

Soundness 

- Clearly defined the problem, and experiment setup
- Clearly defined and properly used known metrics
- Fitness Function makes sense overall
- Accounted for worst scenarios, and impact of constraints

Novelty 

- Makes use of machine learning – Decision Tree
- Brilliant use of statistical tests

Significance 

- Adapted ADAS testing from static combinatorial sampling to a learnable, adaptive evolutionary search
- Guided search improves convergence behavior
- Research Questions with Doubt → Contributions

Verifiability & Transparency 

- They maintained explainability for practitioners
- Clearly stated chosen parameters..
- Clearly evaluated outcomes with graphs.
- Employed standard, open tools like PLEDGE

CONCLUSION

1. How is ML being used in the loop to guide the evolutionary search? Is it effective enough?

- Guided by Decision Tree
- Significantly outperforms the ordinary algorithm (Converging better in multiple metrics)
- Decision Tree helps generate more critical scenarios for practitioners

2. Does it help characterize and converge towards homogeneous critical regions (high-risk scenarios)?

- Better Pareto Fronts generated by Decision Tree
- Attempt to minimize metrics: Region Size, Goodness of Fit
- Number of Critical Regions generated by Decision Tree
- Visualized Representation

POSSIBLE FUTURE WORKS

- Other Machine Learning Models?
- Computational Overhead (DT)
- Further System/Unit Testing

OTHER REFERENCES

- Christopher Henard, Mike Papadakis, Gilles Perrouin, Jacques Klein, and Yves Le Traon. 2013. PLEDGE: a product line editor and test generation tool. In Proceedings of the International Software Product Line Conference co-located workshops (SPLC'13). ACM, New York, NY, USA, 126–129.
- Deb et al. 2002. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II", IEEE Trans. Evolutionary Computation.
- D. Richard Kuhn, Dolores R. Wallace, and Albert M. Gallo. 2004. Software Fault Interactions and Implications for Software Testing. IEEE Transactions on Software Engineering 30, 6 (2004), 418–421.
- Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. Classification and Regression Trees. Wadsworth, Belmont, CA, U.S.A.
- Tribikram Dhar, Nilanjan Dey, Surekha Borra, R. Simon Sherratt: Challenges of Deep Learning in Medical Image Analysis – Improving Explainability and Trust. Jan 2023, DOI: 10.1109/TTS.2023.3234203.
- Van Veldhuizen. 1999. "Multiobjective Evolutionary Algorithm Test Suites", PhD thesis, Air Force Institute of Technology.
- Zitzler & Thiele. 1998, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach", IEEE Trans. Evolutionary Computation.

THANK YOU

KMOCK073@UOTTAWA.CA