Hillis's Parasite Program Recreation

Noah Sapire

I. Introduction

Daniel Hillis's 1990 paper introduced a revolutionary concept in optimization: co-evolving parasites as a means to improve the evolution of sorting networks. By introducing dynamically generated adversarial test cases, Hillis demonstrated how evolving challenges could prevent premature convergence in evolutionary processes, ultimately driving populations to discover more robust and efficient solutions. This paper was a pivotal moment in evolutionary computation, showcasing the power of co-evolution in fostering diversity and adaptability in optimization problems.

This project aimed to recreate the methods presented in Hillis's paper using modern programming techniques. By implementing genetic algorithms to evolve sorting networks and integrating co-evolving parasites as dynamic test cases, the project sought to validate Hillis's results. Specifically, it explored three key systems inspired by his work: a static sorting network for benchmarking, an evolving sorting network optimized under static test conditions, and a co-evolving sorting network integrated with dynamic adversarial inputs. The study highlights how co-evolving parasites can enhance the discovery of optimal solutions, even in computationally constrained environments.

II. Hillis's Paper

In his groundbreaking paper, W. Daniel Hillis explored the limitations of traditional evolutionary algorithms in solving optimization problems. He observed that static environments often led to premature convergence, where populations quickly stagnated around suboptimal solutions. To address this, Hillis proposed the concept of co-evolution, where two interacting populations evolve simultaneously. One population represents the solutions being optimized (in this case, sorting networks), while the other represents dynamic challenges or adversaries (referred to as parasites). This interaction creates a constantly shifting fitness landscape, forcing both populations to adapt continuously.

Hillis applied this framework to evolve sorting networks, which are mathematical constructs designed to sort fixed-size inputs using a predefined sequence of comparison-exchange operations. Sorting networks are inherently parallel, making them ideal for hardware implementations in high-speed systems. However, discovering efficient sorting networks with minimal comparators is a complex optimization problem. In his experiments, Hillis evolved sorting networks for 16 inputs, achieving a design with just 61 comparators—the theoretical minimum for this problem size.

The parasites in Hillis's system were binary test cases designed to challenge sorting networks. Unlike static random inputs, these test cases evolved alongside the networks, targeting their specific weaknesses. By doing so, the parasites maintained evolutionary pressure, ensuring the networks continually improved. This dynamic interplay between networks and parasites demonstrated the efficacy of co-evolution in overcoming stagnation and discovering innovative solutions.

III. Methods

A. Static Sorting Networks

Sorting networks are deterministic systems that sort fixed-size inputs through a series of comparators. Comparators operate on two input lines, exchanging their values if the upper line contains a larger value. A sorting network is evaluated by its ability to sort all permutations of its inputs correctly. This project included a static implementation of Batcher's odd-even mergesort network and a manually optimized 16-input sorting network with 60 comparators. These static networks served as benchmarks to assess the performance of evolved networks.

B. Evolving Sorting Networks

The Evolution Network script implemented a genetic algorithm to evolve sorting networks. A population of random networks was initialized, and their fitness was evaluated based on their ability to sort random test cases. Fitness was defined as the proportion of test cases sorted correctly. High-performing networks were selected for reproduction, with mutations introducing variations such as adding or removing comparators. Over successive generations, the population converged toward networks with higher fitness. This system emulated Hillis's baseline evolutionary process, which demonstrated improvement under static conditions but often stagnated once a high level of fitness was achieved.

C. Co-Evolving Inputs

To replicate Hillis's co-evolving parasites, the Parasite Network script introduced dynamic test cases that evolved alongside the sorting networks. These parasites were binary input arrays that targeted weaknesses in the networks, such as specific permutations that were difficult to sort. The fitness of a parasite was determined by its ability to disrupt the sorting performance of networks in the current population. Parasites that consistently exposed vulnerabilities were retained and propagated, ensuring the evolutionary process remained competitive.

This co-evolutionary framework created a feedback loop where networks adapted to challenging inputs, and parasites adapted to exploit weaknesses in the networks. This dynamic interaction mimicked the shifting fitness landscape described

in Hillis's paper, driving the discovery of diverse and robust sorting solutions.

IV. Results

A. Static Evolutionary Results

Under static test conditions, the evolutionary process demonstrated steady improvements in sorting network performance over generations. Starting from random populations, networks with 16 inputs and up to 65 comparators achieved near-optimal sorting accuracy after several generations. However, as Hillis predicted, the population eventually stagnated once most networks reached high fitness levels. This stagnation resulted in a lack of diversity, with the majority of networks converging to similar designs.

B. Co-Evolutionary Results

Introducing co-evolving parasites significantly altered the evolutionary dynamics. Networks faced continuously adapting challenges, preventing premature convergence. The most challenging binary input cases forced networks to refine their sorting strategies, resulting in diverse and innovative designs. This interaction led to the emergence of networks that matched or exceeded the robustness of those evolved under static conditions.

One notable outcome was the discovery of a 16-input sorting network with 61 comparators, matching the theoretical minimum achieved in Hillis's experiments. The networks evolved with co-evolving parasites consistently outperformed their static counterparts when tested on unseen inputs, demonstrating greater generalization and adaptability.

C. Graphical Representation

The structure of the evolved sorting networks was visualized using directed acyclic graphs. Each comparator was represented as a vertical arrow connecting two input lines, illustrating the sequence of operations required to sort the inputs. The final co-evolved network displayed a compact and efficient structure, closely resembling the theoretical 61-comparator design.

V. Discussion

The results obtained from the co-evolutionary system validate the core principles of Hillis's work. By introducing adversarial test cases that evolved alongside the sorting networks, the system maintained diversity and avoided premature convergence. This dynamic interaction drove the networks to discover efficient sorting strategies, achieving optimal designs with minimal comparators.

However, replicating Hillis's experiments posed challenges. Hillis's use of the Connection Machine enabled massive parallelism, allowing him to simulate large populations and rapidly evaluate fitness. In contrast, this implementation relied on Python, which limited the scale of experiments due to computational constraints. Additionally, without access to the full text of Hillis's paper, precise details about his experimental setup were unavailable. Despite these limitations, the results closely mirrored Hillis's outcomes, demonstrating the robustness of the co-evolutionary approach.

The success of this project underscores the broader applicability of co-evolutionary systems. By fostering continuous adaptation through adversarial interactions, these systems can tackle complex optimization problems in various domains, from algorithm design to artificial intelligence.

VI. Conclusion

This study successfully recreated Hillis's co-evolutionary framework for evolving sorting networks using parasites. By integrating dynamic adversarial test cases, the system achieved robust and efficient solutions, including a 16-input sorting network with 61 comparators. These results validate Hillis's hypothesis that co-evolving parasites enhance optimization by maintaining evolutionary pressure and preventing stagnation.

While computational and informational constraints limited the scope of this implementation, the outcomes reaffirm the general principles of Hillis's work. The co-evolutionary system demonstrated here highlights the enduring relevance of these ideas for solving modern optimization challenges, providing a foundation for future research and applications.

References

Hillis, W. D. (1990). Co-evolving parasites improve simulated evolution as an optimization procedure. Physica D: Nonlinear Phenomena, 42(1-3), 228-234. https://doi.org/10.1016/0167-2789(90)90076-2

A. A Subsection

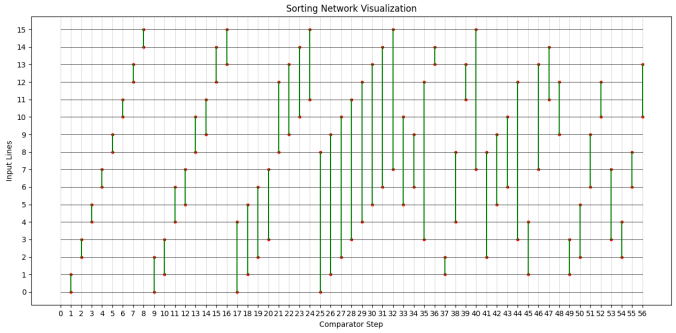

Fig. 1: Optimal Sorting

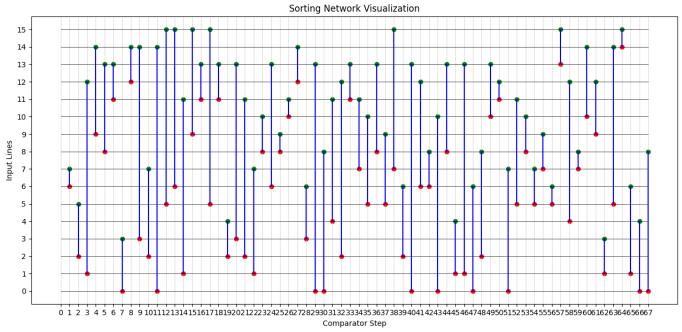

Fig. 2: Evolutionary Sorting

```
Generation 99: Best fitness = 0.4438629150390625
Generation 100: Best fitness = 0.4438629150390625
Original array: [87, 70, 96, 73, 84, 66, 39, 24, 6
Sorted array: [13, 23, 24, 18, 39, 25, 44, 53, 66,
Number of comparators: 61
```

Fig. 3: Parasite Sorting