A Genetic Algorithm for Output-Boundary Value Analysis

Byron DeVries
College of Computing
Grand Valley State University
Allendale, Michigan, USA
devrieby@gvsu.edu

Erik M. Fredericks
College of Computing
Grand Valley State University
Allendale, Michigan, USA
frederer@gvsu.edu

Abstract

Traditional boundary value analysis (BVA) targets input-domain limits to expose defects but relies on predefined input boundaries. Many systems instead exhibit output-defined boundaries arising from behavioral changes. We propose a genetic algorithm that discovers output-based equivalence classes and approximates their corresponding input-domain boundaries by identifying extreme inputs within each behavior. This enables output-driven BVA without prior knowledge of input constraints.

CCS Concepts

• **Software and its engineering → Search-based software engineering**; **Software testing and debugging**; • **Computing methodologies → Genetic algorithms**.

Keywords

boundary value analysis, search-based software engineering, genetic algorithms, black-box testing, novelty search

ACM Reference Format:
Byron DeVries and Erik M. Fredericks. 2026. A Genetic Algorithm for Output-Boundary Value Analysis. In *Genetic and Evolutionary Computation Conference (GECCO Companion '26), July 13–17, 2026, San Jose, Costa Rica.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3795101.3814649

1 Introduction

Boundary value analysis (BVA) identifies inputs that are likely to expose faults at the extremes of a system under test (SUT) [10]. Traditionally, BVA focuses on boundaries defined over the input domain; however, many systems exhibit complex input–output relationships in which meaningful boundaries emerge from changes in output behavior rather than explicitly specified input constraints [6]. Identifying such output-induced boundaries is challenging, and most existing approaches therefore restrict BVA to the input domain [5]. Recent work has shown that search-based techniques, including genetic algorithms, can effectively explore boundary regions of the input space [1–4, 8].

Prior approaches characterize boundaries by analyzing output changes using derivatives [4], distance metrics [3], or quality-diversity methods [1], as well as by maximizing distance from valid inputs to induce invalid behavior [8]. While effective, these methods focus on detecting transitions between behaviors or validity regions rather than approximating the full input regions associated with each behavior.

In this work, we take a complementary perspective by maximizing coverage of input regions associated with each observed behavior. We treat each equivalence partition—defined by similar outputs—as a behavior class and maintain a dedicated archive for each class. Within each archive, we promote individuals at the extremes of the discovered input region by maximizing distance from the archive median. This boundary-focused search enables identification of input-space limits associated with output-defined equivalence classes without prior knowledge of those classes.

The contributions of this paper are a genetic algorithm for output-driven boundary value analysis that (1) discovers behavior-defined equivalence classes and (2) approximates their corresponding input-domain boundaries. Section 2 presents the approach and preliminary results, and Section 3 concludes with directions for future work.

2 Approach & Preliminary Results

This section introduces BVA and our approach while applying it to a specific problem.

2.1 Boundary Value Analysis

Boundary Value Analysis (BVA) is a black-box testing technique that targets inputs near the edges of equivalence partitions, where defects are more likely to occur [10]. Testers typically select values at and near these boundaries. Traditional BVA focuses on partitions defined over the input domain; however, outputs can also induce partitions, with boundaries arising from changes in system behavior rather than explicitly specified input constraints [6]. Identifying such output-defined boundaries is more challenging because they must be inferred from observed behavior rather than known input limits.

2.2 Identifying Boundaries

Our approach identifies input-domain boundaries associated with output-defined behaviors using a behavior-partitioned evolutionary search. We assume that the system under test induces a finite set of equivalence partitions over the input space, where each partition corresponds to a distinct behavior class. The algorithm maintains a separate archive for each discovered behavior class and focuses on identifying the extreme input values associated with each partition. Unlike traditional archive-based novelty search [7], which accumulates diverse individuals, our method maintains per-behavior archives that retain extreme inputs and selects candidates that maximize distance from the archive median.

GECCO Companion '26, San Jose, Costa Rica

ACM ISBN 979-8-4007-2488-6/2026/07
https://doi.org/10.1145/3795101.3814649

The algorithm operates as a generational evolutionary process over a population P of size M, initialized uniformly at random over the input domain. At each generation, individuals are assigned to behavior classes, with new archives created as needed, and inserted into the corresponding archive. Individuals are reassigned if their behavior differs from their current archive, maintaining boundary-focused representations of each behavior as individuals are scored by distance from the archive median. Selection is performed using tournament selection (size 3), followed by blend crossover ($p_c = 0.5$) and Gaussian mutation ($p_m = 0.3$, $\sigma = 50$) to generate offspring that replace the current population.

After a fixed number of generations, the algorithm returns the set of archives $\{A_k\}_{k \in \mathcal{B}}$, where each archive approximates the boundary of the input region associated with a behavior class. The extreme values retained in each archive correspond to the estimated input-domain boundaries for output-based BVA.

This method is related to MAP-Elites [9] and Novelty Search [7], but differs in two key ways: (1) multiple behavior classes (i.e., multiple archives) are discovered dynamically rather than predefined, and (2) search is explicitly biased toward boundary exploration by promoting individuals that maximize distance from the archive median, rather than optimizing for performance or global novelty in a single archive.

2.3 Preliminary Results

We evaluated the proposed method on a 1D continuous domain $x \in [-5000, 5000]$ with a discrete behavior mapping inputs to 12 classes (months), inducing non-uniform partitions with boundaries unknown to the algorithm. Each input is interpreted as a day-of-year value and assigned to a month based on its interval. Inputs falling outside the valid range are treated as infeasible.

This mapping induces 12 equivalence partitions over the input domain, with boundaries defined by transitions between adjacent classes. These class boundaries were not provided to the algorithms a priori, and the task is to recover them through exploration.

The proposed method was compared against two baselines: (1) standard novelty search and (2) random sampling. All methods were executed for 50 independent trials. The evolutionary approaches used a population size of $M = 120$ and were run for $T = 40$ generations. Variation consisted of blend crossover with probability $p_c = 0.5$ and Gaussian mutation with probability $p_m = 0.3$, with mutation standard deviation $\sigma = 50$. The proposed method maintained per-class archives of fixed size $N = 30$, retaining extreme values, while standard novelty search used a single global archive with k-nearest neighbor novelty ($k = 10$) and a novelty threshold of 5.0.

Performance for all methods was measured using total boundary error, defined as the sum of absolute differences between estimated and true boundary locations across all classes, with a penalty (1,000) applied for missing classes.

The proposed method achieved a mean error of 13.27 ± 2.91 (median 12.40), substantially outperforming both standard novelty search (38.89 ± 6.21, median 39.70) and random sampling (41.90 ± 9.15, median 40.30). The minimum and maximum errors observed were 9.45–26.51 for the proposed method, 24.47–51.34 for standard novelty search, and 24.50–63.80 for random sampling.

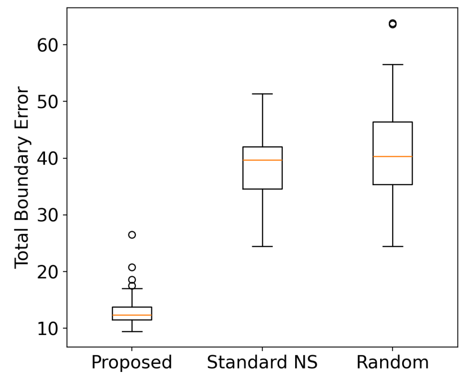

Figure 1: Distribution of total boundary error across 50 trials.

Mann–Whitney U tests confirm significant improvements over both baselines $p < 10^{-8}$, with no difference between baselines.

These results demonstrate that the proposed boundary-preserving archive structure provides a substantial and statistically significant improvement over both unstructured novelty-driven exploration and random search in recovering class boundaries.

3 Conclusions & Future Work

The results suggest that boundary-focused novelty search can effectively recover the input-space support of discrete behavior classes, outperforming random sampling under identical conditions. Future work will explore hybridization with fitness-based methods such as MAP-Elites, adaptive archive structures, and extension to higher-dimensional behavior spaces to support additional BVA formalizations.

Acknowledgments

We gratefully acknowledge support from Grand Valley State University.

References

[1] Sabinakhon Akbarova, Felix Dobslaw, Francisco Gomes de Oliveira Neto, and Robert Feldt. 2025. SETBVE: Quality-diversity driven exploration of software boundary behaviors. *ACM Transactions on Software Engineering and Methodology* (2025).

[2] Felix Dobslaw, Francisco Gomes de Oliveira Neto, and Robert Feldt. 2020. Boundary value exploration for software analysis. In *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE, 346–353.

[3] Felix Dobslaw, Robert Feldt, and Francisco Gomes de Oliveira Neto. 2023. Automated black-box boundary value detection. *PeerJ Computer Science* 9 (2023), e1625.

[4] Robert Feldt and Felix Dobslaw. 2019. Towards automated boundary value testing with program derivatives and search. In *International Symposium on Search Based Software Engineering*. Springer, 155–163.

[5] Xiujing Guo, Hiroyuki Okamura, and Tadashi Dohi. 2024. Optimal test case generation for boundary value analysis. *Software Quality Journal* 32, 2 (2024), 543–566.

[6] Paul C. Jorgensen and Byron DeVries. 2021. *Software Testing: A Craftsman's Approach* (5th ed.). Routledge, Boca Raton, FL.

[7] Joel Lehman and Kenneth O Stanley. 2011. Novelty search and the problem with objectives. *Genetic programming theory and practice IX* (2011), 37–56.

[8] Bogdan Marculescu and Robert Feldt. 2018. Finding a boundary between valid and invalid regions of the input space. In *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*. IEEE, 169–178.

[9] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909* (2015).

[10] Glenford J Myers, Tom Badgett, Todd M Thomas, and Corey Sandler. 2004. *The art of software testing*. Vol. 2. Wiley Online Library.