Efficient Multi-Objective Optimization of AEB Systems: A Comparative Study of Sweeping and NSGA-II

Solano Rigotti
Ygor Galbier
Leimar Malfort
Robert Bosch LTDA, BR

Jian Tang
Robert Bosch LLC, US

E-mails: solanorigotti@hotmail.com, ygorgalbierlopes@gmail.com, leimar.mafort@br.bosch.com, jian.tang2@us.bosch.com

ABSTRACT

Parameter tuning in Autonomous Emergency Braking (AEB) system development is critical, particularly when balancing multiple performance objectives. This study employs the Non-dominated Sorting Genetic Algorithm II (NSGA-II) to efficiently identify optimal scenarios that represent a trade-off between multi-objectives that may dominate each other. The algorithm optimizes time-to-collision (TTC) as a metric for collision risk, and jerk as an indicator of passenger discomfort.

The proposed approach is integrated into a Software-in-the-Loop (SIL) simulation environment using CarMaker, enabling scalable and effective test scenario generation. NSGA-II is used to explore the multi-objective space, identifying a diverse set of Pareto-optimal scenarios that reflect critical trade-offs. The results demonstrate that this approach can be used to efficiently identify critical test scenarios, enhancing the development and validation of ADAS features for AEB systems.

INTRODUCTION

Advanced Driver Assistance Systems (ADAS) have become increasingly important in the automotive industry, providing functionalities that enhance vehicle safety—not only for passengers but also for other road users such as pedestrians and cyclists. In addition to preventing collisions, these systems aim to mitigate their severity when they cannot be avoided [1]–[3].

However, developing ADAS involves several challenges. Among the most significant are the costs and complexity associated with test vehicle instrumentation, which often requires specialized hardware interfaces and licensed software [2], [3]. From an industrial perspective, preparing such vehicles can be time-consuming and bureaucratically complex, especially when international transportation is involved.

Software-in-the-Loop (SIL) simulation has emerged as a promising alternative to address these issues. Many ADAS functionalities can now be validated in a virtual environment, enabling the execution of numerous test scenarios under varying conditions [4], [5]. This offers a practical and cost-effective initial step in the ADAS development and validation pipeline—saving both time and resources.

With this shift toward virtual testing, new challenges arise. This study focuses on such challenges, expanding on prior work by our research group on intelligent vehicle preparation with calibration [6]—previously presented at this conference—and now addressing the intelligent optimization of multi-objective trade-offs in ADAS testing.

Traditional methods like Sweeping, though widely used, are computationally expensive and struggle to identify diverse non-dominated solutions—especially as the number of parameters increases [7]–[9]. These limitations reduce their effectiveness in exploring high-dimensional spaces and capturing meaningful trade-offs. Methods that do not incorporate non-dominated sorting often miss important Pareto-optimal solutions, making them less suitable for complex scenarios where conflicting goals that are subject to dominance effects must be balanced [10], [11].

In this work, we propose a NSGA-II framework to search for optimal trade-offs between multi-objectives in critical driving scenarios. NSGA-II is a well-established evolutionary algorithm for multi-objective optimization that maintains a diverse set of non-dominated solutions across generations. It uses a fast non-dominated sorting approach combined with a crowding distance mechanism to ensure both convergence toward the Pareto front and the preservation of solution diversity [11]. CarMaker, developed by IPG Automotive, is used as the simulation and validation environment. It provides a comprehensive SIL platform for vehicle dynamics, sensors, and control-system modeling [12].

Comparative results show that the NSGA-II framework can reach solutions close to the global optimum, with significantly lower computational cost and processing time compared to exhaustive methods like Sweeping. Specifically, NSGA-II reduced the number of simulations by 77% (120 vs. 528) and the total execution time by 37% (1129 s vs. 1800 s), while maintaining comparable solution quality.

METHODOLOGY

The AEB test scenario was created in IPG CarMaker as a simulation baseline, adapted from an NCAP AEB-CCRm protocol with two modifications, when moving, the target vehicle maintains a constant velocity in the ego vehicle's direction (replacing the standard deceleration profile). The design control parameters were defined within the following boundaries:

- ***Target velocity (V_{tgt}):*** 0–10 m/s
- ***Ego velocity (V_{ego}):*** 14–25 m/s

The multi-objective optimization evaluates the trade-off between two key performance objectives:

- ***95th percentile jerk*** (not maximum jerk) to quantify passenger discomfort. This metric improves robustness to outliers while correlating with subjective discomfort in braking scenarios [13].
- ***Minimum Time-to-Collision (TTC)*** as the collision risk metric.

A multi-objective optimization algorithm based on NSGA-II was implemented using the PyGAD library [14] in Python. This configuration was selected to balance convergence speed and computational cost, especially given the high runtime per simulation (~9 s). Despite the relatively low number of generations, convergence was empirically achieved for the defined problem. Each candidate solution encoded two floating-point genes representing the ego and target vehicle velocities. The fitness function evaluated each solution with respect to two conflicting objectives: minimizing the *minimum Time-to-Collision (TTC)* and maximizing the *95th percentile of jerk*.

A single-point crossover operator was used to recombine genetic material between parents, while uniform mutation was applied. NSGA-II's core mechanisms—fast nondominated sorting and crowding distance calculation—were employed to maintain a well-distributed Pareto front across generations. The optimization process was stopped after five generations. This setup represented in Table 1 promoted diversity and robustness in exploration.

Table 1. NSGA-II Configuration Parameters

Parameters	Value
Population Size	20
Generations	5
Number of Genes	2
Genes Ranges	Ego: [14–25] m/s; Target: [0–11] m/s
Crossover Type	Single-Point
Mutation Type	Adaptative
Parent Selection	NSGA-II
Stopping Criteria	Finish after 5 generations
Python Library	PyGAD
	(Ref: https://pygad.readthedocs.io/)

Figure 1 illustrates the flowchart of the implemented NSGA-II algorithm. The optimization process begins with an initial random sampling of 20 individuals in the design space, representing combinations of *ego* and *target* velocities. Each individual is then evaluated by its fitness in the objective space (e.g., TTC and jerk). Subsequently, the algorithm proceeds through 5 generations, each with a population size of 20 individuals, resulting in 100 additional solutions evaluated during the evolutionary process (120 simulations in total).

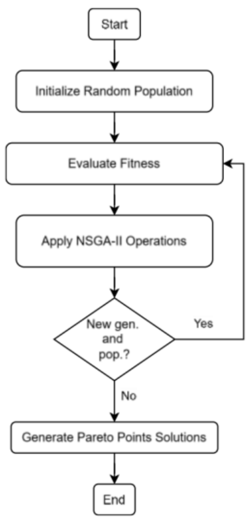

Figure 1. NSGA-II algorithm flowchart.

In each generation, the population undergoes nondominated sorting and crowding distance calculation to rank solutions and maintain diversity. The standard genetic operators—selection, crossover, and mutation—are applied to generate offspring for the next generation. This iterative procedure continues until the stopping criterion is met, yielding a set of Pareto-optimal solutions that balance the objectives.

RESULTS AND DISCUSSION

After conducting the AEB tests in CarMaker—a fully deterministic simulation environment—it was possible to evaluate, for each combination of control variables (*ego* and *target* vehicle speeds), the corresponding *95th percentile of jerk* and the *minimum Time-to-Collision (TTC)* and the other meanings signals. The sweep method was executed with a resolution of 0.25 m/s, generating a total of 529 simulations with a runtime of approximately 30 minutes. Figure 2(a) shows the objective and Figure 2(b) the design space obtained through the sweep method, where the filled circle solutions represent the Pareto front—non-dominated solutions that highlight the optimal trade-off between the objectives.

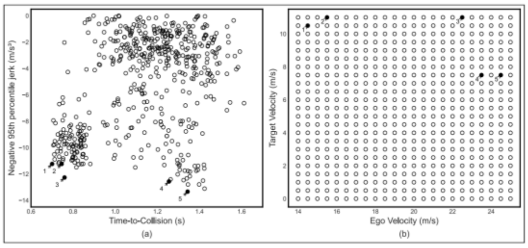

Figure 2. Sweep test results: (a) solutions and Pareto's points in objective space, and (b) in design space.

The same optimization problem was addressed using the NSGA-II evolutionary algorithm, as shown in Figure 3. This configuration used, represented in Table 1, resulted in only 120 simulations. Remarkably, with nearly 80% fewer simulations, NSGA-II was able to identify a Pareto front very similar to the one obtained via sweeping (Figure 6), effectively covering the same critical region in the objective space. Is possible to identify that the solutions intelligently converge to the same critical regions shown in Figure 3(b)—demonstrating the algorithm's effectiveness with much lower computational cost compared with sweeping.

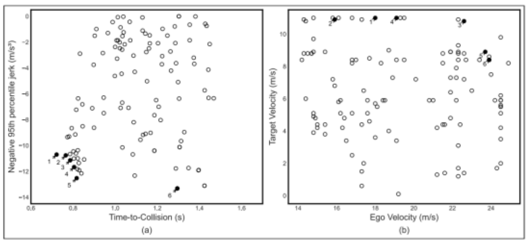

Figure 3. NSGA-II test results: (a) solutions and Pareto's points in objective space, and (b) in design space.

Furthermore, NSGA-II proved more efficient at prioritizing regions of the design space with a higher impact on the objectives, avoiding redundant simulations in low-impact areas. This targeted sampling is one of the key advantages of evolutionary algorithms over exhaustive methods. Another crucial point is scalability: while the sweep method quickly becomes infeasible as the number of design variables increases, NSGA-II maintains its efficiency in higher-dimensional search spaces, making it a promising approach for real-world ADAS calibration and validation tasks.

To assess the quality of the Pareto fronts generated by both methods, two key performance indicators (KPIs) were adapted from the literature. The first, Hypervolume (HV) [15], was computed for each set of Pareto-optimal solutions relative to a common reference point in the objective space, set as [1.8, –6.0]. A larger hypervolume indicates that the Pareto front reflects a more comprehensive coverage of solution space, reflecting better performance.

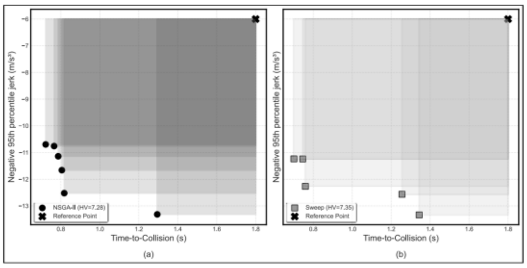

Figure 4. Hypervolume (HV) comparison results: (a) NSGA-II optimization, and (b) Sweep method.

The second KPI is a simplified form of Inverted Generational Distance (IGD) [16], computed relative to the point [0.0, –17.0] which represents the origin. Although this is not a strict IGD definition, the metric provides useful comparative insight.

It is important to note that the choice of reference points has a significant impact on the results, and for a more robust evaluation, the Hypervolume (HV) is a more reliable indicator in this context due to the absence of a known reference Pareto front. A lower IGD value indicates better convergence toward the origin, reflecting superior solution quality.

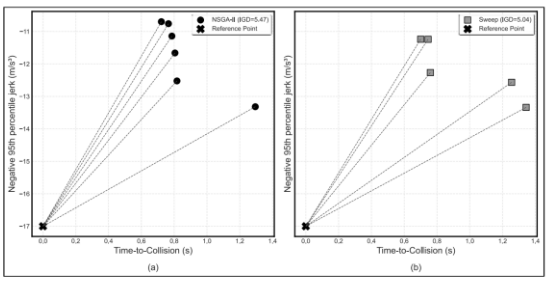

Figure 5. Inverted Generational Distance (IGD) comparison results: (a) NSGA-II optimization, and (b) Sweep method.

As shown in Table 2, NSGA-II demonstrated substantial computational advantages over the sweep method, achieving a 77% reduction in the number of simulations and a 37% faster execution time.

Table 2. Comparison of computational cost between sweep and NSGA-II.

Optimization Approach	Number of Simulations	Total Execution Time (s)
Sweep	528	1800
NSGA-II	120 (**-77**%)	1129 (**-37**%)

While Table 3 shows that NSGA-II produced slightly lower performance in both quality metrics, these modest differences are offset by the significant savings in computational cost. Figure 6 visually confirms that NSGA-II generates Pareto fronts of comparable quality while requiring far fewer simulations.

Table 3. Pareto Front performance metrics for each method, with relative differences.

Optimization Approach	Hypervolume (HV)	Inverted Generational Distance (IGD)
Sweep	7.35	5.04
NSGA-II	7.28 (**-1**%)	5.47 (**+9**%)

This cost-benefit trade-off makes NSGA-II a viable and practical option for applications requiring fast and iterative optimization, especially in scenarios with time constraints or limited computational resources. The choice between methods should, therefore, be based on the desired balance between solution accuracy and computational efficiency.

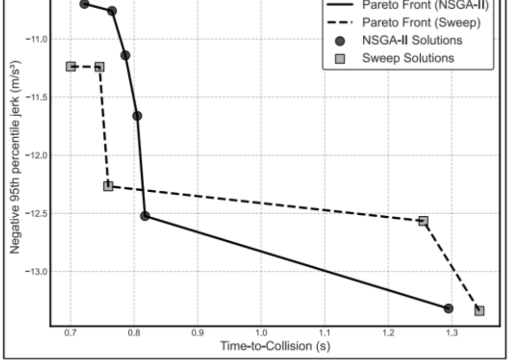

Figure 6. NSGA-II and Sweep Pareto front comparison.

These results demonstrate that NSGA-II-based multi-objective optimization not only significantly reduces simulation time but also preserves the quality and diversity of critical solutions. This reinforces NSGA-II's potential as a robust and efficient tool for trade-off analysis and critical scenario generation in AEB systems.

CONCLUSION

This study demonstrated the effectiveness of the NSGA-II evolutionary algorithm as a computationally efficient alternative to exhaustive sweep-based simulations for multi-objective optimization in AEB systems. By significantly reducing the number of required simulations and overall runtime—while still generating Pareto fronts of comparable quality—NSGA-II proves to be a robust and scalable tool for identifying optimal trade-offs among multiple objectives. Despite minor differences in solution quality in terms of HV and IGD, the significant gains in computational efficiency make NSGA-II a compelling choice for real-time and resource-constrained optimization tasks.

Nevertheless, one limitation of NSGA-II lies in its need to evaluate an entire population at each iteration, which can still represent a substantial simulation cost. To further improve efficiency, future works will explore the application of alternative efficient algorithms for scenario generation and optimization. These approaches aim to further reduce simulation demands—especially in terms of simulation time—without compromising solution quality.

REFERENCES

[1] K. Bengler; K. Dietmayer; B. Farber; M. Maurer; C. Stiller; H. Winner. *Three Decades of Driver Assistance Systems: Review and Future Perspectives*. IEEE Intelligent Transportation Systems Magazine, vol. 6, n. 4, p. 6–22, 2014. DOI: 10.1109/MITS.2014.2336271.

[2] A. Ziebinski; R. Cupek; D. Grzechca; L. Chruszczyk. *Review of Advanced Driver Assistance Systems (ADAS)*. Proceedings of the International Conference of Computational Methods in Sciences and Engineering (ICCMSE-2017), Thessaloniki, Greece, 2017, p. 120002. DOI: 10.1063/1.5012394.

[3] R. Bhagat; P. Singh; A. K. Srivastava; E. S. Khanna. *A Comprehensive Review on Advanced Driver Assistance Systems (ADAS)*. International Journal of Advances in Engineering and Management, vol. 5, n. 5, p. 115–122, maio 2023. DOI: 10.35629/5252-0505115122.

[4] Z. Fei; M. Andersson; A. Tingberg. *Correlation of Software-in-the-Loop Simulation with Physical Testing for Autonomous Driving*. arXiv, 5 jun. 2024. Disponível em: http://arxiv.org/abs/2406.03040. Acesso em: 23 abr. 2025.

[5] R. Keil; J. A. Tschorn; J. Tümler; M. E. Altinsoy. *Evaluation of SiL Testing Potential — Shifting from HiL by*

Identifying Compatible Requirements with vECUs. Vehicles, vol. 6, n. 2, p. 920–948, maio 2024. DOI: 10.3390/vehicles6020044.

[6] S. R. Caiano; Y. G. D. M. Lopes. *Usando algoritmo baseado em dados de IA para auxiliar no desenvolvimento de ADAS*. Blucher Engineering Proceedings, São Paulo, Brasil, ago. 2024, p. 202–208. DOI: 10.5151/simea2024-PAP34.

[7] Y. L. Murphey; R. Milton; L. Kiliaris. *Driver's Style Classification Using Jerk Analysis*. 2009 IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems, Nashville, TN, EUA, mar. 2009, p. 23–28. DOI: 10.1109/CIVVS.2009.4938719.

[8] X. Zhang et al. *Finding Critical Scenarios for Automated Driving Systems: A Systematic Literature Review*. arXiv, 16 out. 2021. Disponível em: http://arxiv.org/abs/2110.08664. Acesso em: 23 abr. 2025.

[9] L. Yang et al. *A Systematic Review of Autonomous Emergency Braking System: Impact Factor, Technology, and Performance Evaluation*. Journal of Advanced Transportation, 2022, p. 1–13, abr. 2022. DOI: 10.1155/2022/1188089.

[10] N. Subaşı. *Comprehensive Analysis of Grid and Randomized Search on Dataset Performance*. European Journal of Engineering and Applied Sciences, vol. 7, n. 2, p. 77–83, dez. 2024. DOI: 10.55581/ejeas.1581494.

[11] K. Deb; A. Pratap; S. Agarwal; T. Meyarivan. *A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II*. IEEE Transactions on Evolutionary Computation, vol. 6, n. 2, p. 182–197, abr. 2002. DOI: 10.1109/4235.996017.

[12] C. Miquet; A. Frings. *The Vehicle-in-the-Loop Method*. Karlsruhe: IPG Automotive GmbH, 2019. Disponível em: https://www.ipg-automotive.com/fileadmin/data/products_and_solutions/test_systems/vehicle-in-the-loop/product_sheets/VIL_Whitepaper_EN.pdf. Acesso em: 5 maio 2025.

[13] N. de W. Ks, er. *Standards for Passenger Comfort in Automated Vehicles: Acceleration and Jerk*. Applied Ergonomics, 2023.

[14] A. F. Gad. *PyGAD: An Intuitive Genetic Algorithm Python Library*. 2021. Disponível em: https://pygad.readthedocs.io/.

[15] A. P. Guerreiro; C. M. Fonseca; L. Paquete. *The Hypervolume Indicator: Problems and Algorithms*. ACM Computing Surveys, vol. 54, n. 6, p. 1–42, jul. 2022. DOI: 10.1145/3453474.

[16] D. A. van Veldhuizen; G. B. Lamont. *Multiobjective Evolutionary Algorithm Test Suites*. Proceedings of the 1999 ACM Symposium on Applied Computing, San Antonio, Texas, EUA, fev. 1999, p. 351–357. DOI: 10.1145/298151.298382.