软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007566] [CSTR: 32375.14.jos.007566]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

结合模糊测试的安全攸关车辆配置搜索*

王铁鑫[1], 马健伟[1], 林 聪[1], 杨 科[1], 王 飞[2]

[1](南京航空航天大学 计算机科学与技术学院, 江苏 南京 211106)

[2](苏州科技大学 电子与信息工程学院, 江苏 苏州 215009)

通信作者: 王铁鑫, E-mail: tiexin.wang@nuaa.edu.cn

摘 要: 随着自动驾驶应用的快速普及, 其安全性问题成为学术界及工业界共同关注的焦点. 针对自动驾驶系统(autonomous driving system, ADS)的测试是解决该问题的有效手段. 目前, 主流测试方法是基于驾驶场景的仿真测试, 即通过模拟各种场景元素, 如道路、行人等, 评估待测 ADS 的决策. 然而, 现有方法多聚焦于关键驾驶场景的构建与动态生成, 忽视了车辆自身配置变化, 如车重、扭矩等, 对部署于其上的 ADS 的决策影响. 针对该问题, 基于课题组前期工作 SAFEVAR, 提出安全攸关的车辆配置高效搜索方法 SAFEVCS. SAFEVAR 采用搜索算法, 探索暴露 ADS 安全隐患的车辆配置设置 (VCS); 为提高搜索结果的多样性, SAFEVCS 引入模糊测试, 改进搜索算法交叉与变异算子的条件限定及约束; 为提高搜索效率, SAFEVCS 进一步结合车辆动力学知识, 实现搜索终止策略和去重策略的自适应. 为评估 SAFEVCS 的有效性及执行效率, 以 SAFEVAR 为对比基线, 在 3 个驾驶场景下进行大规模实验. 实验结果表明, SAFEVCS 生成的 VCS 能够有效暴露 ADS 安全隐患. 在晴天、雨天两种天气条件下, 行人横穿马路的仿真场景中, SAFEVCS 搜索到的解集能够显著降低 ADS 的安全表现, 且在相同的实验环境下, 仿真效率提升近 2.5 倍.

关键词: ADS 安全; 仿真测试; 模糊测试; 搜索算法; 车辆配置

中图法分类号: TP311

中文引用格式: 王铁鑫, 马健伟, 林聪, 杨科, 王飞. 结合模糊测试的安全攸关车辆配置搜索. 软件学报. http://www.jos.org.cn/1000-9825/7566.htm

英文引用格式: Wang TX, Ma JW, Lin C, Yang K, Wang F. Search for Safety-critical Vehicle Configurations with Fuzzing Testing. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7566.htm

Search for Safety-critical Vehicle Configurations with Fuzzing Testing

WANG Tie-Xin[1], MA Jian-Wei[1], LIN Cong[1], YANG Ke[1], WANG Fei[2]

[1](College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

[2](School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou 215009, China)

Abstract: As autonomous driving applications are rapidly popularized, their safety has become the common focus of both academia and industry. Autonomous driving system (ADS) testing is an effective means for solving this problem. Currently, the mainstream testing method is the scenario-based simulation test, which evaluates the decision of ADS to be measured by simulating various elements of driving scenarios, such as roads and pedestrians. However, existing methods mainly focus on the construction and dynamic generation of critical driving scenarios, neglecting the influence of configuration changes of the vehicle itself, such as its weight and torque, on the decision-making of ADS deployed on the vehicle. To address this issue, based on the previous work SAFEVAR, this study proposes SAFEVCS, an efficient search method for safety-critical vehicle configurations. SAFEVCS employs a search algorithm to explore the vehicle configuration setting (VCS) that exposes safety vulnerabilities of ADS. Furthermore, to improve the diversity of the search results, SAFEVCS introduces fuzzing to optimize the conditions and constraints of crossover and mutation operators in search algorithms. To improve search efficiency, SAFEVCS further combines the vehicle dynamics knowledge, which achieves the self-adaption of search

* 收稿时间: 2025-01-14; 修改时间: 2025-04-21, 2025-06-17, 2025-08-28; 采用时间: 2025-10-29; jos 在线出版时间: 2026-02-11

termination strategy and deduplication strategy. To evaluate the effectiveness and execution efficiency of SAFEVCS, the study takes SAFEVAR as the baseline for comparison and carries out extensive experiments under three driving scenarios. The experimental results show that VCS generated by SAFEVCS can effectively expose the safety vulnerabilities of ADS. In the two weather conditions of sunny and rainy days, under the simulation scenarios of pedestrians crossing the road, the obtained solution set significantly decreased the safety performance of the ADS under test, and under the same experiment environment, the simulation efficiency is increased by approximately 2.5 times.

Key words: autonomous driving system (ADS) safety; simulation testing; fuzzing testing; search algorithm; vehicle configuration

人工智能及通信等技术的持续发展与逐步成熟, 提升了自动驾驶服务场景的多样性. 在服务场景多元化的背景下, 自动驾驶的安全性受到越来越多的关注. 作为保障自动驾驶安全的关键, 针对自动驾驶系统 (autonomous driving system, ADS) 的测试方法成为该领域的研究热点之一. 传统的测试方法需要进行数亿甚至数百亿公里的道路测试, 不仅耗时耗力, 且在覆盖驾驶场景方面存在不足, 很难完全验证 ADS 的安全性[1]. 为解决这一问题, 自动驾驶仿真测试应运而生. 仿真测试通过在虚拟环境中模拟各种驾驶场景, 评估待测 ADS 决策的安全性. 仿真测试不仅可以大幅度减少所需的测试里程, 还能在短时间内覆盖更多的驾驶关键场景, 从而更全面地评估 ADS 在不同天气条件、道路状况下的决策准确性、车辆控制稳定性和安全性. 作为 ADS 测试的重点, 关键场景能够在特定驾驶情境下评估 ADS 的表现[1], 协助测试人员发现 ADS 的潜在问题. 因此, 针对自动驾驶的关键场景生成已成为自动驾驶测试的重要方法[2].

现有的基于场景的自动驾驶测试方法, 大多使用搜索或深度学习技术实现, 聚焦于关键场景的构建和生成[3−7], 能够生成不同天气条件、不同道路条件下的关键场景, 实现对自动驾驶车辆 (autonomous vehicle, AV) 的测试. 然而, 这些方法在关键场景的构建与生成中, 聚焦于 AV 外部驾驶环境的配置与动态改变, 忽视了 AV 自身的车辆配置设置 (vehicle configuration setting, VCS), 如车重、轮胎摩擦系数等, 对 ADS 安全性的影响. 在真实驾驶场景中, 生产中的误差、使用中的自然损耗等, 都会导致车辆 VCS 的变化, 进而影响 ADS 决策的安全性表现, 如在需要紧急制动的场景下, 由于车辆的重量变化导致 ADS 无法正确预测安全的刹车距离, 进而不能及时做出相应的决策, 导致车辆发生不安全行为.

经过详细的文献调研, 在自动驾驶测试方向, 尚未发现除本课题组文献 [8,9] 以外的, 其他关注 AV 自身 VCS 的已发表研究工作. 然而, 车辆动力学领域的经典文献 [10], 对 VCS 进行了系统分析并强调了不同 VCS 对车辆行驶安全的影响. 实际上, 由于 VCS 及其中每一配置项取值变化的多样性, 无法对待测自动驾驶车辆 (autonomous vehicle under test, AVUT) 的所有 VCS 组合进行测试, 因此, 本课题组先后提出了基于搜索的测试方法. 其中, 文献 [8] 针对高级驾驶辅助系统 (advanced driving assistance system, ADAS) 的紧急刹车功能, 使用遗传算法构建相应的搜索问题与搜索场景, 实现了对不同模拟器、不同 ADAS 的不安全配置搜索. 文献 [9] 则针对 VCS (如轮胎摩擦系数、发动机最大转速等) 对 ADS 进行了关键配置搜索, 通过设置 VCS 内不同配置项的范围, 搜索获得大量能够暴露 ADS 安全隐患的 VCS, 并论证了在不同天气场景下不同 VCS 对 ADS 的影响.

然而, 上述针对潜在 ADAS、ADS 安全隐患的 VCS 搜索方法, 在执行仿真实验的算力消耗及获得测试结果的多样性等方面, 均存在一定的局限性. 具体表现如下: 1) 由于搜索过程缺乏车辆动力学领域相关知识的指导, 导致搜索结果相似度高、多样性差; 2) 搜索时忽略 VCS 配置项间的相互影响, 配置项间协同变化时缺乏合理性, 导致无效搜索; 3) 在搜索过程中使用固定种群代数作为终止搜索条件, 缺乏针对搜索过程的度量, 无法在达到收敛时及时终止搜索过程, 导致搜索效率较低.

为解决以上问题, 本研究提出了一个结合模糊测试的安全攸关车辆配置搜索方法 SAFEVCS, 旨在高效地搜索可导致潜在 ADS 安全隐患的多样性 VCS. SAFEVCS 通过引入模糊测试方法, 改进了搜索算法算子的多样性条件限定, 有效提高了搜索获得 VCS 的多样性; 同时, 设置相关配置项组合关联限定, 提高 VCS 的变异合理性. 针对搜索效率低下的问题, 引入终止策略与去重策略, 能够在减少重复运行的情况下, 及时检测搜索算法的运行状态, 判断搜索算法的收敛情况, 减少算法运行时间. 本研究主要贡献如下.

(1) 结合模糊测试与搜索算法提出 SAFEVCS, 探索不同 VCS 对 ADS 安全性表现的影响.

(2) 在 SAFEVCS 中引入对算子的模糊测试方法, 改进搜索算法的运行过程, 使得搜索过程能够更全面、更高效地探索潜在的危险配置和场景组合, 提升 VCS 搜索的覆盖率和运行效率.

(3) 结合 CARLA 模拟器设计 SAFEVCS 验证实验, 在 3 个驾驶场景下验证 SAFEVCS 的有效性与高效性, 并论证 VCS 对 ADS 安全性影响的普适性.

本文第 1 节介绍自动驾驶测试的相关工作. 第 2 节介绍 SAFEVCS 涉及的核心理论与关键技术, 包括遗传算法、自动驾驶测试的安全性评估指标等. 第 3 节介绍结合模糊测试的安全攸关 VCS 搜索方法 SAFEVCS. 第 4 节通过分析与对比实验, 验证了 SAFEVCS 的有效性与高效性. 最后, 第 5 节总结全文.

1 相关工作

1.1 基于场景的自动驾驶测试

基于场景的自动驾驶测试是一种通过构建场景环境要素, 对 ADS 进行测试的方法. 该方法使用不同算法生成测试场景. Tang 等人[11]总结了基于场景的自动驾驶测试方法, 给出了现阶段自动驾驶测试的挑战, 指出了自动驾驶测试的复杂性. Ben Abdessalem 等人[12]使用神经网络及搜索算法, 实现了特定场景下, 针对 ADAS 的安全性验证. 此外, 文献 [13] 中, Ben Abdessalem 等人使用机器学习方法与基于搜索的方法生成关键场景, 即通过引入机器学习模型检查场景状态, 进而加速搜索过程. Luo 等人[14]研究了测试用例优先级技术, 并采用多目标搜索算法, 实现了针对 AV 违规行为的高效搜索. 测试用例优先级技术还被用于加速 ADS 的回归测试, 并取得了显著的效果[15]. 为了填补场景设计和执行之间的差异, Rodrigo 等人[5]搭建了仿真驾驶员-车辆系统 (simulated driver-vehicle, SDV), 将车辆表示为动态实体, 并通过场景设计和测试人员设定的目标对其行为进行约束, 从而实现精确和可靠的模拟过程. 李文礼等人[6]为提高自动驾驶车辆仿真测试场景的可解释性和高风险场景覆盖度, 提出了一种博弈论与神经网络结合的仿真测试场景生成算法, 能够生成大量具有现实博弈交互行为的高风险交互轨迹. Zong 等人[16]基于多目标遗传算法, 提出了行为树驱动的场景生成方法; 该方法能够灵活建模多类型交通参与者及其复杂行为, 强调通过多维度指标综合优化以识别关键场景. Tian 等人提出了 MOSAT[17]和 CRISCO[18]方法. 前者通过建模交通参与者行为并优化风险性、干扰性和多样性, 系统性暴露 ADS 漏洞; 后者则从真实数据挖掘关键行为模式, 强化场景临界性, 提升测试挑战性. 相较之下, Zohdinasab 等人[19]提出的方法在进化搜索中引入替代模型, 显著降低了对 ADS 执行的依赖, 提升了搜索效率. Humeniuk 等人[20]针对自主信息物理系统提出 AmbieGen 框架, 通过简化系统模型和多目标优化, 有效平衡了故障揭示能力与搜索资源开销. Li 等人[21]针对深度神经网络模糊测试, 提出了结合优先采样、蒙特卡洛树搜索与归档机制优化的多目标方法, 进一步在种群多样性与局部搜索能力之间取得了平衡. 综上, 当前基于场景的测试方法在提升关键场景发现能力的同时, 逐步引入了资源感知优化与局部强化搜索策略, 但在复杂行为建模、动态场景适应性与高效性等方面仍有改进空间.

强化学习能够自适应学习和优化场景搜索, 因此使用强化学习实现的基于场景的自动驾驶测试得到研究人员的广泛关注. Feng 等人[7]提出了 D2RL, 利用深度强化学习技术, 从真实驾驶数据中学习安全关键事件, 训练神经网络以加快 AV 在现实场景下的测试效率. 强化学习方法往往忽视了模型预测中的不确定性, 可能输出危险的决策结果. 为解决该问题, Diehl 等人[22]提出了一种针对不确定性感知的基于模型的离线强化学习框架. 该方法在训练过程中集成不确定性估计, 增强了决策模型对潜在风险的感知; 利用历史数据构建环境模型, 通过预测模型的不确定性指导策略优化, 在降低风险的同时提高决策效率, 提高生成场景的真实性. Lu[23]提出了 RLTester, 采用强化学习实现关键的环境配置, 以揭示 AV 的不安全行为, 能够生成多样化的测试场景.

上述基于场景的测试方法虽然在生成和模拟多样化驾驶场景方面具有优势, 但往往忽视了 VCS 对 ADS 的影响, 即由于 VCS 改变, 影响到 ADS 对 AV 的控制. 为解决该问题, Yin 等人[8]使用遗传算法对 ADAS 进行测试, 证明了车辆配置及其相互作用对 ADAS 具有重要影响. 本课题组已提出的 SAFEVAR 方法[9], 针对特定驾驶场景验证不同 VCS 对 ADS 的影响, 验证了遗传算法在搜索暴露潜在 ADS 安全隐患的 VCS 的有效性, 分析了不同场景下不同 VCS 对 AV 的影响. 然而, SAFEVAR 使用传统的搜索算法及其算子, 运行过程需要调用自动驾驶模拟器

对每一个搜索到的 VCS 进行仿真验证, 消耗大量的计算成本; 亦存在搜索获得的解集间差异小, 搜索过程覆盖率低的问题.

1.2 基于模糊测试的自动驾驶测试

模糊测试是一种主要用于检测安全漏洞的测试技术, 通过向软件中注入异常的测试输入, 引导软件出现异常行为. 模糊测试方法广泛应用在代码检查、图像处理、API 函数测试领域[24−26], 通过集成深度学习技术, 实现了高效的测试用例生成. ADS 中集成了神经网络及深度学习模型, 因此模糊测试方法可应用于 ADS 测试. 根据测试数据的生成方式, 可分为基于生成的模糊测试与基于变异的模糊测试. 基于生成的模糊测试使用预定义规范生成测试数据; 基于变异的模糊测试针对合法输入数据, 通过一系列随机变化生成新的测试用例.

Li 等人[3]提出了 AV-FUZZER 测试框架, 用于识别 AV 的安全违规行为; 该框架通过扰动交通参与者的行为, 并使用遗传算法对扰动行为进行搜索, 实现关键场景的识别. 然而, AV-FUZZER 测试过程仅关注 AV 的起点与终点位置, 而非测试过程整体的安全性. 为解决该问题, Sun 等人[27]提出了面向驾驶过程的规范语言, 用以描述交通场景, 并使用模糊引擎实现了关键场景的搜索. 在 AV-FUZZER 的基础上, Kim 等人[4]提出了基于模糊测试的框架 DriveFuzz, 将待测 ADS 接入模糊测试框架, 通过插入鲁莽驾驶输入, 如过度加速, 扰动待测场景, 使用驾驶安全评估指标引导后续测试场景生成. Zhong 等人[28]提出了 AutoFuzz, 利用模拟器生成初始复杂驾驶场景, 并采用基于神经网络的搜索方法改进搜索过程. 该方法通过训练神经网络, 从搜索过程中选择和变异场景, 能够有效地识别交通违法行为. 然而, 作为一种黑盒方法, AutoFuzz 无法确定这些违规行为的触发原因.

模糊测试也广泛应用于针对联网车辆的 ADS 测试. Moukahal 等人[29]提出了一种面向漏洞的模糊测试框架 VulFuzz, 利用专为联网的 AV 设计的安全漏洞, 诱导最易受攻击的组件进行模糊测试. 在此基础上, Moukahal 等人[30]改进并提出了混合模糊测试框架 VulFuzz++, 为自动驾驶测试工程师提供了可靠的安全测试工具. 与 VulFuzz 相比, VulFuzz++能够在无法探索不同测试用例时终止, 并检查未遍历的分支, 根据暴露漏洞的可能性进行优先排序重启测试过程, 进而实现高效的测试覆盖.

上述基于模糊测试的研究, 通过扰动测试场景或搜索过程, 实现了高风险的关键驾驶场景构建. 然而, 这些工作虽然能保证搜索到的场景间具有差异性且有较好的覆盖率, 但未考虑 VCS 对 ADS 的影响.

2 基础知识

SAFEVCS 是一个模糊测试增强的 VCS 搜索方法, 其核心为设计不同的搜索算子. SAFEVCS 针对交叉、变异等算子进行改进, 以契合自动驾驶测试对 VCS 搜索的需求. 为量化描述 AV 的行为, SAFEVCS 使用自动驾驶安全评估指标, 评估关键场景, 并指导 SAFEVCS 的搜索过程. 对 SAFEVCS 中相关概念和基础知识的介绍如下.

2.1 遗传算法算子

遗传算法是一种通过模拟自然进化搜索最优解的搜索算法. 遗传算法算子在算法运行过程中, 指导其探索搜索空间、生成待测种群、评估算法运行, 是遗传算法实现搜索的核心. 常见的搜索算子, 包括选择算子、交叉算子、变异算子等. 同时, 遗传算法的搜索过程需要恰当的去重策略和终止策略, 以限定算法的搜索过程, 通过减少冗余计算, 并确保算法在合理的时间内收敛, 来提高算法运行效率. 在 ADS 测试中使用遗传算法, 首先需要对待测场景进行编码, 随后使用遗传算子针对搜索目标进行选择、交叉、变异等操作, 在自动驾驶模拟器的仿真测试过程中生成不同的关键场景.

选择算子用于从当前的种群中抽取个体进行交叉与变异, 决定哪些个体将继续繁衍. 在基于搜索的 ADS 测试中, 选择算子能够根据给定测试场景的适应度, 使用不同的选择策略决定子代待测场景. 常见的选择算子包括: 轮盘赌选择、锦标赛选择. 轮盘赌选择根据个体的适应度决定个体的选择概率, 锦标赛选择方法通过划分个体组, 选取组内适应度最高的个体; 通过引入竞赛压力, 锦标赛选择能够对遗传算法的收敛速度进行设置.

交叉算子用于组合不同的父代个体生成子代个体, 是搜索算法的核心. 合适的交叉算子能够加快算法的搜索过程. 常见的交叉方法, 包括二进制交叉 (simulated binary crossover, SBX)、点交叉、指数交叉[31]. 点交叉选择两

条父代个体的随机位置分割, 将两个个体按照分割位置进行交换. 基于点交叉的 SBX 算子, 能够模拟二进制串的单点交叉, 将其作用于以实数表示的父代个体.

变异算子用于随机改变某个个体的部分基因, 以增加种群的多样性, 协助搜索算法跳出局部最优, 提高种群的覆盖范围. 常见的变异算子, 包括多项式变异、位翻转变异等. 多项式变异使用多项式分布来生成变异量, 保证变异后的种群仍符合一定的概率分布特性. 位翻转变异是一种常用于二进制编码的变异方法, 其特点是随机翻转个体的一个或多个位置, 从而提升新种群的多样性.

搜索算法中常见的去重策略, 包括个体检查、相似性度量. 个体检查是检查新生成个体是否已经存在于种群中. 相似性度量使用欧氏距离等方法判断当前个体与种群其他个体的相似性. 终止策略是在计算每一代种群适应度后, 决定算法的运行状态. 常见的终止策略, 包括最大代数、多样性度量、种群规模等.

2.2 ADS 测试安全性评估指标

ADS 测试的安全评估指标, 根据参考的时间或距离可划分为基于时间的近端指标和基于距离的近端指标. 其中, 基于时间的近端指标主要关注最近的时间段内 AV 的驾驶安全性表现, 能够细粒度的描述 AV 的行为状态. 恰当的安全指标度量选取, 能够准确刻画 AV 的运行过程, 有助于加快搜索过程.

基于时间的近端指标关注当前速度和路径下, 车辆与物体的交互时间. 其中基于碰撞时间 (time to collision, *TTC*)、基于碰撞暴露时间 (time exceed *TTC*, *TET*) 和碰撞时间积分 (time integrated *TTC*, *TIT*) 用于评估车辆碰撞风险[32]. *TTC* 的计算方法如公式 (1) 所示, 其中 L_s 表示车辆与物体的距离, v_L 表示车辆与物体间的相对速度.

$$TTC(t) = \frac{L_s}{v_L} \tag{1}$$

在此基础上, 计算得到 *TET*、*TIT* 安全指标. *TET*、*TIT* 是 *TTC* 的细化指标, 能够计算一段时间内的车辆安全性, 计算方法如公式 (2)–(4) 所示. *TET* 计算车辆运行中超出 TTC^*的时间, TTC^*为 *TTC* 的阈值, 不同的场景复杂度下阈值设置不同, 一般为 1–3 s, τ_{sc} 为传感器收集数据的时间步长, *T* 为一次测试周期时长, *t* 为当前所处的采样时刻. *TIT* 侧重于描述车辆的碰撞风险程度, 计算超出 TTC^*的积分面积.

$$TET_i^* = \sum_{t=0}^{T} \delta_i(t) \cdot \tau_{sc} \tag{2}$$

$$\delta_i(t) = \begin{cases} 1, & \forall 0 \leqslant TTC_i(t) \leqslant TTC^* \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$TIT^* = \sum_{t=0}^{T} [TTC^* - TTC(t)] \times \tau_{sc} \tag{4}$$

为了更直观地刻画车辆的安全性, 文献 [8,9] 中使用了 *safetyDegree* 指标计算车辆的安全度. 该指标使用正负值来标记车辆是否发生碰撞, 数值大小能够描述碰撞的强度或潜在碰撞发生的距离. 具体来说, 若没有发生碰撞, *safetyDegree* 的取值为车辆与物体之间的距离; 若发生碰撞, *safetyDegree* 取值为碰撞发生时车辆与碰撞物体的相对车速 *cs* 的负值, 计算方法如公式 (5) 所示:

$$safetyDegree(res) = \begin{cases} distance, & \text{if } distance > 0 \\ -cs, & \text{if } distance \leqslant 0 \end{cases} \tag{5}$$

在涉及刹车的测试场景下, 需要对车辆减速过程的整体制动性能进行考量, 本研究使用平均减速度 (*aveDece*) 指标. 具体计算方法如公式 (6) 所示:

$$aveDece = \frac{\Delta v}{\Delta t} \tag{6}$$

2.3 模糊测试与遗传算法的结合

模糊测试[33]是一种自动化测试技术, 根据对测试目标的了解, 可以分为白盒模糊测试、灰盒模糊测试和黑盒模糊测试. 模糊测试的主要目标是揭示测试目标在处理非预期输入时的行为, 尤其是在处理边界条件和异常情况时的表现. 模糊测试的优点在于测试用例的随机性和覆盖广泛性, 能够在短时间内生成大量测试用例, 从而最大限

度地挖掘待测系统的潜在缺陷. SAFEVCS 使用黑盒模糊测试, 即不关注 ADS 的内部具体实现; 在测试数据的生成中, 使用变异模糊的方法指导测试过程.

遗传算法[34]是一种基于自然选择和遗传学原理的优化算法. 常用的遗传算法, 包括多目标遗传算法、精英保留遗传算法等. 遗传算法适用于复杂、多维的优化问题, 能够在庞大的搜索空间内快速找到近似最优解. 其核心思想是从初始种群出发, 通过多代的迭代演化, 逐渐逼近问题的最佳解决方案.

模糊测试能够实现较高的测试覆盖率, 但由于缺乏明确的搜索方向, 会产生冗余的测试用例. 遗传算法具有明确的搜索方向, 能够在目标函数的引导下逐步优化问题解空间, 然而在搜索过程中, 遗传算法容易陷入局部最优, 导致搜索获得的解集个体间缺乏差异.

SAFEVCS 通过结合黑盒模糊测试、变异模糊测试, 对遗传算法进行了优化. 在搜索中使用黑盒模糊测试, 结合变异模糊测试的数据生成方法, 对遗传算法的交叉和变异算子进行了改进. 通过引入终止策略和去重策略, SAFEVCS 能够引导遗传算法在多个维度上探索问题解空间, 跳出局部最优, 实现更高的搜索效率和覆盖率.

3 基于模糊测试的 ADS 关键车辆配置搜索

为解决 SAFEVAR 搜索获得的 VCS 相似程度高且缺乏合理性、搜索效率低等问题, 本研究提出了 SAFEVCS. 该方法以遗传算法为基础, 结合模糊测试对搜索过程进行改进优化, 集成自动驾驶仿真模拟器验证特定关键场景下的 ADS 表现, 旨在发现导致潜在 ADS 安全隐患的 VCS.

如图 1 所示, SAFEVCS 主要由 3 个部分组成, 即结合模糊测试的安全攸关车辆配置搜索方法、车辆配置搜索问题及待测 ADS 与仿真环境.

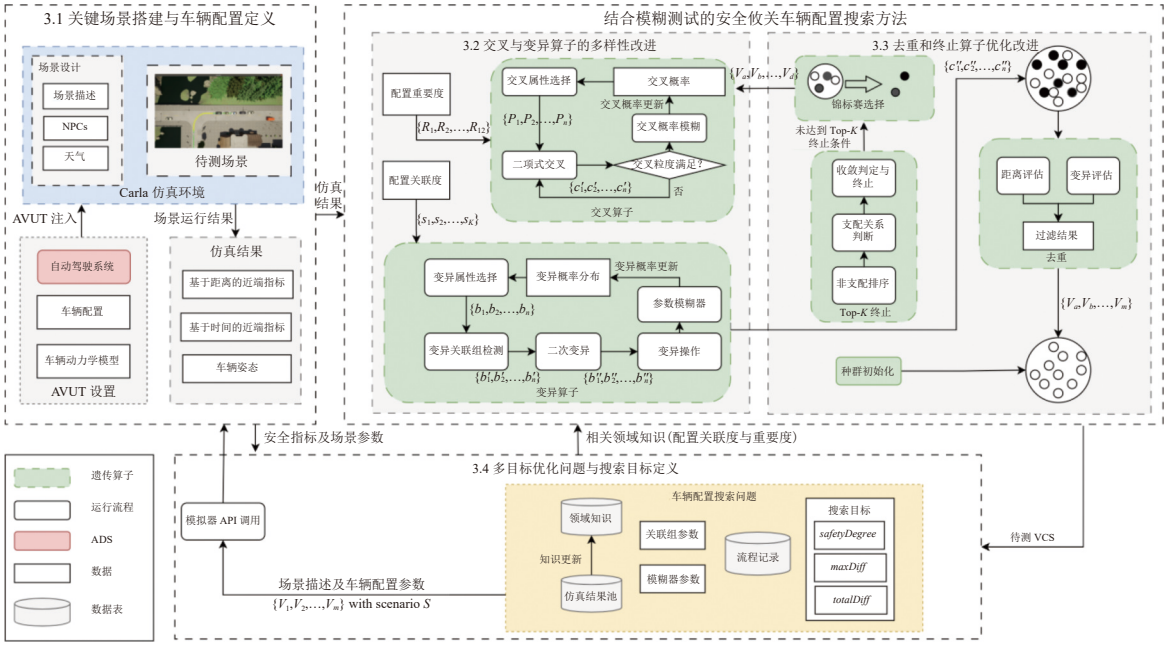


图 1 SAFEVCS 整体研究框架

首先, 根据领域知识和历史实验数据, 初始化车辆配置搜索问题, 驱动搜索方法进行 VCS 种群初始化, 将产生的待测种群输入到搜索问题中. 搜索问题连接自动驾驶模拟器, 设置待测 ADS、配置车辆动力学模型, 将初始化的 VCS 载入 AVUT. 设计并加载场景描述, 生成待测场景并进行仿真实验, 分析 VCS 待测种群的运行结果. 根据仿真结果, 收集近端安全性评估指标及车辆运行状态, 传入搜索问题进行评估与处理. 使用优化后的遗传算法对搜索过程进行评估, 判断搜索过程是否终止. 若终止, 则保存实验记录, 停止 SAFEVCS 运行. 若未终止, 则对父代种

群进行选择, 使用基于模糊测试的交叉算子和变异算子对父代进行交叉变异, 通过引入更多的随机性, 实现对种群的模糊操作. 在获取到子代种群后, 对其进行去重操作, 生成完整的子代种群. 最后接入搜索问题, 循环 VCS 的搜索.

为优化搜索过程, SAFEVCS 采用一个基于多目标优化的车辆配置搜索问题, 该搜索问题关注以下 3 个目标:AVUT 的驾驶安全, VCS 的最大改变百分比, 及 VCS 累计改变百分比. 根据上述 3 个目标, 搜索过程在 ADS 安全性与车辆配置变化幅度之间进行权衡, 从而在不同解中寻找最优解集.

3.1 关键场景与车辆配置定义

基于搜索的自动驾驶测试中, 关键场景需要恰当的形式化定义, 以确保场景的准确描述和测试过程的有效性. 参考文献 [35] 中的定义, 关键场景应描述为场景元素、交通参与者, 在一段时间步长的动作和事件输出. 因此, 本研究将关键场景 S 定义为六元组 $\{E,C,I,T,A,O\}$, 其中 E 是驾驶环境中的一组特征 (如行人、交通指示灯、行车线等), C 代表 AVUT 的 VCS. I 是该场景下的输入参数, 如传感器数据、地图数据、实时交通数据等. T 表示一段时间步长, 是场景中的时间要素. A 是场景 S 下的对应动作输出空间, 表示车辆的操作, 如加速、减速、转弯等. O 表示该场景下执行 A 动作后在 T 时间步长的输出状态, 包括车辆运行的安全指标、车辆状态.

SAFEVCS 中, AVUT 的 VCS 是一个 n 维向量 $C=\{C_1,C_2,\ldots,C_n\}$, 即 n 个可供调整的配置项. 任一 C_i 的值位于给定的域 $D_i=[l_i,u_i]$ 之内, 取值范围为 $|u_i-l_i|$. 例如, AVUT 的质量配置项可以在 2 040 –2 700 kg 之间变化, 其范围域为 $D_i=[2\,040,2\,700]$. $AVUT_C$ 表示具有 $VCSc$ 的待测车辆, 由多个赋予具体取值的配置项构成, 如 $AVUT_v$ 表示 AVUT 具有 VCS_v, 其中配置项共有 n 个, 每个配置项取原始值 $[v_1,v_2,\ldots,v_n]$.

3.2 交叉与变异算子的多样性改进

在 SAFEVAR 方法中, 搜索 VCS 的过程使用固定的交叉概率与变异概率, 这种固定的搜索方式能够公平地搜索不同的 VCS 组合. 然而, 在搜索过程中, 固定的交叉变异概率生成的 VCS 间差异有限. 为解决该问题, SAFEVCS 针对交叉与变异算子进行改进. 保证整体变化分布可控的情况下, 使用模糊测试提高个体的多样性与搜索过程的合理性. 具体而言, 通过结合模糊测试与搜索算法的交叉与变异过程, 探索 VCS 多样性组合下的 AVUT 表现; 在搜索过程中动态调整交叉和变异的概率, 优化 VCS 的生成过程. 结合模糊测试与搜索算法的优势, SAFEVCS 不仅可以探索到较容易暴露 ADS 安全隐患的 VCS, 亦能找到出现概率极低的但可能暴露 ADS 安全隐患的 VCS, 从而更高效地识别 ADS 的潜在隐患.

3.2.1 交叉过程

车辆不同配置项对 ADS 的影响是有差别的, 在交叉的过程中, 需要对配置项设置恰当的交叉概率, 以提高对 VCS 的搜索能力. 为更加接近搜索过程中的帕累托前沿, SAFEVCS 对不同配置项的交叉概率进行调整, 即针对不同的配置项 C_i, 引入了交叉概率模糊策略, 该模糊策略依据搜索的不同目标设置不同的概率分布, 实现差异化的交叉概率选择. 具体交叉过程如图 1 整体研究框架中 3.2 部分交叉算子所示.

通过指定各个配置项 C_i 的交叉概率 $probC_i$, $i\in\{1,\ldots,n\}$. VCS_C 的交叉概率可以表示为 $\overline{probC}=\{probC_1,\ldots,probC_n\}$. 针对不同配置项生成的交叉概率, 能够有效区分其重要程度, 进而针对性地对搜索空间进行探索.

在交叉过程中, 挑选 VCS_{C_i} 与 VCS_{C_j} 两个个体进行交叉, 为保证子代的差异性, SAFEVCS 通过计算个体间标准化后的欧氏距离, 来对父代进行筛选, 决定是否使用这两个父代进行交叉.

在生成 $\overline{probC}$ 并选择符合交叉条件的父代后, 将使用 SBX 进行交叉操作, 获得交叉后的子代 VCS_{C_k}. SAFEVCS 将对子代与父代进行对比, 记录交叉点的位置, 对 $\overline{probC}$ 的对应交叉点进行更新. 为提高交叉位置的多样性, 将减少交叉点位置 i 对应的配置项的交叉概率, 并对未变化的配置项提高交叉概率, 具体计算方法如公式 (7) 所示.

$$probC_i=\begin{cases} probC_i-\dfrac{\sigma}{1+\dfrac{\Delta_i}{\max(\Delta)}}, & \text{if } \Delta_i>0 \\ probC_i+\dfrac{\sigma}{1+\dfrac{\Delta_i}{\max(\Delta)}}, & \text{if } \Delta_i<0 \\ probC_i, & \text{if } \Delta_i=0 \end{cases} \tag{7}$$

其中, $probC_i$ 表示索引为 i 的配置项的交叉概率, Δ_i 是交叉点位置 i 的变化量, max(Δ) 是所有变化量的最大值, σ 是浮动因子, 能够对交叉的概率进行反馈调节. 为保持不同代之间的交叉独立性, 在每一代的子类交叉完毕后, 交叉概率模糊策略的 $\overline{probC}$ 将重新覆盖为初始交叉概率再进行交叉操作.

3.2.2 变异过程

SAFEVCS 中采用的具体变异过程如图 1 中 3.2 部分的变异算子所示. 在针对 VCS 的搜索中, 某个配置项的改变可能会引起其他相关配置项的变动. 因此应当重视同一系统, 如传动系统、刹车系统等, 间配置项的关联, 以实现更加合理的变异操作. SAFEVCS 对变异过程进行改进, 以提高种群的多样性与变异合理性. 变异算子的具体实现如算法 1 所示.

算法 1. SAFEVCS 变异算子.

输入: 个体矩阵 X, 下界 x_l, 上界 x_u, 变异概率 $probM$, 关联变量 $connected\_var$;
输出: 变异后的个体矩阵 X_p.

1. 初始化个体数 n 和配置项个数 m, 初始化输出矩阵 X_p
2. 生成一个 $n \times m$ 的变异掩码矩阵 mut, 其中 $mut[i, j]$ 由概率 $probM_i$ 确定
3. **for** 每个关联组 $g \in connected\_var$ **do**
4. **for** 每个个体 i **do**
5. **if** 关联组内任意位置变异 **do**
6. 重新对未变异位置 $mut[i, j]$ 进行变异尝试, 由概率 $probM_i$ 确定
7. **end for**
8. **end for**
9. 将 X_p 初始化为 X 的副本
10. **for** 每个个体 $X_i \in X$ **do**
11. **for** 每个属性 j **do**
12. **if** $mut[i, j]$ **do**
13. 计算变异比例矩阵 δ_1 和 δ_2
14. 生成变异方向和变异量矩阵 δ_q, 计算新的配置项值 j'
15. 修复异常值, $X_{p_{i,j}} = j'$
16. **end for**
17. **end for**

变异过程中, 需要计算各配置项的变异概率 $\overline{probM}$. 由于 $\overline{probM}$ 一定程度上决定了搜索过程的种群多样性, 恰当的变异概率能够帮助搜索过程跳出局部最优解, 实现更高的搜索覆盖率.

搜索初始阶段的变异概率 $probM_i$ 由公式 (8) 定义. x_i 表示配置项 C_i 的频次排名, *Sigmoid* 函数将配置项的频次排名映射到区间 [0, 1] 上, 使用 ω 因子进行缩放, 控制生成概率的合理性, 具体计算方法如公式 (8) 所示:

$$probM_i = \omega \cdot Sigmoid(0.5 \cdot x_i) \tag{8}$$

随后, SAFEVCS 将针对给出的 $\overline{probM}$ 生成变异掩码矩阵 mut, 该矩阵存储对应位置的变异标志. 由于单一的配置项变化往往会引起其他配置项的变化, 如轮胎直径增加, 会引起车辆抓地力上升, 因此 SAFEVCS 引入关联变量组的概念, 即当关联配置项 C_j 发生变异, 与之相关的 C_{j+1} 也可能会发生变异. SAFEVCS 使用多项式变异的方法, 对具体的配置项进行变异. 首先, 计算变异比例矩阵 δ_1 和 δ_2, 矩阵中各个体的配置项的变异比例 $\delta_{1[i,j]}$ 和 $\delta_{2[i,j]}$, 计算如公式 (9) 所示, 其中 $\delta_{1[i,j]}$ ($\delta_{2[i,j]}$) 表示个体中的各配置项下界 (上界) 与当前值的比例, 用于确定配置项的变异方向和变异量的大小.

$$\delta_{1[i,j]} = \frac{X_{[i,j]} - xl_j}{xu_j - xl_j},\ \delta_{2[i,j]} = \frac{xu_j - X_{[i,j]}}{xu_j - xl_j} \tag{9}$$

接着, 随机生成 [0, 1] 区间内的随机数, 再由该随机数的取值确定变异量矩阵 δ_q 的大小和变异方向. 最后, 根据变异量矩阵 δ_q 的大小和变异方向, 计算新的配置项值 $X'_{[i,j]}$. 在生成新的配置项后进行检验, 确保配置项在其固定取值区间之内.

3.3 去重与终止算子的优化改进

SAFEVCS 需要对每次搜索到的 VCS 进行仿真评估, 该过程会消耗大量的计算资源. 使用去重与终止算子, 能通过减少冗余个体运行次数, 避免无效的仿真运行, 在搜索收敛时及时终止搜索过程, 从而提高搜索效率. SAFEVCS 对去重与终止算子进行了针对性的改进. 去重与终止算子的工作过程如图 1 中的 3.3 部分所示.

3.3.1 去重算子

恰当的去重算子能够增加种群, 即 SAFEVCS 中 VCS 间的差异. 常用的基于距离的去重方法实现简单, 只需对不同 VCS 的欧氏距离进行累加计算. 然而这种方法忽视了不同配置项间的差异. 此外, ADS 对不同配置项变化的感知敏感度存在差异, 部分配置即使发生微小变化, 对 ADS 不会产生影响. 因此, 单纯基于欧氏距离的去重策略在实际应用中可能存在不合理性. 为解决该问题, SAFEVCS 引入了变动阈值及模糊边界的阈值设计, 实现去重操作.

具体来说, 去重算子针对不同的配置项 C_i 计算其阈值 Th_i, 该阈值根据配置项的取值范围和缩放因子 β 计算. 具体计算方法如公式 (10) 所示:

$$Th_i = \beta_i \times (u_i - l_i),\ D_i = [l_i, u_i] \tag{10}$$

其中, u_i 和 l_i 分别为配置项 C_i 的最大取值与最小取值. 参考范围 D_i 的区间, β_i 由配置项取值范围确定. 在确定了每个配置项的 β_i 后, 针对每个个体, 将检查该个体与本代内其他个体之间的差值是否在设定阈值之内, 具体计算方法如公式 (11) 所示:

$$isEqual(a,b) = |a_i - b_i| \leqslant Th_i,\ \forall i \in \{1,\ldots,n\} \tag{11}$$

其中, a 代表待测个体, b 代表同一种群中的其他个体. 如果任意配置项的差值超过其阈值, 则认为 a 与 b 是不同的, 从而获得保留, 否则抛弃 a, 并在本代种群评估后, 重新生成并补齐个体平衡种群规模. 最终, SAFEVCS 获取本代的种群, 将该代种群作为输入, 在模拟器中运行, 计算每个个体的适应度函数, 以获取到对应的运行结果, 以实现对其的安全性评估.

3.3.2 终止算子

SAFEVCS 使用两种终止算子度量终止条件, 包括固定代数终止、Top-K 终止. 多个终止条件的组合能够从不同维度评估搜索过程的运行状态, 以便在恰当的时机终止搜索.

在 SAFEVCS 中, 搜索过程将在任意一个终止条件被满足后终止. 其中, Top-K 终止通过对每代种群中的所有个体进行非支配排序, 维护一个 Top-K 列表记录当前最优的 K 个个体. 在每一轮迭代结束后, 如果该代种群中有 Top-K 个个体能够支配原 Top-K 列表中的个体, 则认为搜索过程尚未收敛, 更新 Top-K 队列并重置停滞计数. 否则, 计算 Top-K 列表未更新的代数, 若连续固定代数未更新, 则判定搜索过程收敛, 终止搜索过程.

相比传统只采用固定代数终止的方式, 结合基于 Top-K 列表变化检测的终止策略能够更灵敏地反映搜索过程中的停滞趋势, 提高终止搜索的合理性与鲁棒性.

3.4 多目标优化问题与搜索目标定义

为驱动上述搜索过程, 需要定义相应的多目标优化问题. SAFEVCS 提出以下搜索目标: 通过对待测 VCS 造成较小的扰动, 引起 AVUT 最大程度的安全性下降. 即针对场景 S, 在 $AVUT_v$ 下, AVUT 能够保证一定程度的安全性. 通过扰动 AVUT 的 VCS, 生成 $AVUT_{v'}$, 在原先的场景 S 下, 引起其安全性的下降甚至发生碰撞.

3.4.1 多目标优化问题表示

针对 $AVUT_v$, 有 n 个可供修改的配置项 $C = \{C_1, C_2, \ldots, C_n\}$, 配置项的初始值为 $v = \{v_1, v_2, \ldots, v_n\}$. 由搜索算法分配恰当的配置值 $v' = \{v'_1, v'_2, \ldots, v'_n\}$, 异于初始值. 从而, AVUT 获得了以 $AVUT_v$ 为原始车辆配置, $AVUT_{v'}$ 为待测车辆

配置.

然而, 较小的 VCS 变动也会增加搜索过程的搜索范围, 且这些变动可能不会对 ADS 决策产生影响. 因此, SAFEVCS 在搜索问题中引入 VCS 模糊策略, 该模糊策略通过限定配置项的变动范围, 过滤掉相似的 VCS 扰动. 具体来说, 该模糊策略计算 v_i 与 v_i' 的绝对差值 $|v_i - v_i'|$, 当 $|v_i - v_i'|$ 小于 Th_i, 认为针对 C_i 的扰动可忽略, 将 C_i 恢复为其初始值 v_i. 否则, 更新其值为 v_i'. 扰动因子依据配置项的初始值决定, 具体计算方法如公式 (12) 所示:

$$filter(v_i) = \begin{cases} C_i = v_i, & \text{if } \left|v_i - v_i'\right| < Th_i \\ C_i = v_i', & \text{otherwise} \end{cases} \tag{12}$$

据此, 搜索问题获得了经过模糊后的配置组合 $v'' = [filter(v_1'), \ldots, filter(v_n')]$.

3.4.2 搜索目标函数定义

仅关注 VCS 的安全性而忽视 VCS 的变化程度, 可能会使 VCS 变动过大. 在 SAFEVAR 的基础上, SAFEVCS 进一步对搜索目标做出约束: 尽量减小配置项变化值与原始值的差异. SAFEVCS 定义以下 3 个具体目标函数.

(1) 最小化 AVUT 安全性

safetyDegree 能够有效刻画 AVUT 在运行中的安全程度, 因此, SAFEVCS 使用其作为第 1 个目标函数, 即最小化 AVUT 安全性. 针对给定 $AVUT_{v'}$, 其适应度计算如公式 (13) 所示:

$$f_{\text{safe}}(v'') = safetyDegree(res) \text{ with } res = simulation(AVUT_{v''}, S) \tag{13}$$

其中, $AVUT_{v''}$ 为经过模糊后的 AVUT 具体配置, S 为给定的待测场景. 经过模拟器仿真运行后, 模拟器将返回相应的安全性指标集合 *res*.

(2) 最小化 $AVUT_v$ 中各配置项的最大变动百分比

为了找到配置项变化差异不过于大的配置组合, 将限定配置项的最大改变百分比, 记录给定 VCS 所有配置项的最大改变百分比的计算方法如公式 (14) 所示:

$$f_{\text{diff}}(v'') = \max_{i \in \{1,\ldots,n\}} \frac{\left|v_i - v_i''\right|}{v_i} \tag{14}$$

(3) 最小化 $AVUT_{v''}$ 的累计变动百分比

在限定不同配置项最大改变百分比的基础上, SAFEVCS 进一步计算每个配置项的改变百分比的累加和. SAFEVCS 考虑所有的配置项的改变量并通过关联配置组进行区分与计算. 具体来说, 将关联配置组内改变幅度最大的配置项作为引起配置组内关联变异的主要原因, 直接累加该配置项的改变百分比, 其他关联组内的配置项进行加权累加. 即设非关联配置组的配置项 $\{C_{m+1}, \ldots, C_n\}$ 的改变百分比为 $\Delta p_{m+1}, \ldots, \Delta p_n$; 设有关联配置组 $G = \{C_1, C_2, \ldots, C_m\}$, 对于组内的每个配置项 C_i, 其改变百分比为 Δp_i, 定义最大改变百分比为 $\Delta p_{\max}$, 其具体计算方法如公式 (15) 所示:

$$\Delta p_{\max} = \max(\Delta p_1, \Delta p_2, \ldots, \Delta p_m) \tag{15}$$

对于组内其他配置项进行加权累加, 具体计算方法如公式 (16) 所示:

$$\Delta p_{\text{weighted}} = \sum_{i=1}^{m} \frac{\Delta p_i}{m} \tag{16}$$

最终的累计变动百分比计算方法如公式 (17) 所示:

$$f_{\text{totaldiff}}(v'') = \Delta p_{\max} + \Delta p_{\text{weighted}} + \sum_{j=m+1}^{n} \Delta p_j \tag{17}$$

其中, $\Delta p_{\max}$ 是引起关联组内关联变异的主要原因, $\Delta p_{\text{weighted}}$ 是根据关联配置组的配置项个数进行的加权累加, 非关联配置组的配置项改变百分比 Δp_j 直接累加.

4 实验分析

为了对比 SAFEVCS 与基线方法 SAFEVAR 的表现, 设计并执行相关实验, 以回答如下 3 个研究问题 (research

question, RQ).

● RQ1: SAFEVCS 是否在各个评估指标上优于 SAFEVAR[9]?

● RQ2: SAFEVCS 的运行过程是否在收敛性和运行效率上较 SAFEVAR[9]表现更优?

● RQ3: 不同搜索方法下 VCS 搜索过程及结果是否存在一致性?

最后, 通过消融实验, 分析验证 SAFEVCS 中针对性改进方法的有效性.

4.1 实验设置与实验环境

4.1.1 AVUT 及驾驶场景设置

(1) 模拟器

真实的车辆动力学模型是 SAFEVCS 研究的基础. 为保证测试精度, 本研究使用 CARLA 模拟器[36]进行联合仿真实验, CARLA 使用了 NVIDIA 提供的车辆动力学模型, 能够真实模拟车辆在不同路况、不同天气下的驾驶场景. 作为较为通用的自动驾驶仿真模拟器, CARLA 广泛应用于训练 ADS、验证 ADS 安全性表现. 文献 [4,7–9] 均使用 CARLA 实现仿真实验.

(2) 待测系统

SAFEVCS 以 WOR (world on rails)[37]作为待测 ADS. WOR 从驾驶日志中学习基于视觉的交互式驾驶策略, 通过模仿学习与强化学习, 在 CARLA 公共排行榜 (https://leaderboard.carla.org/)、Town05 基线[38]、NoCrash 基线[39]、CARLA 42 条路径基线[40]中, 表现均优于类似的端到端 ADS, 如 LBC[37]、TransFuser[38]. SAFEVCS 延用 WOR 使用的训练车辆 LincolnMkz2017. 该车辆模型在 ADS 相关研究中应用广泛, 如文献 [41].

(3) 驾驶场景

本研究以美国国家公路交通安全管理局 (NHTSA) 发布的 37 个预碰撞场景为指导, 筛选验证 SAFEVCS 的仿真实验场景, 即“行人碰撞”与“跟车碰撞”这两个场景, 及其与晴雨天的组合. 因为, 行人是最有可能在交通碰撞中受伤甚至死亡的道路使用者, 而跟车场景则涉及前车突发减速、紧急制动等复杂动态变化, 具有较高的事故风险和挑战性. 根据文献 [42,43] 中的场景构建方法, SAFEVCS 选择了 CARLA 内置的 Town01 作为场景地图, 其场景如图 2、图 3 所示.

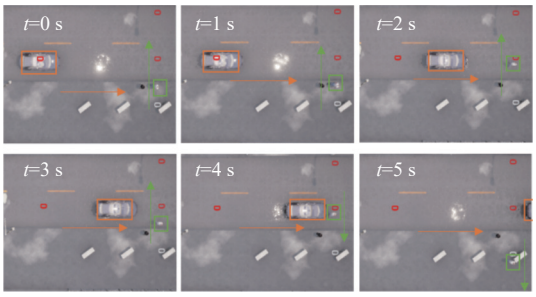


图 2 行人碰撞场景示意图

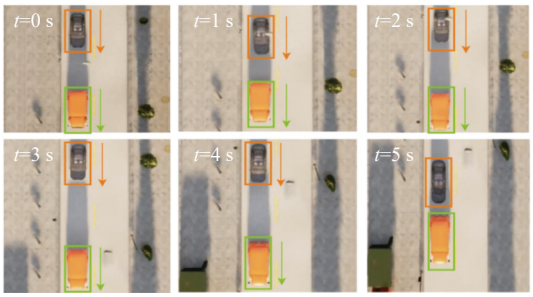


图 3 跟车碰撞场景示意图

“行人碰撞”场景运行过程为 AVUT 在晴天与雨天两种天气条件下, 从指定的起点驶向终点, 一名行人在 AVUT 行驶路径上的固定位置出现, 并沿预定轨迹横穿马路. 当面对横穿马路的行人时, AVUT 应紧急停车, 以避免撞到行人, 并等待行人通过马路后继续驶向终点.

“跟车碰撞”场景中, AVUT 在晴天天气条件下, 从指定的起点驶向终点. 其前方设置有一辆 NPC 车辆, 在行驶过程中会随机执行突发减速及紧急制动等行为. 面对前方车辆的突发状况, AVUT 需及时作出响应, 通过紧急减速或停车等操作避免发生碰撞, 并在前车恢复正常行驶后继续跟随, 直至安全到达终点.

“行人碰撞”场景中, 行人出现时机相对稳定, 且始终按照固定路径穿越道路, 有利于待测 ADS (WOR) 的提前感知与应对. “跟车碰撞”场景引入了动态交通参与者的非确定性行为, 以更全面地评估 WOR 在突发交通状况下的感知与决策能力. 在该场景中, 前方 NPC 车辆的减速与紧急制动行为依据其位置信息定点触发执行, 可有效模拟

真实道路上常见的突发事件, 如因障碍物、临时交通信号等, 前车突然减速甚至急刹. 且在“跟车碰撞”场景中, AVUT 始终被前车遮挡视野, 无法预知前车行为的动态变化, 对 WOR 的实时感知及快速响应有更高的要求. 文献 [44] 采用类似的跟车测试场景, 暴露了待测 ADS 在应对高时效性、非确定性交通参与者行为方面存在的潜在缺陷.

表 1 详细地展示了以上 3 个测试驾驶场景的配置信息, 具体包括场景的天气配置参数、NPC 的种类及数量、NPC 的行为特征等. 在感知遮挡复杂度上, 雨天行人碰撞场景因雨滴与水雾导致视觉感知能力下降, 而晴天跟车碰撞场景虽视觉条件良好, 但前车车辆类型为卡车, 且长期处于 AVUT 视野前方, 形成物理遮挡, 增加其对远距目标的感知难度. 因此, 晴天跟车碰撞场景对待测 ADS 造成更大的应对挑战.

表 1　仿真测试场景配置说明

场景	场景参数	NPC数量及类型	行为不确定性	感知遮挡复杂度
晴天行人碰撞场景	云量: 0 降水量: 0 雾密度: 0 太阳方位角: +90°	1 行人	NPC行为: 按预定轨迹穿越斑马线 行为可预测性: 高	视觉条件: 良好 遮挡因素: 无遮挡 感知难度: 低
雨天行人碰撞场景	云量: 100 降水量: 100 雾密度: 7 太阳方位角: +90°			视觉条件: 能见度下降 遮挡因素: 雨滴、水雾 感知难度: 高
晴天跟车碰撞场景	云量: 0 降水量: 0 雾密度: 0 太阳方位角: +90°	1 卡车	NPC行为: 突发减速、制动 行为可预测性: 低	视觉条件: 良好 遮挡因素: 前车遮挡 感知难度: 高

Mahmud 等人[45]研究了高危险驾驶场景下, *TTC* 阈值 TTC^* 的设置. 同时, van der Horst 等人[46]的研究表明, 当车辆接近交叉路口时, 理想的 TTC^* 为 1.5 s. 在晴天和雨天的场景中, 行人横穿马路形成了一个横向的交互场景, AVUT 直行, 可认作为一个交叉路口. 因此, 根据文献 [46] 的研究结果, 本研究将 TTC^* 设置为 1.5 s. 为了统一不同交互场景下的安全反应标准, 确保车辆能够在高风险情况下及时采取避险措施, 在跟车场景中亦设置 TTC^* 为 1.5 s.

(4) 车辆参数设置

参考 CARLA 提供的车辆配置项访问接口, 并便于与 SAFEVAR 进行对比, 本研究选择了同 SAFEVAR 中使用的 12 个配置项构成 VCS. 同时, 本研究中配置项的选择参照了车辆动力学领域的权威文献 [10] 中的相关论述, 确保选取的配置项是足以影响车辆操控性能与安全性的重要车辆动力学特性参数. 具体的 12 个配置项可以根据其功能归为以下 3 类.

● 动力系统相关配置项: 包括车辆发动机的最大转速 (max_rpm)、换挡时间 (gearSwitchTime)、离合器强度 (clutchStrength)、最大制动扭矩 (maxBrakeTorque).

● 离合器相关配置项: 包括油门最大时的阻尼比 (dampRateFullT)、油门为零且离合器接合时的阻尼比 (dampRateZeroT_CE)、油门为零且离合器分离时的阻尼比 (dampRate_zeroT_CD) 以及车轮的阻尼率 (dampRate).

● 物理特性相关配置项: 包括车辆的质量 (mass)、车辆底盘的阻力系数 (dragCoeff)、车轮的半径 (radius) 和车轮的摩擦系数 (tireFric).

以上配置项的初始值及取值范围详见表 2; 关于配置项的取值范围设置, 本研究参考了文献 [8,47] 中的实现验证.

参考车辆动力学领域知识与以上相关配置项的分类, 本研究设置了 2 组相关配置组: {dampRateFullT, dampRateZeroT_CE, dampRate_zeroT_CD, dampRate}和{mass, tireFric, radius}. 前面一个配置关联组中的 4 个配置项均为离合器相关, 能够直接影响 AVUT 在不同油门和离合器状态下的操作行为; 后面一个配置关联组中的 3 个配置项均涉及车辆的基本物理特性, 包括质量、轮胎摩擦系数和车轮半径, 这 3 个配置项的组合影响车辆的操控性能, 是日常条件下最容易发生变化的车辆配置, 如通过乘车人员增减、路面条件变化等. 通过引入上述两组关联配置项组, SAFEVCS 实现了对 VCS 更合理的扰动.

表 2　CARLA 中的车辆配置项范围与初始值

配置项 C_i	备注	原始值 $\bar{v}$	阈值区间 $D_i=[l_i,u_i]$	*prob_rank*
max_rpm (r/min)	发动机最大转速	5 800	[4200, 7000]	6
dampRateFullT (kg·m^2/s)	油门全开时阻尼比	0.15	[0.1, 0.2]	10
dampRateZeroT_CE (kg·m^2/s)	油门为零且离合器接合阻尼比	2	[1.0, 3.0]	7
dampRate_zeroT_CD (kg·m^2/s)	油门为零且离合器分离时阻尼比	0.35	[0.2, 0.4]	9
gearSwitchTime (s)	换挡时间	0.5	[0.3, 0.6]	4
clutchStrength (kg·m^2/s)	车辆离合器强度	10	[8.0, 12.0]	8
mass (kg)	车辆质量	2 404	[2040, 2700]	3
dragCoeff	车辆底盘阻力系数	0.3	[0.2, 0.5]	12
tireFric	轮胎摩擦系数	3.5	[1.0, 3.9]	5
dampRate (kg·m^2/s)	轮胎阻尼率	0.25	[0.20, 0.30]	11
radius (cm)	轮胎半径	35.5	[31.7, 37.0]	2
maxBrakeTorque (N·m)	最大制动扭矩	1 500	[1200, 1650]	1

4.1.2　搜索过程设置

(1) SAFEVCS 搜索过程参数设置

为提高 SAFEVCS 搜索过程的效率, 需要针对其集成的遗传算法内的算子及搜索问题的参数进行设置, 具体数值如表 3 所示. 设置过程的具体代码见 https://github.com/majianwei99/SAFEVCS.

表 3　SAFEVCS 搜索过程参数值

参数	数值
种群规模	50
最大运行代数	100
浮动因子 σ	0.025
缩放因子 ω	0.3
Top-K列表长度	50
终止代数	2

实验中, SAFEVCS 使用 NSGA-II 作为遗传算法的具体实现, 并基于 Pymoo[48]框架对交叉、变异等算子进行改进. 针对 NSGA-II 算法, 种群规模设置为 50, 最大运行代数设置为 100.

在多目标优化问题中, 配置项的重要性按顺序设置为: $prob\_rank=\{6,10,7,9,4,8,3,12,5,11,2,1\}$, $prob\_rank_i$ 对应 C_i 的配置重要性, 设置依据为文献 [9] 的实验结果. *prob_rank* 也可以依据测试目标进行调整, 实现对不同 VCS 的针对性搜索.

在交叉算子的参数选择中, 本实验使用标准正态分布计算 $\overline{probC}$, 通过预实验调整并设定浮动因子 σ 值为 0.025. 设置数值适中的 σ 有助于提高对应种群个体间的差异, 保证交叉稳定性.

在变异算子的参数选择中, $\overline{probM}$ 的计算依赖于配置项的频次排名, 具体由 *prob_rank* 设置. ω 缩放因子依据配置项的长度设置为 0.3, 经过预实验验证, 该缩放因子能够保证各个配置项的变异概率在合适的范围, 保证即使变异概率较低的配置项也有发生变异的可能性.

去重算子中, 阈值 *Th* 由配置项对应的取值区间范围确定, 参考文献 [8] 中的实验结果, SAFEVCS 使用文献 [8] 中推荐的精度值进行实验, 即公式 (10) 中 β_i 的定义如公式 (18) 所示:

$$\beta_i=\begin{cases}0.01, & D_i\in[1000,+\infty)\\ 0.02, & D_i\in[100,1000)\\ 0.04, & D_i\in[1,100)\\ 0.08, & D_i\in[0,1)\end{cases} \tag{18}$$

举例说明, 对于车辆质量配置项 mass, 由于其 D_i 范围为 [100, 1000), 因此其对应β_i 取值为 0.02, Th_{mass} 为 $0.02\times(2700-2040)=13.2$.

针对终止算子的设置, Top-K 列表的长度设置为 50, 与每代种群的规模相同. 由于仿真运行实验场景的耗时较高, 每代种群规模均为 50 且多目标优化问题涉及多个目标函数, 会导致生成大量的帕累托前沿, 因此参考 AV-FUZZER 设置并结合预实验结果将终止代数设置为 2. 即当连续两代 Top-K 列表均无更新时, 认为算法收敛.

(2) 对比基线

为验证 SAFEVCS 针对搜索过程的改进, 同时考虑到目前鲜有关注车辆配置搜索的自动驾驶测试方法, 本研究使用文献 [9] 给出的 SAFEVAR 方法作为对比基线. 为公平比较两个方法, 在同一版本模拟器下配置相同的仿真场景、待测 AVUT、搜索种群大小及最大运行代数等.

(3) 实验环境

实验的环境配置为 Linux Ubuntu 18.04.5 系统, 2.2 GHz Intel Xeon 处理器, RTX NVIDIA 2080Ti 显卡和 150 GB 内存.

4.1.3 搜索解集评估方法

为系统比较 SAFEVCS 与 SAFEVAR 在不同自动驾驶测试场景下搜索获得 VCS 解集的表现, 本研究采用反世代距离 (inverted generational distance, *IGD*) 和超体积 (hyper volume, *HV*) 作为评估指标, 并结合 Mann-Whitney U 检验与 Vargha and Delaney 检验, 从统计学角度衡量两种方法在各安全性评估指标, 如 TET 上的统计显著性差异与性能表现.

(1) 反世代距离 (*IGD*)

IGD 是一种常用于评估多目标优化算法性能的指标, 用于度量算法所得解集与帕累托前沿之间的接近程度. *IGD* 通过计算帕累托前沿上每一个点与算法解集中最近一点之间的欧几里得距离, 并对这些距离取平均, 反映解集在搜索空间中的分布密度与收敛性. 其数值越小, 说明当前解集越接近帕累托前沿, 表明算法具有更优的收敛性能和解质量. *IGD* 的具体计算方法如公式 (19) 所示:

$$IGD(P,P^*)=\frac{1}{|P^*|}\sum_{v\in P^*}\min_{u\in P}d(u,v) \tag{19}$$

其中, P 为算法当前运行获得的解集, P^* 为帕累托前沿, $d(u,v)$ 表示解 $u\in P$ 与 $v\in P^*$ 之间的欧几里得距离, $|P^*|$ 为帕累托前沿中解的数量.

(2) Mann-Whitney U 检验

Mann-Whitney U 检验是一种经典的非参数统计方法, 常用于比较两个独立样本集合 (对应本研究中 SAFEVCS 与 SAFEVAR 搜索得到的 VCS 解集) 在某一数值指标上 (对应本研究所采用的评估指标, 如 *IGD* 及 TET 等) 的差异是否具有统计显著性. Mann-Whitney U 检验不依赖样本的正态分布假设, 适用于小样本、非对称分布或含有离群值的情形, 因此被广泛应用于复杂实验数据分析. 给定两个独立样本集 X 与 Y, 对应的 U 统计量计算方法如公式 (20) 所示:

$$U_X=R_X-\frac{m(m+1)}{2},\ U_Y=R_Y-\frac{n(n+1)}{2},\ U=\min(U_X,U_Y) \tag{20}$$

其中, R_X、R_Y 分别为样本 X 和 Y 在合并排序后的秩值总和; m、n 分别为样本 X 和 Y 的样本容量; U 为最终用于检验的统计量. 最后, 将 U 计算转换为 z-score 值, 利用正态分布表可查询得到对应的 p-value 值; 若 p-value 小于 0.05, 则说明两种对比方法搜索得到的解集具有显著差异. 通过对 *safetyDegree*、*TIT*、*TET*、*aveDece* 进行 Mann-Whitney U 检验, 可以判断在每个单项指标上 SAFEVCS 与 SAFEVAR 的解集是否有显著差异. 例如, 在 *safetyDegree* 指标上若 p-value 小于 0.05, 则说明 SAFEVCS 与 SAFEVAR 所生成解集, 即测试场景的安全度存在显著差异, 表明 SAFEVCS 在引导生成更危险场景方面可能更有效. 反之, 若 p-value 大于或等于 0.05, 则表示两种方法在该指标下产生的解集不具有显著差异, 无法得出哪一种方法更优的结论. 然而 p-value 只能说明两组解集在某个指标下是否存在差异, 而无法反映差异的方向与大小. 因此在检验结果存在明显差异时, 需要进一步结合

Vargha and Delaney 检验判断哪种方法更优.

(3) Vargha and Delaney 检验

Vargha and Delaney 检验分析评估两种方法在某一指标下解集表现优劣. 具体的计算方法如公式 (21) 所示:

$$\hat{A}_{12}=\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\varphi(x_i,y_j),\ \ \varphi(x_i,y_j)=\begin{cases}1, & \text{if } x_i>y_j\\ 0.5, & \text{if } x_i=y_j\\ 0, & \text{if } x_i<y_j\end{cases} \tag{21}$$

其中, x_i 为 SAFEVCS 在某一指标 (如 *TET*) 下搜索获得 VCS 解集中的任意个体; y_j 为 SAFEVAR 在相同指标下 VCS 解集的任意个体; m、n 为两组解集的大小. 若解集间存在显著差异, 将利用 Vargha and Delaney 检验对比其 $\hat{A}_{12}$ 值. $\hat{A}_{12}$ 值用于衡量 SAFEVCS 相较于 SAFEVAR 生成更优解集的概率. 对于衡量测试场景风险水平的 *TET* 与 *TIT* 指标, 此类指标数值越大代表生成测试场景的风险水平越高, 因此 $\hat{A}_{12}$ 大于 0.5 表明 SAFEVCS 能够构造出风险更高、对 ADS 更具挑战性的测试场景. 对于 *IGD* 与 *safetyDegree* 这类反映解集质量与 ADS 安全性的指标, 指标数值越小代表性能越优, 因此 $\hat{A}_{12}$ 小于 0.5 表明 SAFEVCS 搜索解集质量更高.

(4) 超体积 (*HV*)

HV 是指搜索算法所得解集中的个体与目标空间中相应的参考点所围成的超立方体体积, 值越大代表算法所得解集的多样性越好. *HV* 的具体计算方法如公式 (22) 所示:

$$HV=\delta\left(\bigcup_{i=1}^{|X|}v_i\right) \tag{22}$$

其中, δ 代表 Lebegue 测度, 用来测量体积, $|X|$ 表示搜索算法得到解集的数目, v_i 表示解集中第 i 个个体与参考点所围成的超立方体.

4.1.4 实验设计

在实验开始前, 通过预实验完成对 SAFEVCS 搜索算法相关参数的设置, 确保 SAFEVCS 的运行过程符合预期设定. 同时, 针对 AVUT 的 VUT_{v} 在实验场景下进行测试, 重复运行 30 次以确保没有碰撞发生.

在实验过程中, 遵循文献 [49] 的方法, 为减少随机性对搜索过程的影响, SAFEVCS 与 SAFEVAR 方法均在两个“行人碰撞”场景运行 30 次, “跟车碰撞”场景运行 10 次, 对搜索获得种群中的每个个体进行仿真验证, 获得目标函数计算适应度, 并使用 *TIT*、*TET*、*aveDece* 指标进一步刻画场景运行过程. SAFEVAR、SAFEVCS 将在晴、雨天行人碰撞场景各获得 50 个/次×30 次=1500 个最优解, 在晴天跟车碰撞场景获得 50 个/次×10 次=500 个最优解, 其中 50 为对应算法的每一代的种群规模, 30 为文献 [49] 中指导的运行次数, 设定为该次数能够有效降低由于搜索算法的随机性而产生的解集不稳定的情况. 考虑到跟车碰撞场景仿真过程中涉及连续的复杂动态变化, 单次仿真时长及计算资源开销均高于行人碰撞场景, 实验中跟车碰撞场景中的重复运行次数设定为 10 次, 以平衡实验资源开销与获得实验数据量能够支撑统计分析与算法性能评估. 尽管运行次数有所减少, 但通过合理设定种群规模, 仍能有效抑制搜索过程中随机性带来的结果波动, 保证实验数据的稳定性与可重复性.

针对 RQ1, 为对比不同搜索算法的运行效果, 参考文献 [49], 首先对 SAFEVCS 与 SAFEVAR 的 IGD 和安全指标, 即 *safetyDegree*、*TET*、*TIT* 和 *aveDece*, 进行 Mann-Whitney U 检验, 以检验其最优解是否存在统计显著性差异; 若存在显著差异, 将进一步使用 Vargha and Delaney 检验对比其 $\hat{A}_{12}$ 值. 为度量二者的性能, 参考文献 [50], 将计算 SAFEVCS 与 SAFEVAR 所有重复运行生成的最优解集合来获取帕累托前沿.

针对 RQ2, 为检验 SAFEVCS 的收敛情况并对比算法的运行时间开销, 记录 SAFEVCS 与 SAFEVAR 在 3 个场景下的不同收敛趋势. 由于 *HV* 能够评估帕累托前沿的疏密度、多样性和分布来评估搜索算法的结果[51]. 因此, 将记录 *HV* 在算法运行过程的改变趋势及运行时间开销, 以对比评估两种方法的搜索过程.

为进一步验证搜索获得的 VCS 对 ADS 安全性的影响, 以佐证 VCS 对 ADS 的影响具有普适性, RQ3 中将针对两种方法不同配置项的累计变动百分比作频次分析, 对比不同场景、不同配置项在搜索过程中的变化情况.

4.2 实验结果与分析

4.2.1 RQ1 结果

为回答 RQ1, 本节首先针对 SAFEVCS 与 SAFEVAR 的指标显著性进行了对比, 随后针对不同配置项的累计变化百分比、不同指标的平均值进行分析.

(1) 指标显著性对比

如表 4 所示, 在 3 个场景中, 对比 SAFEVCS 与 SAFEVAR 在所有指标下的解集质量时, 除晴天跟车碰撞场景的 *aveDece* 指标外, 其余 p-value 均小于 0.05, 说明 SAFEVCS 与 SAFEVAR 搜索获得解集间有显著差异. $\hat{A}_{12}$ 效应值分为 4 个级别: 无差异 ((0.444, 0.556)), 差异较小 ([0.556, 0.638) 或 (0.362, 0.444]), 差异中等 ([0.638, 0.714) 或 (0.286, 0.362]), 差异明显 ((0.714, 1.0] 或 (0, 0.286]). 对于 *IGD* 与 *safetyDegree* 指标, $\hat{A}_{12}$ 效应值越小则说明 SAFEVCS 优于 SAFEVAR 的概率越高. 对于 *TET*、*TIT* 指标, $\hat{A}_{12}$ 效应值越大则说明 SAFEVCS 优于 SAFEVAR 的概率越高. 加粗数值说明 $\hat{A}_{12}$ 效应值处于差异中等或差异明显区间内.

表 4 SAFEVCS 与 SAFEVAR 搜索 VCS 解集的指标显著性比较

场景	*IGD*↓	*safetyDegree*↓	*TET*↑	*TIT*↑	*aveDece*
	p-value/ $\hat{A}_{12}$	p-value/ $\hat{A}_{12}$	p-value/ $\hat{A}_{12}$	p-value/ $\hat{A}_{12}$	p-value/ $\hat{A}_{12}$
晴天行人碰撞场景	**<0.01/0.215**	**<0.01/0.093**	**<0.01**/0.392	**<0.01**/0.430	**<0.01/0.255**
雨天行人碰撞场景	0.011/**0.301**	**<0.01/0.144**	**<0.01**/0.581	**<0.01**/0.633	**<0.01**/0.366
晴天跟车碰撞场景	**<0.01/0.080**	**<0.01/0.255**	**<0.01**/0.628	**<0.01**/0.179	0.305/0.474

注: p-value值小于0.01表明两种方法具有统计显著性差异

晴天行人碰撞场景中, p-value 在 *IGD* 指标上小于 0.01, 且 $\hat{A}_{12}$ 值落入差异明显区间, 表明 SAFEVCS 搜索到的解集中, 有很高的概率更加靠近帕累托前沿, 说明在多目标优化过程中 SAFEVCS 具有更强的收敛性与搜索稳定性. 在 *safetyDegree*、*aveDece* 两个指标上, $\hat{A}_{12}$ 均小于 0.286, 说明两个方法在这两个指标上差异明显, SAFEVCS 方法能够生成更小的 *safetyDegree* 与 *aveDece*, 意味着其能探索到更危险的解集空间. *TET* 与 *TIT* 数值差异较小, 这是由于 SAFEVCS 搜索获得的 VCS 能够更频繁地触发碰撞场景, 而 *TET* 与 *TIT* 只记录未碰撞场景数, 碰撞场景下 *TET* 与 *TIT* 取值为 0, 进而导致 *TET*、*TIT* 的平均数下降, 两种方法在 $\hat{A}_{12}$ 指标上体现的差异较小. 然而, 如表 5 所示, SAFEVCS 在去除了 0 值后, *TET*、*TIT* 的平均值更大, 说明在未碰撞的场景下, SAFEVCS 搜索到的 VCS 能够更充分暴露 ADS 的安全隐患.

表 5 SAFEVCS、SAFEVAR 与原始配置指标的平均值

场景	方法	*safetyDegree*	*TET*	*TIT*	*aveDece*
晴天行人碰撞场景	SAFEVCS	**−1.114**	**1.879**	**1.481**	**4.005**
	SAFEVAR	1.200	1.570	0.830	4.390
	VCS_v	2.700	1.200	0.440	5.970
雨天行人碰撞场景	SAFEVCS	**−0.712**	1.291	**1.773**	6.687
	SAFEVAR	0.280	**1.720**	1.020	**4.310**
	VCS_v	2.200	1.400	0.610	5.420
晴天跟车碰撞场景	SAFEVCS	**−1.561**	0.775	1.351	6.173
	SAFEVAR	0.246	0.811	**1.687**	5.930
	VCS_v	0.704	**1.627**	0.699	**6.319**

注: *TET*与*TIT*均为去除0值后的平均值

雨天行人碰撞场景中, p-value 在 *IGD* 指标上为 0.011, 略高于该场景晴天天气下的指标值. 其原因为: 由于场景的复杂性增加, 如光照条件改变、路面条件恶化、传感器 RGB 图像质量降低等因素的综合影响, SAFEVAR 中 AVUT 在雨天场景下安全性表现下降, 使得在晴天未能暴露出安全隐患的 VCS 在雨天中意外地触发了 ADS 的潜在安全性问题, 从而提升了 SAFEVAR 的有效搜索能力. 因 SAFEVAR 的 *IGD* 上升, 导致 SAFEVCS 与 SAFEVAR 的解集质量差距缩小 ($\hat{A}_{12}$ 从 0.215 升高到 0.301), 体现在 *safetyDegree* 及 *aveDece* 指标上即为两者的差异更小, 即

$\hat{A}_{12}$ 更接近 0.5.

晴天跟车碰撞场景中, p-value 在 *IGD* 指标上小于 0.01, 且 $\hat{A}_{12}$ 差异明显, 即 SAFEVCS 方法搜索得到的解集在更大概率上接近帕累托前沿, 说明其搜索能力更强. *safetyDegree* 指标下的 $\hat{A}_{12}$ 小于 0.286, 意味着 SAFEVCS 方法能够更有效地发现不安全的 VCS, 从而挖掘出更具挑战性的测试关键场景. 而在 *TET* 指标上, SAFEVCS 显著优于 SAFEVAR, 表明其生成的测试场景能够更早地促使 AVUT 触发紧急应对机制, 即更快地构造高风险的测试场景; 在 *TIT* 指标上 SAFEVCS 的表现不如 SAFEVAR, 这一结果说明 SAFEVCS 生成的场景能够更早、更频繁的暴露出 AVUT 在极端条件下的潜在漏洞. 对于 *aveDece* 指标, 两种方法的差异并不明显, 这主要归因于测试场景为一个典型的两阶段减速过程, 其中 AVUT 先以较低幅度进行预减速, 在突发情况出现后再进行紧急制动. 这种结构性行为使得 AVUT 在整个过程中的平均减速度趋于一致, 从而削弱了该指标对不同方法差异的敏感性.

(2) 指标平均值对比

如表 5 所示, SAFEVCS 在晴天行人碰撞场景中, 在所有的指标平均值上均优于 SAFEVAR 及初始配置 v, 说明在较为安全的场景下, SAFEVCS 也能搜索到暴露潜在 ADS 安全隐患的 VCS. 在雨天天气情况下, 由于场景复杂度提升, 部分之前安全的 VCS 能够诱发 ADS 安全隐患; 类似的实验结果, 隐藏了 SAFEVAR 在搜索过程中的不足, 导致 SAFEVCS 的对比优势下降. 值得注意的是, 如表 4 所示, 雨天行人碰撞场景下 *TET* 与 *TIT* 指标的 $\hat{A}_{12}$ 均大于 0.5. 在晴天跟车碰撞场景中, SAFEVCS 在 *safetyDegree* 指标上的表现优于 SAFEVAR 及初始配置 v, 表明其更擅长挖掘严重碰撞场景. 在 *TET* 和 *TIT* 两项时间类指标的均值上, SAFEVCS 低于 SAFEVAR, 反映出其能够在复杂测试场景下生成更多发生碰撞的测试场景, 而非仅识别出具有潜在高风险的场景. 相比之下, SAFEVAR 和初始配置虽然定位到了危险程度较高的场景, 但这些场景并不总是触发安全事故.

(3) 指标累计变化对比

后文表 6 为 SAFEVCS 与 SAFEVAR 在晴天行人碰撞场景下, AVUT 配置项的变动百分比结果. PC 表示各配置项在不同累计变化百分比区间内的平均百分比变化 (即 $|v_i - v_i''|/v_i$); Δ代表相对差异 (即 $v_i - v_i''$), 表明了配置项变化的方向 (即正或负) 与大小. 以 mass 配置项为例, 在 SAFEVCS 方法下该配置项正向变化 9.27% 为 222.97 kg. 首先收集所有的末代种群, 计算对应配置项的累计变动百分比, 接着对各个配置项及评估指标进行计算. 通过对比不同累计变化百分比在不同区间下的数值, 能够发现 SAFEVCS 有效降低了 AVUT 的安全性, 且对应的关联组内的配置项变化更为明显, 如 dampRateZero_CE, 在 SAFEVAR 中, 其变动有限, 仅在累计变动百分比提高到 200% 时才能进入优先解集中. 在较小的变动累计变动百分比下, 由于关联配置组的引入, dampRateZero_CE 改变的幅度与百分比有所提升. 说明关联配置组的引入, 能够使原先独立的配置项搜索过程发现到更多容易暴露 ADS 风险的 VCS, 导致 AVUT 出现不安全行为, 证明 SAFEVCS 比 SAFEVAR 的搜索更具有效性.

综上, 针对 RQ1, 使用 p-value 及 $\hat{A}_{12}$ 对 SAFEVCS 与 SAFEVAR 进行了指标的显著性对比, 在晴天与雨天行人碰撞场景、晴天跟车碰撞场景下, SAFEVCS 能够搜索到更接近帕累托前沿的 VCS 解集; 同时, 不同场景下指标平均值的对比结果, 证明了 SAFEVCS 能够搜索到更有效暴露潜在 ADS 安全隐患的 VCS. 通过计算每个配置项的具体变化情况与累计变化, 揭示了引入关键配置组对优化搜索过程的有效性.

4.2.2 RQ2 结果

为回答 RQ2, 对每次运行的每 5 代种群计算其 HV, 用于评价搜索空间被生成种群覆盖的程度. 在定义的搜索问题时, 3 个搜索目标的参考点需要设置为理论能够达到的最大值, 即将 [*safetyDegree*, *maxDiff*, *totalDiff*] 均设置为 [12.0, 1.0, 12.0]. 具体原因为: *safetyDegree* 的最大值由所有运行数据统计后统计为 11.54, 设置为 12.0 能够限制搜索的 *safetyDegree* 指标均优于此数值; 搜索到的配置项的最差结果为完全变动, 即搜索到边界条件, 此时 *maxDiff* 为 1.0; 当所有的配置项均完全变动, 此时 *totalDiff* 为 12.0.

如图 4 所示, 与 SAFEVAR 相比, SAFEVCS 生成的 VCS 种群在初始阶段拥有更大的 *HV*. 然而, 中位数和四分位数范围较大, 体现出 SAFEVCS 的较大不确定性. 这些不确定性可能由不同的配置项变化概率导致. 随着代数的增加, SAFEVCS 相较于 SAFEVAR 体现更好的解集 VCS 的分布稳定性, 说明 SAFEVCS 能够稳定的搜索到有

效的 VCS, 能够实现更快的收敛. SAFEVAR 在搜索过程中生成了更多的离群点. 在晴天行人碰撞场景中, 这些离群点多为 *HV* 更大的样本; SAFEVCS 的离群点数量较少, 多数为 *HV* 更小的样本. 造成这种现象的原因可能为 SAFEVAR 在搜索过程中缺乏多样性的维护, 而 SAFEVCS 通过改进交叉与变异算子, 并引入去重策略, 能够保证搜索到帕累托前沿并充分扩展该前沿区域; 而针对 *HV* 小的样本, SAFEVCS 有效识别并抛弃适应度低的个体, 将搜索注意力集中在更接近帕累托前沿的搜索空间中.

表 6　不同累计变动值下配置平均变化率与配置平均变化对比

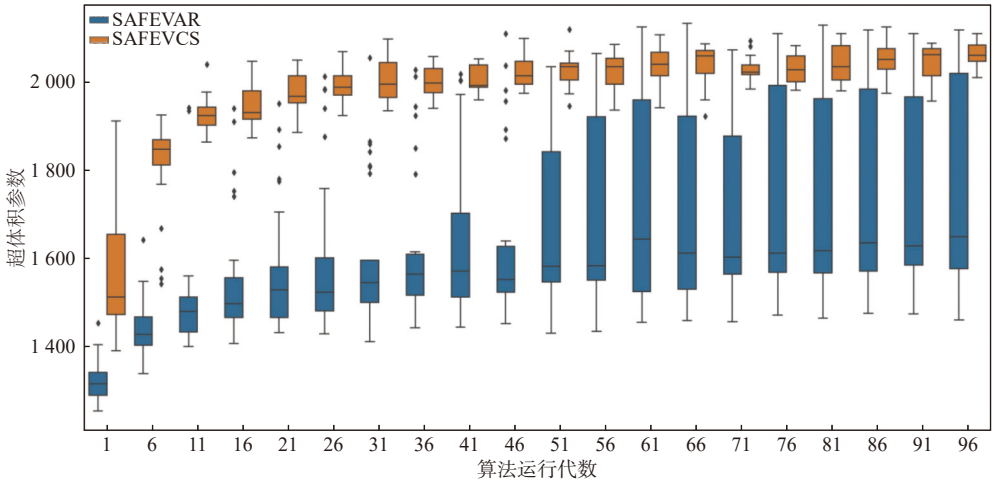
类别	计算单元	(0.5, 1.0]		(1.0, 1.5]		(1.5, 2.0]		(2.0, 2.5]		(2.5, 3.0]	
		PC (%)	Δ	PC (%)	Δ	PC (%)	Δ	PC (%)	Δ	PC (%)	Δ
配置项 C_i	max_rpm	1.44	−83.76	0.64	−37.39	3.60	−208.81	11.19	−649.12	15.83	−918.39
		2.82	−163.53	4.91	−284.63	8.73	−506.05	9.70	−562.34	3.97	−230.35
	dampRateFullT	**1.31**	**0.00**	**7.41**	**−0.01**	**10.14**	**−0.02**	8.98	−0.01	22.99	−0.03
		0.43	0.00	0.65	0.00	0.95	0.00	6.15	−0.01	13.27	−0.02
	dampRateZero_CE	**25.58**	**−0.51**	**40.28**	**−0.81**	**43.24**	**−0.86**	40.33	−0.81	41.82	−0.84
		6.23	−0.12	13.33	−0.27	28.94	−0.58	47.43	−0.95	49.44	−0.99
	dampRateZero_CD	**4.28**	**−0.01**	**9.28**	**−0.03**	**20.01**	−0.07	15.97	−0.06	21.63	−0.08
		0.57	0.00	0.54	0.00	1.82	−0.01	10.07	−0.04	28.75	−0.10
	gearSwitchTime	**2.16**	**−0.01**	**1.15**	**0.01**	0.59	0.00	12.56	−0.06	32.39	−0.16
		0.25	0.00	0.29	0.00	2.83	0.01	6.06	−0.03	24.35	−0.12
	clutchStrength	**1.31**	**0.13**	0.40	−0.04	**0.99**	**−0.10**	**1.95**	**−0.19**	**9.50**	**−0.95**
		0.37	0.04	0.76	0.08	0.22	0.02	0.48	−0.05	3.49	−0.35
	mass	9.27	222.97	9.22	221.67	6.25	150.14	1.81	43.48	0.11	2.65
		7.54	181.26	9.22	221.57	8.48	203.92	8.11	195.04	10.18	244.63
	dragCoeff	0.96	0.00	0.87	0.00	0.40	0.00	18.53	0.06	43.30	0.13
		0.85	0.00	0.39	0.00	0.19	0.00	1.52	0.00	12.55	0.04
	tireFric	**39.07**	**−1.37**	**50.14**	**−1.76**	**62.48**	**−2.19**	67.31	−2.36	70.28	−2.46
		3.26	−0.11	38.22	−1.34	55.77	−1.95	64.86	−2.27	66.51	−2.33
	dampRate	0.27	0.00	0.29	0.00	2.81	−0.01	3.33	−0.01	2.41	−0.01
		0.19	0.00	0.22	0.00	0.27	0.00	0.15	0.00	2.00	−0.01
	radius	7.08	−2.51	7.28	−2.59	7.00	−2.48	5.34	−1.90	4.17	−1.48
		7.39	−2.62	8.40	−2.98	8.62	−3.06	8.97	−3.19	8.17	−2.90
	maxBrakeTorque	11.64	−174.61	15.52	−232.82	14.26	−213.83	12.91	−193.65	12.93	−193.89
		13.21	−198.16	15.08	−226.17	16.84	−252.53	18.80	−282.06	18.02	−270.25
评估指标	*safetyDegree*	**−189.03**	**−70.01**	**−358.14**	**−132.64**	**−433.32**	**−160.49**	**−362.32**	**−134.19**	**−297.81**	**−110.30**
		−18.11	−6.71	−71.03	−26.31	−127.95	−47.39	−244.93	−90.71	−240.87	−89.21
	TIT	63.18	143.60	25.20	57.28	17.82	40.49	45.22	102.78	108.45	246.48
		5.46	12.41	30.12	68.45	38.95	88.52	41.14	93.51	58.75	133.53
	TET	44.20	36.83	−29.27	−24.39	−47.43	−39.52	−9.49	−7.91	62.50	52.08
		5.65	4.71	24.99	20.82	21.94	18.29	10.44	8.70	25.45	21.21
	aveDece	−149.34	−25.01	−188.02	−31.49	−217.83	−36.49	−165.74	−27.76	−128.80	−21.57
		−67.27	−11.27	−131.42	−22.01	−155.63	−26.07	−189.60	−31.76	−162.69	−27.25

注: 表内计算单元的第1行为SAFEVCS方法的结果, 第2行为SAEFVAR方法的结果

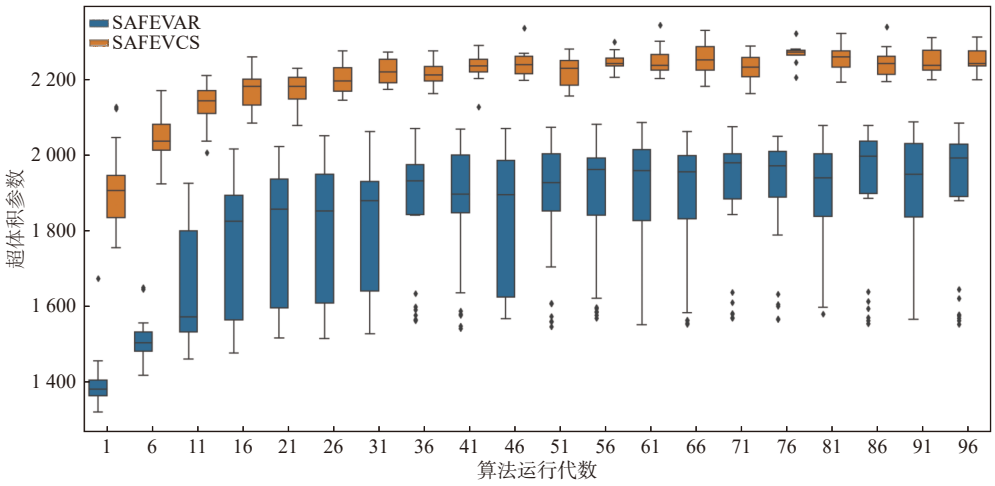
对比晴、雨天行人碰撞场景下的搜索过程, SAFEVCS 与 SAFEVAR 的 HV 在雨天场景下均大于同代数下晴天场景的种群的 HV, 且浮动更为明显. 该现象进一步佐证了: 随场景复杂度提升, VCS 暴露 ADS 安全隐患的可能性随之增加.

表 7 展示了两种方法的平均运行时间开销. 晴、雨天行人碰撞及晴天跟车碰撞场景, 运行单个场景的平均时长分别为 15.0 s、16 s 及 16 s. 晴、雨天场景下的运行时间差异, 主要由 CARLA 渲染及 AVUT 在雨天场景下驾驶表现更为谨慎, 而导致. 相同天气条件下, 跟车碰撞场景相比于行人碰撞场景, 整体交互过程更加复杂, 导致单个仿

真实例的平均运行时长略高. 相较于 SAFEVAR, SAFEVCS, 在 3 个场景中, 仿真效率分别提升了约 2.56 倍、2.66 倍、2.57 倍. 这是由于两种方法的运行时间主要为模拟器的仿真运行时间, 方法自身的运行时间对总体耗时的影响可忽略. 在仿真运行单个场景时长相对固定的情况下, 决定方法运行时间的主要因素为仿真运行次数. 引入恰当的个体选择策略及终止策略, 能够显著减少无效的运行次数, 提高搜索的整体效率.



(a) 晴天行人碰撞场景下的搜索过程 *HV* 对比



(b) 雨天行人碰撞场景下的搜索过程 *HV* 对比

图 4　不同场景下不同方法的 *HV* 对比

表 7　不同场景下 SAFEVCS 与 SAFEVAR 平均运行时间

场景	每轮运行时间 (s)	SAFEVAR (h)	SAFEVCS (h)	时间加快 (倍)
晴天行人碰撞场景	15.0	22.22	8.67	2.56
雨天行人碰撞场景	16.0	23.61	8.89	2.66
晴天跟车碰撞场景	16.0	23.19	9.02	2.57

综上, 通过对比搜索到的每一代种群的 *HV* 及其变化趋势, 证明了 SAFEVCS 能够搜索到更多有效的 VCS. 由于引入了恰当的终止策略与个体选择策略, SAFEVCS 对比 SAFEVAR 提升了 2.5 倍左右的运行效率, 验证了 SAFEVCS 的高效性.

4.2.3 RQ3 结果

为回答 RQ3, 实验中对比两种方法在不同搜索设置下的配置项累计变动百分比的排名, 对比不同配置项在同一场景下变动的一致性, 来佐证 VCS 对 ADS 安全性表现的影响. 通过统计 3 种测试场景下, 两种方法中 12 个配置项改变的频次, 得到表 8 及表 9 所示的配置项累计变动排序. 如表 8 中, 在 (0.5, 1.0] 区间中, 配置项 max_rpm 在累积变动百分比, 在 SAFEVCS 与 SAFEVAR 方法下的变动频次的排名分别为 8 和 6.

表 8 晴天行人碰撞场景下不同配置在 SAFEVCS 与 SAFEVAR 方法中累计变动排序

车辆配置	(0.5, 1.0]	(1.0, 1.5]	(1.5, 2.0]	(2.0, 2.5]	(2.5, 3.0]	平均排名差
max_rpm	8/6	10/6	8/4	7/5	7/10	1.8
dampRateFullT	9/9	6/8	5/9	8/8	5/6	−1.4
dampRateZero_CE	**2/4**	**2/3**	**2/2**	**2/2**	**3/2**	**−0.4**
dampRateZero_CD	6/8	4/9	3/8	4/4	6/3	−1.8
gearSwitchTime	7/11	8/11	11/7	6/9	4/4	−1.2
clutchStrength	10/10	11/7	10/11	11/11	9/11	**0.2**
mass	4/2	5/4	7/6	12/7	12/8	2.6
dragCoeff	11/7	9/10	12/12	3/10	2/7	−1.8
tireFric	**1/5**	**1/1**	**1/1**	**1/1**	**1/1**	**−0.8**
dampRate	12/12	12/12	9/10	10/12	11/12	**−0.8**
radius	5/3	7/5	6/5	9/6	10/9	1.8
maxBrakeTorque	3/1	3/2	4/3	5/3	8/5	1.8

表 9 雨天行人碰撞场景下不同配置在 SAFEVCS 与 SAFEVAR 方法中累计变动排序

车辆配置	(0.5, 1.0]	(1.0, 1.5]	(1.5, 2.0]	(2.0, 2.5]	(2.5, 3.0]	平均排名差
max_rpm	9/5	10/4	7/5	5/5	6/6	2.4
dampRateFullT	12/11	8/11	12/11	9/9	11/8	**0.4**
dampRateZero_CE	**2/4**	**1/1**	**2/1**	**2/2**	**2/2**	**−0.2**
dampRateZero_CD	6/6	3/8	4/8	3/7	4/4	−2.6
gearSwitchTime	11/8	12/6	8/7	10/8	7/10	1.8
clutchStrength	5/10	9/9	9/9	12/11	9/12	−1.4
mass	3/2	4/3	6/4	8/4	10/5	2.6
dragCoeff	4/7	11/10	10/12	7/12	3/9	−3
tireFric	10/12	5/7	**1/2**	**1/1**	**1/1**	−1
dampRate	8/9	6/12	11/10	11/10	12/11	**−0.8**
radius	7/3	7/5	5/6	6/6	8/7	1.2
maxBrakeTorque	**1/1**	**2/2**	**3/3**	**4/3**	**5/3**	**0.6**

如表 8 所示, 在晴天行人碰撞场景中, SAFEVCS 与 SAFEVAR 在各配置项的平均排名上差别不大; 最大排名差值为 5, 即 dampRateZero_CD 在累计变动百分比为 (1.0, 1.5]、(1.5, 2.0] 时取得, 分别为 4/9 及 5/8, 出现该现象的原因是 SAFEVCS 引入了相关配置组的关联变异, 提高了该配置项的变动排名. 同时, 在两种方法中, 所有的配置项在变化的频次排名上结果相似. 其中, 多数配置项的平均排名差控制在较小的范围内, 全部 12 个配置项的平均排名差均在 2.0 之内. 该结果表明: 在相同的测试场景下, 不同的搜索方法搜索到的 VCS 具有一致性. 部分配置项的不一致性是源于采用搜索算法自身的随机性及 SAFEVCS 中采用的算子策略.

如表 9 所示, 雨天行人碰撞场景下, SAFEVCS 与 SAFEVAR 显现出了对不同配置项的偏好, 两个关联组{dampRateFullT, dampRateZeroT_CE, dampRate_zeroT_CD, dampRate}和{mass, tireFric, radius}中的配置项累计变动排名均有浮动. 然而, 两个方法依旧在某些配置项上, 显现出了较高的一致性, 如在 dampRateZeroT_CE、maxBrakeTorque 配置项上, 其差异排名均控制在 2.0 以内, 这证明了配置项对 ADS 安全性的影响具有相同趋势, 与具体搜索方法无关.

综上, 对比 SAFEVCS 与 SAFEVAR 中不同配置项的累计变动百分比, 发现了在相同测试场景下, 同一个变化区间内, 产生了相似的配置项变动累计值排名; 而同一种方法, 在不同的测试场景下的配置项变动累计值排名, 也

针对场景特性产生了变化, 证明了 VCS 对 ADS 安全性表现的普遍影响.

4.2.4 消融实验

为进一步验证 SAFEVCS 方法中改进算子的有效性, 进行了以下消融实验. 在雨天行人碰撞测试场景下, 对 4 种搜索方法分别重复执行了 5 次实验, 以减少随机性带来的影响; 并对每一种搜索方法的最后一代种群进行评估. 评估使用 HV、IGD、碰撞率指标进行衡量. 碰撞率为给定场景下车辆的碰撞可能性, 数据越高代表搜索到的 VCS 能够有效地暴露 ADS 的潜在碰撞风险.

实验结果如表 10 所示, 对比的 4 种搜索方法依次为 SAFEVAR、SAFEVCS、在 SAFEVCS 中去除交叉改进算子的 SAFEVCS-crossover 以及在 SAFEVCS 中去除变异改进算子的 SAFEVCS-mutation. 实验结果显示, 相较于 SAEFVAR, SAFEVCS 在各项指标上均有更好地表现. 同时, 改进的交叉算子极大提高了搜索结果与帕累托前沿的接近程度, 说明生成的最终种群效果更优. 在去除掉改进的变异算子之后, 由于缺少了随机性扰动, 搜索陷入了局部最优, 导致 *HV* 和碰撞率有了提高. 然而改进的变异算子在 *IGD* 指标上仍然保持优势, 说明引入更多特异化的变异操作能够提高最终种群的多样性, 使解集更紧密逼近帕累托前沿. 通过实验结果能够证明本文提出的 SAFEVCS 方法在搜索车辆配置上有更好的效果.

表 10 消融实验结果

方法	*HV*	*IGD* (均值±标准差)	碰撞率 (%)
SAFEVAR	149.908	1.194±0.231	10.66
SAFEVCS	165.325	**0.641±0.033**	47.92
SAFEVCS-crossover	164.248	0.806±0.044	47.50
SAFEVCS-mutation	**166.513**	0.674±0.040	**51.00**

5 总 结

本文提出了一种结合模糊测试的安全攸关车辆配置搜索方法 SAFEVCS. 相较于 SAFEVAR 方法, SAFEVCS 通过引入模糊测试技术, 改进搜索算法交叉与变异算子的多样性条件限定及约束, 提高了输出 VCS 结果的多样性. 为提高搜索效率, SAFEVCS 实现了自适应的搜索终止策略和去重策略. 通过分析 SAFEVCS 及 SAFEVAR 的 VCS 输出结果, 说明了不同的 VCS 配置项组合能够暴露潜在 ADS 安全隐患. 在未来研究中, 为进一步提高 SAFEVCS 在多场景下暴露潜在 ADS 安全隐患的 VCS 搜索效率, 可以借助大模型的学习与泛化能力生成场景描述信息, 自动构建关键场景, 增加对不同场景间, 如交叉路口转弯、夜间行车等, 及针对不同待测 ADS 间的 VCS 搜索. 针对模拟器与真实世界驾驶环境的差异, 未来将考虑使用硬件在环的方式实现不同 VCS 下 ADS 的高保真测试.

References

[1] Koopman P, Wagner M. Challenges in autonomous vehicle testing and validation. SAE Int'l Journal of Transportation Safety, 2016, 4(1): 15–24. [doi: 10.4271/2016-01-0128]

[2] Jiang ZM, Dang SB, Li HY, Pan Y. A survey on the research progress of scenario-based testing for autonomous vehicles. Automobile Technology, 2022(8): 10–22 (in Chinese with English abstract). [doi: 10.19620/j.cnki.1000-3703.20211088]

[3] Li GP, Li YR, Jha S, Tsai T, Sullivan M, Hari SKS, Kalbarczyk Z, Iyer R. AV-FUZZER: Finding safety violations in autonomous driving systems. In: Proc. of the 31st IEEEInt'l Symp. on Software Reliability Engineering (ISSRE). Coimbra: IEEE, 2020. 25–36. [doi: 10.1109/ISSRE5003.2020.00012]

[4] Kim S, Liu M, Rhee JJ, Jeon Y, Kwon Y, Kim CH. DriveFuzz: Discovering autonomous driving bugs through driving quality-guided fuzzing. In: Proc. of the 2022 ACM SIGSAC Conf. on Computer and Communications Security. Los Angeles: ACM, 2022. 1753–1767. [doi: 10.1145/3548606.3560558]

[5] Queiroz R, Sharma D, Caldas R, Czarnecki K, García S, Berger T, Pelliccione P. A driver-vehicle model for ADS scenario-based testing. IEEE Trans. on Intelligent Transportation Systems, 2024, 25(8): 8641–8654. [doi: 10.1109/TITS.2024.3373531]

[6] Li WL, Li C, Zhang YN, Song Y, Hu X. Game neural network algorithm for generating autonomous driving test scenarios. Computer Engineering and Applications, 2024, 60(22): 335–346 (in Chinese with English abstract). [doi: 10.3778/j.issn.1002-8331.2307-0320]

[7] Feng S, Sun HW, Yan XT, Zhu HJ, Zou ZX, Shen SY, Liu HX. Dense reinforcement learning for safety validation of autonomous vehicles. Nature, 2023, 615(7953): 620–627. [doi: 10.1038/S41586-023-05732-2]

[8] Yin KO, Arcaini P, Yue T, Ali S. Analyzing the impact of product configuration variations on advanced driver assistance systems with search. In: Proc. of the 2021 Genetic and Evolutionary Computation Conf. Lille: ACM, 2021. 1106–1114. [doi: 10.1145/3449639.3459332]

[9] Pan Q, Wang TX, Ma JW, Arcaini P, Yue T. Simulation-based safety assessment of vehicle characteristics variations in autonomous driving systems. ACM Trans. on Software Engineering and Methodology, 2025. [doi: 10.1145/3743673]

[10] Gillespie TD. Fundamentals of Vehicle Dynamics. Warrendale: Society of Automotive Engineers, 1992.

[11] Tang SC, Zhang ZY, Zhang Y, Zhou JX, Guo Y, Liu S, Guo SJ, Li YF, Ma L, Xue YX, Liu Y. A survey on automated driving system testing: Landscapes and trends. ACM Trans. on Software Engineering and Methodology, 2023, 32(5): 124. [doi: 10.1145/3579642]

[12] Ben Abdessalem R, Nejati S, Briand LC, Stifter T. Testing advanced driver assistance systems using multi-objective search and neural networks. In: Proc. of the 31st IEEE/ACM Int'l Conf. on Automated Software Engineering. Singapore: IEEE, 2016. 63–74. [doi: 10.1145/2970276.2970311]

[13] Ben Abdessalem R, Panichella A, Nejati S, Briand LC, Stifter T. Testing autonomous cars for feature interaction failures using many-objective search. In: Proc. of the 33rd IEEE/ACM Int'l Conf. on Automated Software Engineering. Montpellier: IEEE, 2018. 143–154. [doi: 10.1145/3238147.3238192]

[14] Luo YX, Zhang XY, Arcaini P, Jin Z, Zhao HY, Ishikawa F, Wu RX, Xie T. Targeting requirements violations of autonomous driving systems by dynamic evolutionary search. In: Proc. of the 36th IEEE/ACM Int'l Conf. on Automated Software Engineering. Melbourne: IEEE, 2021. 279–291. [doi: 10.1109/ASE51524.2021.9678883]

[15] Birchler C, Ganz N, Khatiri S, Gambi A, Panichella S. Cost-effective simulation-based test selection in self-driving cars software with SDC-Scissor. In: Proc. of the 29th IEEE Int'l Conf. on Software Analysis, Evolution and Reengineering. Honolulu: IEEE, 2022. 164–168. [doi: 10.1109/SANER53432.2022.00030]

[16] Zong HX, Hou ZL, Liu H. Safety-violation scenarios search for ADS via multi-objective genetic algorithm. In: Proc. of the 2024 IEEE Smart World Congress (SWC). Nadi: IEEE, 2024. 1961–1966. [doi: 10.1109/SWC62898.2024.00301]

[17] Tian HX, Jiang Y, Wu GQ, Yan JR, Wei J, Chen W, Li S, Ye D. MOSAT: Finding safety violations of autonomous driving systems using multi-objective genetic algorithm. In: Proc. of the 30th ACM Joint European Software Engineering Conf. and Symp. on the Foundations of Software Engineering. Singapore: ACM, 2022. 94–106. [doi: 10.1145/3540250.3549100]

[18] Tian HX, Wu GQ, Yan JR, Jiang Y, Wei J, Chen W, Li S, Ye D. Generating critical test scenarios for autonomous driving systems via influential behavior patterns. In: Proc. of the 37th IEEE/ACM Int'l Conf. on Automated Software Engineering. Rochester: ACM, 2022. 46. [doi: 10.1145/3551349.3560430]

[19] Zohdinasab T, Riccio V, Tonella P. Focused test generation for autonomous driving systems. ACM Trans. on Software Engineering and Methodology, 2024, 33(6): 152. [doi: 10.1145/3664605]

[20] Humeniuk D, Khomh F, Antoniol G. A search-based framework for automatic generation of testing environments for cyber-physical systems. Information and Software Technology, 2022, 149: 106936. [doi: 10.1016/j.infsof.2022.106936]

[21] Li DC, Wong WE, Liu H, Zhao M. Many-objective search-based coverage-guided automatic test generation for deep neural networks. arXiv:2411.01033, 2024.

[22] Diehl C, Sievernich TS, Krüger M, Hoffmann F, Bertram T. Uncertainty-aware model-based offline reinforcement learning for automated driving. IEEE Robotics and Automation Letters, 2023, 8(2): 1167–1174. [doi: 10.1109/LRA.2023.3236579]

[23] Lu CJ. Test scenario generation for autonomous driving systems with reinforcement learning. In: Proc. of the 45th IEEE/ACM Int'l Conf. on Software Engineering: Companion Proc. Melbourne: IEEE, 2023. 317–319. [doi: 10.1109/ICSE-Companion58688.2023.00086]

[24] Zhang PC, Ren B, Dong H, Dai QY. CAGFuzz: Coverage-guided adversarial generative fuzzing testing for image-based deep learning systems. IEEE Trans. on Software Engineering, 2022, 48(11): 4630–4646. [doi: 10.1109/TSE.2021.3124006]

[25] Xie DN, Li YT, Kim M, Pham HV, Tan L, Zhang XY, Godfrey MW. DocTer: Documentation-guided fuzzing for testing deep learning API functions. In: Proc. of the 31st ACM SIGSOFT Int'l Symp. on Software Testing and Analysis. ACM, 2022. 176–188. [doi: 10.1145/3533767.3534220]

[26] Wang SQ, Zhao JX, Liu C, Wu W, Liu Z. Fuzz testing method of binary code based on deep reinforcement learning. Computer Science, 2024, 51(S1): 230800078 (in Chinese with English abstract). [doi: 10.11896/jsjkx.230800078]

[27] Sun Y, Poskitt CM, Sun J, Chen YQ, Yang ZJ. LawBreaker: An approach for specifying traffic laws and fuzzing autonomous vehicles. In: Proc. of the 37th IEEE/ACM Int'l Conf. on Automated Software Engineering. Rochester: ACM, 2022. 62. [doi: 10.1145/3551349.3556897]

[28] Zhong ZY, Kaiser G, Ray B. Neural network guided evolutionary fuzzing for finding traffic violations of autonomous vehicles. IEEE Trans. on Software Engineering, 2023, 49(4): 1860–1875. [doi: 10.1109/TSE.2022.3195640]

[29] Moukahal LJ, Zulkernine M, Soukup M. Vulnerability-oriented fuzz testing for connected autonomous vehicle systems. IEEE Trans. on Reliability, 2021, 70(4): 1422–1437. [doi: 10.1109/TR.2021.3112538]

[30] Moukahal LJ, Zulkernine M, Soukup M. Boosting grey-box fuzzing for connected autonomous vehicle systems. In: Proc. of the 21st IEEE Int'l Conf. on Software Quality, Reliability and Security Companion. IEEE, 2021. 516–527. [doi: 10.1109/QRS-C55045.2021.00080]

[31] Deb K, Agrawal RB. Simulated binary crossover for continuous search space. Complex Systems, 1995, 9(2): 115–148.

[32] Minderhoud MM, Bovy PHL. Extended time-to-collision measures for road traffic safety assessment. Accident Analysis & Prevention, 2001, 33(1): 89–97. [doi: 10.1016/S0001-4575(00)00019-1]

[33] Manès VJM, Han H, Han C, Cha SK, Egele M, Schwartz EJ, Woo M. The art, science, and engineering of fuzzing: A survey. IEEE Trans. on Software Engineering, 2021, 47(11): 2312–2331. [doi: 10.1109/TSE.2019.2946563]

[34] Alhijawi B, Awajan A. Genetic algorithms: Theory, genetic operators, solutions, and applications. Evolutionary Intelligence, 2024, 17(3): 1245–1256. [doi: 10.1007/s12065-023-00822-6]

[35] Ulbrich S, Menzel T, Reschka A, Schuldt F, Maurer M. Defining and substantiating the terms scene, situation, and scenario for automated driving. In: Proc. of the 18th IEEE Int'l Conf. on Intelligent Transportation Systems. Gran Canaria: IEEE, 2015. 982–988. [doi: 10.1109/ITSC.2015.164]

[36] Dosovitskiy A, Ros G, Codevilla F, López AM, Koltun V. CARLA: An open urban driving simulator. In: Proc. of the 1st Annual Conf. on Robot Learning. Mountain View: PMLR, 2017. 1–16.

[37] Chen D, Koltun V, Krähenbühl P. Learning to drive from a world on rails. In: Proc. of the 2021 IEEE/CVF Int'l Conf. on Computer Vision. Montreal: IEEE, 2021. 15570–15579. [doi: 10.1109/ICCV48922.2021.01530]

[38] Prakash A, Chitta K, Geiger A. Multi-modal fusion Transformer for end-to-end autonomous driving. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 7077–7087. [doi: 10.1109/CVPR46437.2021.00700]

[39] Codevilla F, Santana E, López AM, Gaidon A. Exploring the limitations of behavior cloning for autonomous driving. In: Proc. of the 2019 IEEE/CVF Int'l Conf. on Computer Vision. Seoul: IEEE, 2019. 9329–9338. [doi: 10.1109/ICCV.2019.00942]

[40] Zhang XY, Arcaini P, Ishikawa F. An incremental approach for understanding collision avoidance of an industrial path planner. IEEE Trans. on Dependable and Secure Computing, 2023, 20(4): 2713–2730. [doi: 10.1109/TDSC.2022.3159773]

[41] Zhou Y, Sun Y, Tang Y, Chen YQ, Sun J, Poskitt CM, Liu Y, Yang ZJ. Specification-based autonomous driving system testing. IEEE Trans. on Software Engineering, 2023, 49(6): 3391–3410. [doi: 10.1109/TSE.2023.3254142]

[42] Zhang MH, Du DH, Zhang MZ, Zhang L, Wang Y, Zhou WT. Spatio-temporal trajectory data-driven autonomous driving scenario meta-modeling approach. Ruan Jian Xue Bao/Journal of Software, 2021, 32(4): 973–987 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6226.htm [doi: 10.13328/j.cnki.jos.006226]

[43] Chen JQ, Shu XX, Lan FC, Wang JF. Construction of autonomous vehicles test scenarios with typical dangerous accident characteristics. Journal of South China University of Technology (Natural Science Edition), 2021, 49(5): 1–8 (in Chinese with English abstract). [doi: 10.12141/j.issn.1000-565X.200371]

[44] Xia CY, Huang S, Yao YM, Zheng CY, Wang YT. Generating autonomous driving safety violation scenarios based on multi-objective optimization. In: Proc. of the 23rd IEEE Int'l Conf. on Software Quality, Reliability, and Security Companion (QRS-C). Chiang Mai: IEEE, 2023. 509–515. [doi: 10.1109/QRS-C60940.2023.00076]

[45] Mahmud SMS, Ferreira L, Hoque MS, Tavassoli A. Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs. IATSS Research, 2017, 41(4): 153–163. [doi: 10.1016/j.iatssr.2017.02.001]

[46] van der Horst ARA. A time-based analysis of road user behaviour in normal and critical encounters [Ph. D. Thesis]. Soesterberg: Institute for Perception TNO, 1990. 517.

[47] NVIDIA Corporation. NVIDIA PhysX SDK 3.4. 0 documentation. 2017. https://docs.nvidia.com/gameworks/content/gameworkslibrary/physx/guide/Manual/Vehicles.html

[48] Blank J, Deb K. Pymoo: Multi-objective optimization in Python. IEEE Access, 2020, 8: 89497–89509. [doi: 10.1109/ACCESS.2020.2990567]

[49] Arcuri A, Briand L. A practical guide for using statistical tests to assess randomized algorithms in software engineering. In: Proc. of the 33rd Int'l Conf. on Software Engineering. Honolulu: IEEE, 2011. 1–10. [doi: 10.1145/1985793.1985795]

[50] Ali S, Arcaini P, Pradhan D, Safdar SA, Yue T. Quality indicators in search-based software engineering: An empirical evaluation. ACM Trans. on Software Engineering and Methodology, 2020, 29(2): 10. [doi: 10.1145/3375636]

[51] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Trans. on Evolutionary Computation, 1999, 3(4): 257–271. [doi: 10.1109/4235.797969]

附中文参考文献

[2] 蒋拯民, 党少博, 李慧云, 潘毅. 自动驾驶汽车场景测试研究进展综述. 汽车技术, 2022(8): 10–22. [doi: 10.19620/j.cnki.1000-3703.20211088]

[6] 李文礼, 李超, 张祎楠, 宋越, 胡雄. 面向自动驾驶测试场景生成的博弈神经网络算法. 计算机工程与应用, 2024, 60(22): 335–346. [doi: 10.3778/j.issn.1002-8331.2307-0320]

[26] 王栓奇, 赵健鑫, 刘驰, 武伟, 刘钊. 基于深度强化学习的二进制代码模糊测试方法. 计算机科学, 2024, 51(S1): 230800078. [doi: 10.11896/jsjkx.230800078]

[42] 张梦寒, 杜德慧, 张铭茁, 张雷, 王耀, 周文韬. 时空轨迹数据驱动的自动驾驶场景元建模方法. 软件学报, 2021, 32(4): 973–987. http://www.jos.org.cn/1000-9825/6226.htm [doi: 10.13328/j.cnki.jos.006226]

[43] 陈吉清, 舒孝雄, 兰凤崇, 王俊峰. 典型危险事故特征的自动驾驶测试场景构建. 华南理工大学学报 (自然科学版), 2021, 49(5): 1–8. [doi: 10.12141/j.issn.1000-565X.200371]

作者简介

王铁鑫, 博士, 副教授, CCF 高级会员, 主要研究领域为数字孪生, 自动驾驶仿真测试, 知识表征与融合, 时序数据预测.

马健伟, 硕士, 主要研究领域为自动驾驶仿真测试, 自然语言处理.

林聪, 硕士生, 主要研究领域为自动驾驶仿真测试, 多目标搜索.

杨科, 硕士生, CCF 学生会员, 主要研究领域为自动驾驶仿真测试.

王飞, 博士, 讲师, CCF 专业会员, 主要研究领域为智能化软件工程, 软件安全性, 形式化方法.