



AdvSce: 面向自动驾驶系统的安全关键场景生成工具

李昌文^{1,2}, 晏荣杰^{1,2*}, 张健^{1,2}

1. 计算机科学国家重点实验室 (中国科学院软件研究所), 北京 100190

2. 中国科学院大学, 北京 100049

* 通信作者. E-mail: yrj@ios.ac.cn

收稿日期: 2022-08-11; 修回日期: 2022-11-13; 接受日期: 2023-01-30; 网络出版日期: 2023-04-06

国家自然科学基金 (批准号: 62132020) 资助项目

摘要 AdvSce 是一个面向自动驾驶系统的安全关键场景生成工具, 能够在给定初始场景的基础上, 以添加额外交通参与者的方式, 生成挑战自动驾驶系统控制车辆 (主车) 安全性的扰动场景, 用于自动驾驶系统的仿真测试. 不同于既往研究中, 首先确定场景参数空间, 再使用黑盒优化算法生成安全关键场景的工具或方法, AdvSce 可以基于对主车行为的分析, 在场景生成过程中动态地构造能够挑战主车安全性的场景参数空间, 从而实现高度自动化且高效的安全关键场景生成. 我们使用 AdvSce 在 LGSVL 仿真器中生成了面向 Apollo 7.0 自动驾驶系统的测试场景. 结果表明, 相比现有的安全关键场景生成工具, AdvSce 具有更高的安全问题场景生成效率.

关键词 自动驾驶系统, 仿真测试, 安全关键场景生成, 基于搜索的优化, 行为序列提取

1 引言

安全性是推广与普及自动驾驶技术的关键. 基于场景仿真的方法具有低成本、无安全风险等优势, 已成为测试与评估自动驾驶系统安全性的重要手段. 安全关键场景 (safety-critical scenario) 是指很可能引发主车安全问题的场景, 对测试自动驾驶系统安全性具有重要价值, 但在常规的驾驶环境中非常少见^[1]. 如何自动且高效地生成安全关键场景, 是自动驾驶仿真测试领域的重要问题^[2].

安全关键场景生成问题可以转化为优化问题, 使用优化算法, 在一定的场景参数空间内多次采样, 逐步减小与主车安全性相关的度量指标, 最终获得危险场景或碰撞场景^[3~12]. 然而, 在实施此类方法时, 仍然面临以下问题.

- 场景参数空间确定. 为了实现优化问题的建模, 相关研究基于不同的场景模型设计了相应的场景参数空间. 例如, Viswanadha 等^[9] 使用场景描述语言 SCENIC^[13] 编写了 10 组具有不同道路结构、

引用格式: 李昌文, 晏荣杰, 张健. AdvSce: 面向自动驾驶系统的安全关键场景生成工具. 中国科学: 信息科学, 2023, 53: 815–820, doi: 10.1360/SSI-2022-0317
Li C W, Yan R J, Zhang J. AdvSce: safety critical scenario generation for testing autonomous driving systems (in Chinese). Sci Sin Inform, 2023, 53: 815–820, doi: 10.1360/SSI-2022-0317

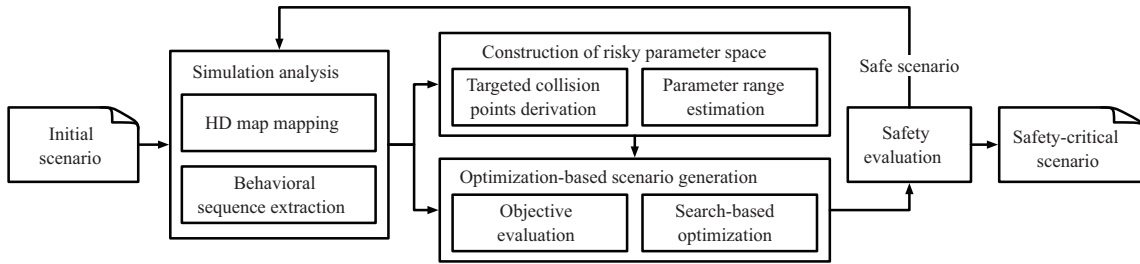


图 1 AdvSce 安全关键场景生成框架

Figure 1 Framework of AdvSce for safety-critical scenario generation

交通参与者行为的抽象场景; AV-FUZZER^[10] 对交通参与者的原子动作进行建模, 并将场景定义为周边参与者原子动作的序列; MOSAT^[11] 基于事故数据库定义了挑战主车安全性的周边参与者复合动作, 将复合动作与原子动作的组合方式作为参数; AdvSim^[12] 基于运动学模型, 在一定范围内扰动给定初始场景中参与者的轨迹。然而, 这些方式通常缺少对主车行为的分析, 无法准确预估其安全风险, 导致确定的参数范围较大, 包含大量安全场景, 使后续优化效率降低。

- 优化方法. 由于自动驾驶系统行为的复杂性, 生成安全关键场景时, 常使用黑盒优化算法^[3~12]。这类算法不分析场景的仿真过程, 仅根据主车的安全评价指标引导优化过程, 因此存在效率低、终止条件不明确等弊端, 难以保证在一定的仿真次数内发现问题场景, 或报告无法发现问题场景。

为此, 我们提出一种新的安全关键场景生成方法, 并实现了相应工具 AdvSce。该方法及工具通过引入新的交通参与者, 对给定场景中主车的行为进行干扰, 生成安全关键场景。AdvSce 的场景生成框架如图 1 所示, 其具有以下特点: (1) 根据地图及初始场景仿真过程中采样的主车位置, 提取主车在地图上的轨迹与行为; (2) 基于对主车行为的分析, 自动构造能够挑战其安全性的场景参数空间; (3) 采用基于搜索的优化方法, 在场景参数空间内生成安全关键场景; (4) 根据主车在不同场景中行为的变化, 构造新的场景参数空间, 实现对主车做出的新行为继续施加干扰, 同时避免对相同行为重复干扰, 进一步提高安全关键场景生成的有效性。

AdvSce 可作为独立的场景增强工具, 扰动已有场景, 完成对自动驾驶系统的测试。工具也提供了丰富的 Python API, 用于创建初始场景、访问地图数据、配置场景生成参数、分析场景仿真结果。用户可调用 AdvSce 的 API, 扩展现有工具的能力; 或基于当前场景生成框架, 做进一步扩展。

2 AdvSce 安全关键场景生成方法

仿真过程分析. 给定初始场景后, AdvSce 首先将仿真过程中采样的主车位置映射至高精地图, 获得其所在车道以及相对车道中心线的横向、纵向偏移; 其次, 结合高精地图提供的车道连通关系, 根据主车的运动轨迹特征, 提取其行为序列。主车的行为序列是对其运动轨迹的抽象, 序列中的每个元素对应轨迹中的一个片段, 表示主车在这段轨迹中的行为类型, 例如车道追踪、右侧变道、左转等。

风险参数空间构造. 基于主车在初始场景中的行为序列, AdvSce 通过两个步骤构造引入交通参与者的初始位置范围, 使得交通参与者引入其中后, 主车必须采取行动, 避免与引入参与者的碰撞。

- 提取目标碰撞点. 目标碰撞点位于被测车在初始场景中的轨迹上, 表示预期主车将与引入交通参与者发生碰撞的位置, 是确定该交通参与者初始位置范围的参考。AdvSce 为每种行为类型预设了

与之对应的目标碰撞点选取规则. 对于不改变车道的行为, 按一定时间间隔将采样位置作为目标碰撞点; 对于变道、借道等改变车道的行为, 目标碰撞点选择主车切换车道后的第一个采样点; 而跨越多个车道的行为如左转、右转等, 选取主车轨迹与其他车道相交位置.

- 估计初始位置范围. 引入交通参与者的控制策略从一组预设集合中枚举. 选定引入参与者的控制策略后, 对于每个目标碰撞点, 根据引入参与者的控制方法和参数, 可以预测其初始位置, 使得该参与者能够与初始场景中的主车在同一时刻到达目标碰撞点位置. 由于估计时存在一定误差, 且主车与引入交通参与者在目标碰撞点附近交互关系的微小变化, 都可能成为主车是否发生安全问题的决定性因素. 因此, AdvSce 将预测的初始位置松弛为一个连续区间. 多个目标碰撞点对应的不同区间, 组成了引入参与者的初始位置范围.

安全关键场景生成. AdvSce 使用基于搜索的优化方法, 生成主车安全度量指标最小的场景. 引入参与者初始位置范围中的每个连续区间, 确定了优化问题的一个解空间, 生成过程还需考虑优化目标与优化方法.

- 优化目标. AdvSce 将主车与引入交通参与者的相对距离作为主车的安全性评价指标. 该距离仅在主车与引入交通参与者均位于目标碰撞点所在车道时有效, 从而实现对主车在目标碰撞点附近的安全性度量. AdvSce 的优化目标为最小化该相对距离.

- 优化方法. AdvSce 使用二分搜索的方法, 在一个区间内调整引入交通参与者的初始位置. 搜索的每次迭代选取当前区间的中点作为引入交通参与者的初始位置生成场景, 并根据场景仿真中主车与引入交通参与者到达目标碰撞点附近的先后关系, 调整下次迭代的搜索区间. 若引入交通参与者在主车之前到达目标碰撞点, 则将初始位置缩小至距离目标碰撞点更远的区间; 否则, 将初始位置缩小至距离目标碰撞点更近的区间. 迭代持续进行, 直到主车与周边交通参与者发生碰撞, 或待搜索区间的长度小于一定阈值. 该方法不仅具有较高的搜索效率, 还具有明确的终止条件, 即使一个区间内不存在引发主车碰撞的场景参数, 也能够一定次数内终止搜索.

场景生成调度. 如上文所述, AdvSce 首先根据主车在扰动前场景中的行为, 提取目标碰撞点; 其次对每个目标碰撞点, 确定与之对应的优化问题解空间; 最终通过求解该优化问题生成安全关键场景. 在此过程中, 一个初始场景可能存在多个目标碰撞点, 从而对应多个具有不同解空间的优化问题; 求解一个优化问题可能生成多个场景, 其中未发生碰撞的场景可作为初始场景, 继续引入交通参与者进行扰动. 为此, AdvSce 提供一组调度策略, 管理优化任务的求解顺序与初始场景的选取.

AdvSce 通过维护存储目标碰撞点的优先级队列实现上述调度, 目标碰撞点的优先级由其对应的行为类型决定. 通过设置不同行为类型的优先级, 可优先测试主车在执行特定行为时的安全性.

从给定的初始场景中提取目标碰撞点, 并将其加入队列后, AdvSce 迭代地执行调度. 每次迭代从队列头部取出一个目标碰撞点, 并求解与之对应的优化问题. 对于求解过程中生成的无碰撞场景, 若主车在其中的行为序列未出现在已生成场景中, 则 AdvSce 将以该场景为新的初始场景, 提取其中的目标碰撞点加入优先级队列, 等待继续引入交通参与者. 因此, AdvSce 能够在生成场景的迭代过程中, 根据主车行为序列及其变化, 有针对性地向场景中逐个添加交通参与者, 对主车做出的新行为施加干扰, 同时避免干扰主车的重复行为, 实现安全问题场景生成的高效性.

表 1 AdvSce 与其他安全关键场景生成方法的性能对比

Table 1 Performance comparison between AdvSce and other safety-critical scenario generation methods

Generation method	Total number of generated scenarios	Number and ratio of generated unsafe scenarios			Time to generate one unsafe scenario (s)
		Total	Collision	Dangerous	
VERIFAI (BO)	350	64 (18.2%)	20 (5.7%)	44 (12.6%)	509.09
VERIFAI (CE)	350	72 (20.6%)	27 (7.7%)	45 (12.9%)	756.19
VERIFAI (HA)	350	96 (27.4%)	32 (9.1%)	64 (18.3%)	368.81
AV-FUZZER	329	30 (9.1%)	6 (1.8%)	24 (7.3%)	1272.90
AdvSce	156	64 (41.0%)	46 (29.4%)	18 (11.5%)	175.62

3 实现与评估

AdvSce¹⁾ 使用 Python 语言实现, 能够方便地与其他自动驾驶仿真、测试工具交互. 其内部包含独立于其他工具的场景、地图数据结构. 通过开发相关接口, 用户可加载多种格式的高精地图, 并实现不同自动驾驶系统在不同仿真工具中的测试. 目前, 工具支持加载 Apollo HD Map 格式的高精地图, 并提供与 Apollo 自动驾驶系统、LGSVL 仿真器^[14] 对接的接口实现.

我们将 Apollo 7.0²⁾ 作为被测系统, 以 LGSVL 仿真器内置的 San Francisco 地图作为运行操作域, 分别使用安全关键场景生成工具 VERIFAI^[15], AV-FUZZER^[10], 与 AdvSce 在 LGSVL 中生成场景. 其中, VERIFAI 提供了 3 种采样方法: 贝叶斯优化 (Bayesian optimization, BO)、交叉熵采样 (cross entropy, CE)、霍尔顿采样 (halton, HA).

我们随机生成了 35 个场景作为初始场景, 并预设了两种控制策略下的车辆作为引入的交通参与者: 静止不动与车道追踪, 分别模拟故障车辆与正常行驶车辆. 车道追踪策略包含防碰撞功能, 并遵守交通规则. 对于每个初始场景, 设置最多生成 10 个扰动后场景. 由于使用 VERIFAI 与 AV-FUZZER 时必须指定引入参与者初始位置范围, 我们参考系统性场景测试研究中的实验方案^[8], 将引入参与者的初始位置范围设置为主车经过区域中任意车道的中心线上, 并与周边交通参与者保持安全距离.

表 1 对比了 VERIFAI, AV-FUZZER 与 AdvSce 生成的场景中, 主车存在安全问题的场景数量及其占比, 以及生成单个安全问题场景的平均用时. 安全问题包括碰撞与危险两种, 其中危险场景指主车与周边交通参与者的最短距离小于 0.5 米, 但未发生碰撞的场景. 结果表明, 相比 VERIFAI 与 AV-FUZZER, 使用 AdvSce 能够大幅提高安全问题场景的生成率及时间效率. 在最多生成 350 个场景的限制下, AdvSce 仅生成了 156 个场景即完成了全部的优化任务, 并发现了更多数量的碰撞场景.

4 总结

本文介绍了一种面向自动驾驶系统的安全关键场景生成方法与相应工具 AdvSce. 该方法能够基于对主车行为的分析, 自动构造挑战主车安全性的场景参数空间, 并使用基于搜索的优化方法生成安全关键场景, 相比现有技术方案达到了更高的安全关键场景生成效率. AdvSce 具有良好的可扩展性与适应性, 未来可进一步与覆盖率驱动的场景生成工具结合, 同时实现安全关键场景生成的多样性与高效性.

1) <https://github.com/LIIHWF/AdvSce/>.

2) <https://github.com/ApolloAuto/apollo/>.

参考文献

- 1 Kalra N, Paddock S M. Driving to safety: how many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Res Part A-Policy Pract*, 2016, 94: 182–193
- 2 Ren B T, Deng W W, Bai X S, et al. Technologies of virtual scenario construction for intelligent driving testing. *J Image Graph*, 2021, 26: 1–12 [任秉韬, 邓伟文, 白雪松, 等. 面向智能驾驶测试的仿真场景构建技术综述. *中国图象图形学报*, 2021, 26: 1–12]
- 3 Ding W, Chen B, Xu M, et al. Learning to collide: an adaptive safety-critical scenarios generating method. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020. 2243–2250
- 4 Althoff M, Lutz S. Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles. In: *Proceedings of IEEE Intelligent Vehicles Symposium*, 2018. 1326–1333
- 5 Klischat M, Althoff M. Generating critical test scenarios for automated vehicles with evolutionary algorithms. In: *Proceedings of IEEE Intelligent Vehicles Symposium*, 2019. 2352–2358
- 6 Calò A, Arcaini P, Ali S, Hauer F, et al. Generating avoidable collision scenarios for testing autonomous driving systems. In: *Proceedings of IEEE 13th International Conference on Software Testing, Validation and Verification*, 2020. 375–386
- 7 Tuncali C, Pavlic T, Fainekos G. Utilizing S-TaLiRo as an automatic test generation framework for autonomous vehicles. In: *Proceedings of IEEE 19th International Conference on Intelligent Transportation Systems*, 2016. 1470–1475
- 8 Tang Y, Zhou Y, Zhang T, et al. Systematic testing of autonomous driving systems using map topology-based scenario classification. In: *Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering*, 2021. 1342–1346
- 9 Viswanadha K, Indaheng F, Wong J, et al. Addressing the IEEE AV test challenge with Scenic and VerifAI. In: *Proceedings of IEEE International Conference on Artificial Intelligence Testing*, 2021. 136–142
- 10 Li G, Li Y, Jha S, et al. AV-FUZZER: finding safety violations in autonomous driving systems. In: *Proceedings of IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*, 2020. 25–36
- 11 Tian H, Jiang Y, Wu G, et al. MOSAT: finding safety violations of autonomous driving systems using multi-objective genetic algorithm. In: *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2022. 94–106
- 12 Wang J, Pun A, Tu J, et al. AdvSim: generating safety-critical scenarios for self-driving vehicles. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 9909–9918
- 13 Fremont D J, Kim E, Dreossi T, et al. Scenic: a language for scenario specification and data generation. *Mach Learn*, 2022. doi: 10.1007/s10994-021-06120-5
- 14 Rong G, Shin B H, Tabatabaee H, et al. LGSVL simulator: a high fidelity simulator for autonomous driving. In: *Proceedings of IEEE 23rd International Conference on Intelligent Transportation System*, 2020. 1–6
- 15 Dreossi T, Fremont D J, Ghosh S, et al. VerifAI: a toolkit for the formal design and analysis of artificial intelligence-based systems. In: *Proceedings of International Conference on Computer Aided Verification*, 2019. 432–442

AdvSce: safety-critical scenario generation for testing autonomous driving systems

Changwen LI^{1,2}, Rongjie YAN^{1,2*} & Jian ZHANG^{1,2}

1. *State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China;*

2. *University of Chinese Academy of Sciences, Beijing 100049, China*

* Corresponding author. E-mail: yrj@ios.ac.cn

Abstract AdvSce is a tool capable of generating safety-critical scenarios for testing autonomous driving systems. The tool introduces additional traffic participants that can perturb the behavior of the ego vehicle from existing scenarios. Existing scenario-generation methods usually require a predefined range of parameters and generate scenarios by using a black-box optimization algorithm. Unlike these methods, AdvSce can automatically analyze the behavior of the ego vehicle and dynamically decide the range of parameters that challenge the safety of the ego vehicle. It is highly automated and efficient in generating safety-critical scenarios. We conducted extensive experiments to compare the performance of AdvSce with various existing tools for testing the autonomous driving system Apollo 7.0 in the simulator LGSVL. The experimental results show that AdvSce is more effective in generating safety-critical scenarios.

Keywords autonomous driving system, simulation-based testing, safety-critical scenario generation, search-based optimization, behavioral sequence extraction