

测试。

以上几种方案的核心是通过 Monkey 方案生成随机操作事件对 App 进行压力测试，但 Monkey 方案本身也存在以下问题。

(1) 无效操作较多，测试效率较低，在一般情况下需要执行较长时间才能覆盖到一定比例的页面。

(2) 发版过程中重复的多轮任务无相关性，无法利用之前的测试数据。

(3) 压测操作事件完全随机，和用户的实际操作行为相差较大。

针对以上问题，结合业界和学术界的理论，我们决定通过以下两种途径对稳定性方案进行优化。

- 稳定性测试事件的优化。
- 引入线上用户行为数据。

前者通过对测试事件生成算法的优化，让原来完全随机生成的事件向更优化的方向演进；后者通过引入用户实际操作的路径数据，让压测更贴近用户行为并尽可能地覆盖用户访问较多的核心页面路径。

第一种途径是对稳定性测试事件的优化，我们最终选择了通过遗传算法（Genetic Algorithm, GA）改进操作事件，提升测试效率，并利用多轮任务测试不断优化测试结果。

第二种途径主要用在小程序的测试场景中，因为小程序承载的业务多种多样，操作行为和展示样式也千差万别。同时随着小程序业务的不断增长，种类也会越来越多，只有将线上用户行为作为输入，才能较好地选择满足不同小程序的方案。

## 1. 基于遗传算法的稳定性测试方案

学术界有很多与自动化测试相关的研究，比较经典的有 Ke Mao 等人在论文 *Sapienz: Multi-objective Automated Testing for Android Applications* 中提出的基于遗传算法的自动化测试，从覆盖页面个数、序列长度和崩溃数目三个方面对测试序列进行优化。可以总结为希望用尽可能少的时间（短序列长度）覆盖尽可能多的页面，并找到尽可能多的崩溃数目。该方法有两个主要特点。

(1) 使用基于搜索的遗传算法指导自动化测试，而不是使用传统的随机算法或者基于模型的方法。

(2) 设计合适的“适应度函数”（Fitness Function）指导 GA 进化，同时考虑了以上三个方面，将问题转化成多目标优化问题，每次成功的进化都必须满足这个多目标。