Time-aware Test Execution Scheduling for Cyber-Physical Systems

Morten Mossige¹  Arnaud Gotlieb²  Helge Spieker²  Hein Meling³  Mats Carlsson⁴


Testing cyber-physical systems involves the execution of test cases on target-machines equipped with the latest release of a software control system. When testing industrial robots, it is common that the target machines need to share some common resources, e.g., costly hardware devices, and so there is a need to schedule test case execution on the target machines, accounting for these shared resources. With a large number of such tests executed on a regular basis, this scheduling becomes difficult to manage manually. In fact, with manual test execution planning and scheduling, some robots may remain unoccupied for long periods of time and some test cases may not be executed. We introduce TC-Sched, a time-aware method for automated test case execution scheduling. TC-Sched uses Constraint Programming to schedule tests to run on multiple machines constrained by the tests’ access to shared resources, such as measurement or networking devices. We will further discuss challenges and requirements encountered when automating testing for industrial robots.

Keywords: Software Testing; Continuous Integration; Test Scheduling; Cyber-Physical Systems

Continuous integration (CI) aims to uncover defects in early stages of software development by frequently building, integrating, and testing software systems. When applied to the development of cyber-physical systems (CPS), which can simply be seen as communicating embedded software systems. The process may include running integration test cases involving real hardware components on different machines or machines equipped with specific devices. In the last decade, CI has been recognized as an effective process to improve software quality at reasonable costs [DMG07, St09, ERP14].

Different from traditional testing methods, running a test case in CI requires tight control over the round-trip time, that is, the time from when a source code change is committed until the success or failure of the build and test processes is reported back to the developer. Admittedly, the easiest way to minimize the round-trip time is simply to execute as many tests as possible in the shortest amount of time. But the achievable parallelism is limited by the availability of scarce global resources, such as a costly measurement instrument or

¹ ABB Robotics, Bryne, Norway & University of Stavanger, Stavanger, Norway morten.mossige@uis.no
² Simula Research Laboratory, Norway firstname@simula.no
³ University of Stavanger, Stavanger, Norway hein.meling@uis.no
⁴ RISE SICS, Kista, Sweden, mats.carlsson@ri.se

doi:10.18420/se2019-36
network device, and the compatible machines per test case, targeting different machine architecture and operating systems. These global resources are required in addition to the machine executing the test case and thereby require parallel adjustments of the schedule for multiple machines. Thus, computing an optimal test schedule with minimal round-trip time is a challenging optimization problem. Since different test cases have different execution times and may use different global resources that are locked during execution, finding an optimal schedule manually is mostly impossible. Nevertheless, manual scheduling still is state-of-the-practice in many industrial applications, besides simple heuristics. In general, successful approaches to scheduling use techniques from Constraint Programming (CP) and Operations Research (OR), additionally metaheuristics are able to provide good solutions to certain scheduling problems.

Informally, the optimal test scheduling problem (OTS) is to find an execution order and assignment of all test cases to machines. Each test case has to be executed once and no global resource can be used by two test cases at the same time. The objective is to minimize the overall test scheduling and test execution time. The assignment is constrained by the compatibility between test cases and machines, that is, each test case can only be executed on a subset of machines.

We introduce TC-Sched, a time-aware method to solve OTS. Using the Cumulatives [AB93, BC02] global constraint, we propose a cost-effective constraint optimization search technique. This method allows us to 1) automatically filter invalid test execution schedules, and 2) find among possible valid schedules, those that minimize the global test execution time (i.e., makespan). TC-Sched has been developed and deployed with ABB Robotics, Norway. An extensive experimental evaluation is conducted over test suites from industrial software systems, namely an integrated control system for industrial robots and a product line of video-conferencing systems. The primary goal in this paper is to demonstrate the scalability of the proposed approach for CI processes involving hundreds of test cases and tens of machines, which corresponds to a realistic development environment. Furthermore, we demonstrate the cost-effectiveness of integrating our approach within an actual CI process.

References


