

Automation of Crash Test Scheduling Using Mathematical Optimization

Toshihiro Watanabe¹⁾ Junichi Abe¹⁾ Hiroshi Yoshimoto¹⁾ Toshiki Terabe¹⁾ Yuya Osamura¹⁾
 Kosuke Suzuki²⁾ Tetsuro Sakamoto²⁾ Tomoko Ikuta²⁾

1) Toyota Motor Corporation 1 Toyota-cho, Toyota, Aichi, 471-8571 Japan
 2) Fixstars Corporation 1-1-1 Shibaura, Minato, Tokyo, 105-0023, Japan

KEY WORDS: Software and Its Underlying Technologies, Data Visualization, Test Scheduling, Crash Test, Mathematical Optimization, RCPSP [E3]

Automation of crash test scheduling using mathematical optimization is proposed to reduce dependence on the experience and know-how of seasoned veterans for test scheduling. Crash tests involve numerous test types, facilities, instruments, human workers, and dummies, each with complex operational rules such as skill requirements, shared use, transportation, and mandatory inspections after a certain number of uses. Because of these characteristics, it is difficult to improve efficiency using conventional scheduling tools as-is, and consequently it took a long time to finalize the schedule.

We formulate the crash test scheduling problem as an extension of the Resource-Constrained Project Scheduling Problem (RCPSP) and implement it using the CP-SAT solver in Google OR-Tools. The structure of the improved RCPSP-based scheduling model, including the handling of multiple resource types and complex constraints, is summarized in Table 1. The model introduces skill-based resources, allowing each job to require specific skills in addition to resource quantities, and represents resource downtime for inspections by linking usage counts to inspections that restore availability; typical examples of skill-based allocation and inspection-induced downtime are illustrated conceptually in Figure 1. We implemented multiple objective functions to reflect schedulers' intentions, such as maximizing the number of tests within the scheduling horizon, respecting resource priorities, and minimizing deviation from requested test dates.

Application to realistic crash test scenarios demonstrates that the proposed method can automatically generate feasible schedules that obey detailed operational rules for workers, facilities, and dummies. The results enable identification of bottleneck resources—such as highly loaded workers or frequently inspected dummies—thus supporting decisions to increase resources or shorten processes. Compared with manual scheduling, the optimization achieves substantial reductions in scheduling time while maintaining practicality of the schedules. The modeling framework presented in Table 1 and Figure 1 is expected to be extendable to other vehicle test domains that involve complex multi-resource constraints.

Table1 Improved RCPSP model

Problem	Schedule image
Improved RCPSP	resource A-1
	resource A-2
	skill resource A-1
	skill resource A-2
	skill resource B-1
	skill resource B-2
	skill resource B-2
	Each job has its own unique constraint regarding the order of resource assignment. There are options for resource assignment, and one must be selected. It is possible to assign multiple resources per job. Added the concept of skill resources. Considered resource downtime (periods of inactivity).

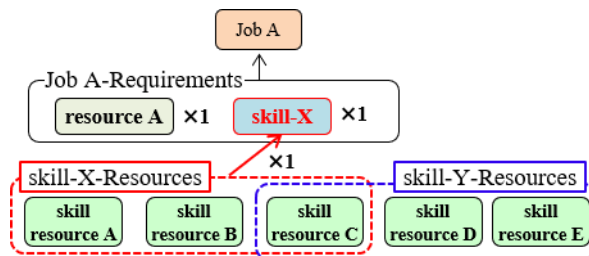


Fig.1-1 Overview of skill-resources

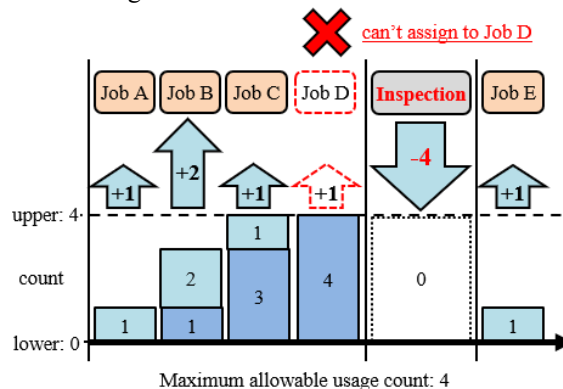


Fig.1-2 Overview of inspection for resources
 Fig.1 Examples of improved RCPSP model