WaP: Computing the Execution Probability of Jobs with Replication in Mixed-Criticality Schedules

Antonin Novak
Faculty of Electrical Engineering &
Czech Institute of Informatics, Robotics, and Cybernetics
Czech Technical University in Prague
Prague, Czech Republic
antonin.novak@cvut.cz

Zdenek Hanzalek
Czech Institute of Informatics, Robotics, and Cybernetics
Czech Technical University in Prague
Prague, Czech Republic

Premysl Sucha
Czech Institute of Informatics, Robotics, and Cybernetics
Czech Technical University in Prague
Prague, Czech Republic

***Abstract*—This extended abstract represents the journal paper published in [13]. In that paper, we study the computation of the execution probability of jobs with uncertain execution times in a static mixed-criticality schedule. In contrast to the majority of research in mixed-criticality systems that work with task models where the jobs are gradually revealed to the scheduler, we assume a time-triggered environment where the offline scheduler generates a static schedule [14], [15], [17]. The execution time of the mixed-criticality jobs is not known in advance and is revealed during the online execution. An online execution policy is designed to handle the prolongations of execution times and escalations of the system mode. The policy may eventually reject some of the low-criticality jobs under some execution scenarios, thus affecting the execution probability of the jobs.**

This paper deals with the complexity and the method for analysis of the execution probability of mixed-criticality jobs in a static schedule. To overcome the rigidity of static scheduling, we introduce job replication, i.e., scheduling multiple time slots for a single job, as a new mechanism for increasing the execution probability of jobs. We show that the general problem with job replication becomes as hard as the counting variant of 3-SAT problem. To compute the execution probability, we propose an algorithm utilizing the framework of Bayesian networks. The proposed methodology demonstrates an interesting connection between schedules with uncertain execution and probabilistic graphical models.

***Index Terms*—execution analysis, mixed-criticality, time-triggered, job replication, Bayesian networks, computational complexity**

I. Introduction

In contrast to the majority of research in mixed-criticality systems [3], [4], [20], we assume a time-triggered environment where the offline scheduler generates a static schedule [1], [14], [15], [17], [18]. Nevertheless, the basic idea is somewhat similar to classical Vestal's model [20] with the key differences described below. We assume that the execution time of the mixed-criticality jobs is not known and follows a probability distribution [13], [17]. The actual execution time is revealed during the online execution of the schedule. To compensate for the prolongations of execution times and the elevations of the system mode observed during the runtime (e.g., in Figure 1 at time $t = 5$, system mode is raised to $e_t = 2$), we employ an online policy to govern the execution of the schedule. The policy may eventually reject some of the low-criticality jobs under some execution scenarios, thus affecting their execution probability.

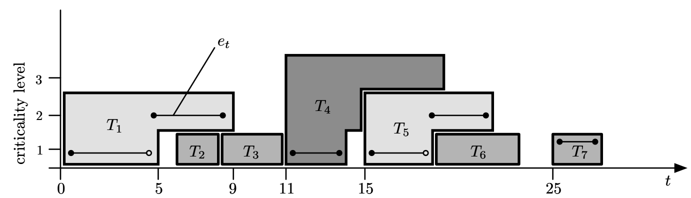
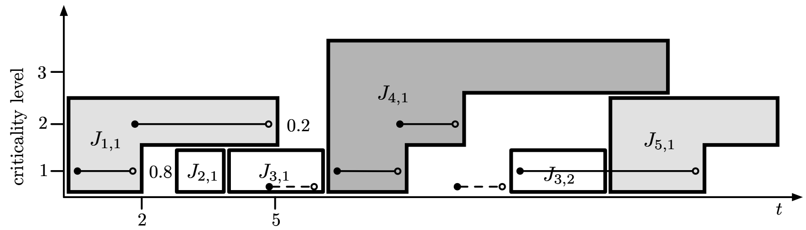


Fig. 1: Schedule with mixed-criticality jobs and an execution scenario e_t.

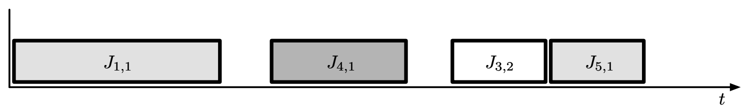
In our mixed-criticality model [8], [13]–[15], we also use the so-called *match-up* property, meaning that the system mode can be lowered and returned to nominal operation mode once it is escalated. This can be seen, e.g., in Figure 1, where after execution of job T_5, the execution scenario matches up with the nominal system mode, i.e., $e_t = 1$ at $t = 25$. This is, in fact, similar to the concepts used in Flexible Mixed-Criticality Systems [6], [11]. In this extended abstract, we will focus on a more intuitive explanation with an application example rather than giving formal definitions first. A more detailed description of the assumed mixed-criticality system model for non-preemptive scheduling in time-triggered environments can be found in the series of works [8], [13]–[15], [17]. For experimental evaluation of the model on a real hardware testbed, please see [9].

Although the static scheduling increases the predictability of

This work was supported by the EU and the Ministry of Industry and Trade of the Czech Republic under the Project OP PIK CZ.01.1.02/0.0/0.0/20_321/0024399. Furthermore, this work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS22/167/OHK3/3T/13.



(a) Mixed-criticality schedule with replication with five jobs where J_3 has two replicas.



(b) An execution scenario.

Fig. 2: Mixed-criticality schedule with three criticality levels with one of the possible execution scenarios.

the system (i.e., its behavior is given by a static schedule which can be analyzed offline), the execution of critical jobs may occasionally require to reject less critical jobs during online execution (e.g., $J_{2,1}$ and $J_{3,1}$ in Figure 2b). Thus, careful scheduling of jobs needs to be used to mitigate the degradation of the execution probability of non-critical jobs without affecting the requirements of critical jobs. To optimize the execution probability of jobs in a schedule, a scheduling algorithm needs to assess the quality (i.e., the objective function) of the current schedule in order to drive the search toward a good solution. Hence, the computation of the objective function of a schedule is the central component of any algorithm that produces high-quality schedules.

A. Contributions

In paper [13], we study job replication, which is a mechanism for increasing the execution probability of jobs in time-triggered mixed-criticality schedules. Specifically, the main contributions are:

– We introduce the concept of replication to mixed-criticality schedules as a mechanism for increasing the execution probability of jobs.
– We show that the general problem of computing execution probability for a job in a mixed-criticality schedule with replication as hard as counting the number of satisfiable assignments of a propositional formula in normal conjunction form, in contrast to the known polynomial-time algorithm for the case without replication.
– We solve the problem by a reduction to the probabilistic inference in a suitably defined Bayesian network.
– We show that the cases of reasonable interests (i.e., constant-bounded number of criticality levels and the maximum number of replicas per job) can be solved in polynomial time in the number of jobs, which enables practical usage of job replication.

II. Static Mixed-Criticality Systems with Job Replication

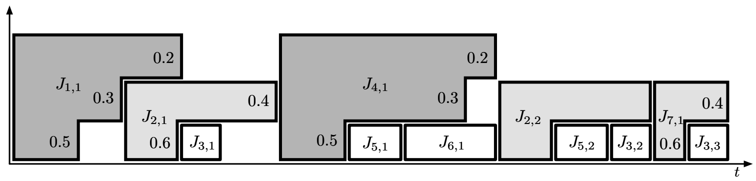
In this section, we describe an application example to illustrate the main concepts of static mixed-criticality systems with job replication. An extension of our mixed-criticality model by a frequency dimension has been implemented and tested in practice on a real-life testbed, e.g., for message scheduling in 5G NR (new radio) networks [9].

A. Application example

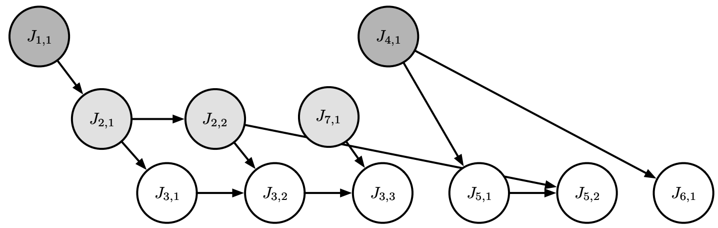
Consider a message scheduling problem on a shared communication bus in modern vehicles. Safety-related standards such as ASIL (Automotive Safety Integrity Levels) [2] introduce the existence of messages with several levels of criticality, such as:

– messages of high criticality (criticality 3) are used for safety-related functionalities (their failure may result in death or severe injury to people), such as steering;
– messages of medium criticality (criticality 2) are used for mission-related functionalities (their failure may prevent activity from being successfully completed), such as parking assist;
– messages of low criticality (criticality 1) are typically used for infotainment functionalities, such as automotive navigation system.

The messages are transmitted via the bus at the moments defined by the static time-triggered schedule [10] (e.g., the static segment of a FlexRay bus [7]), which improves determinism and predictability. The goal is to compute an objective function reflecting statistical properties of a given static schedule that accounts for disruption of the communication according to the message criticality. In real-life environments, the execution of jobs is affected by various sources of uncertainty, causing, e.g., transmission delays. In the above example, criticality expresses



(a) Example schedule $\boldsymbol{s}$.



(b) Corresponding Bayesian network $G(\boldsymbol{s})$.

Fig. 3: Representation of a schedule by a Bayesian network.

TABLE I: Execution probabilities of individual replicas in schedule $\boldsymbol{s}$.

	$J_{1,1}$	$J_{2,1}$	$J_{2,2}$	$J_{3,1}$	$J_{3,2}$	$J_{3,3}$	$J_{4,1}$	$J_{5,1}$	$J_{5,2}$	$J_{6,1}$	$J_{7,1}$
$\Pr\{J_{i,q} \succeq 1\}$	1.0	0.8	0.2	0.68	0.32	0.0	1.0	0.5	0.46	0.5	1.0

the commitment to the transmission when the original transmission is prolonged. Therefore, several transmission attempts are awarded to messages with a high criticality, whereas for low-criticality messages, it might be just a single one.

Figure 2a shows an example of a mixed-criticality static schedule with six job replicas. Each job has a given integer criticality, as it is seen on the vertical axis. For example, job $J_{4,1}$ has a criticality of three, job $J_{1,1}$ has a criticality of two, while $J_{2,1}$ has a criticality of one. Notice that $J_{3,1}$ and $J_{3,2}$ are two replicas of the same job; hence, they have the same parameters but different start times. Job $J_{1,1}$ has execution time 2 time units with probability 0.8, and 5 time units with probability 0.2. The considered values of execution times are derived from the (empirical) cumulative distribution function with respect to the selected probability thresholds [15], [20].

Mixed-criticality schedules contain several alternative execution scenarios, with the one being selected based on the realized execution times of jobs that occur during the runtime execution. It is dispatched in such a way that in any of these scenarios, all highly critical jobs are executed, rejecting jobs with lower criticality only when a highly critical one is prolonged. This leads to more efficient resource usage since the low-criticality jobs may use the resource when the critical ones are not prolonged. To compensate for unexpected prolongations of critical jobs observed at the runtime, some of the less critical ones might not be executed under the specific realization of execution times. This can be seen in Figure 2b, where jobs $J_{2,1}$ and $J_{3,1}$ are rejected if realized execution time of $J_{1,1}$ is equal to 5, happening with probability 0.2. However, the second replica $J_{3,2}$ was executed later on. Finally, we note that, e.g., $J_{1,1}$ is never rejected since it does not share its execution time with any other job with higher criticality. Therefore, the execution probability of $J_{1,1}$, denoted as P_1, is 1 while execution probability of $J_{2,1}$ is $P_2 = 0.8$.

In this paper, we deal with the problem of computing execution probabilities P_i of jobs J_i with replication for the given fixed schedule.

III. Results

A. Time complexity of the problem

We show that the general problem where either the maximum number of criticality levels $\mathcal{L}$ or the maximum number of replicas per job $\mathcal{R}$ is bounded by a polynomial in the number of jobs and the other is equal to some chosen constant remains $\#\mathcal{P}$-hard. We remind that $\#\mathcal{P}$ is a class of *counting problems*, i.e., a set of problems that count the number of accepting paths in a polynomial-time non-deterministic Turing machine [19]. An example of a problem contained in $\#\mathcal{P}$ is the following: What is the number of spanning trees in the given connected simple graph? A problem is said to be $\#\mathcal{P}$-hard, if for every problem in $\#\mathcal{P}$, there exists a polynomial-time counting reduction to it [5].

First, we show that deciding whether a job has a non-zero probability of being executed is as hard as determining whether a CNF (*conjunctive normal form*) formula is satisfiable.

Proposition 1: There exists a finite number of maximum replicas per job $\mathcal{R}$ such that deciding whether $P_i > 0$ for some job J_i is $\mathcal{NP}$-complete.

The reduction suggests that the problem remains hard even for a constant number of the maximum job replicas, i.e., $\mathcal{R} = 4$. Moreover, we will show that a non-constant number of criticality levels is not the only source of hardness. Indeed, the problem remains hard, assuming a constant number of criticality levels when the maximum number of replicas is not fixed to a constant.

Proposition 2: There exists a finite number of criticality levels $\mathcal{L}$ such that determining whether $P_i > 0$ for some job J_i is $\mathcal{NP}$-complete.

To compute the exact execution probability of jobs in a given schedule, we propose an algorithm based on the theoretical framework of the Bayesian network.

B. Algorithm for computation of the execution probability

We show how the statistical properties of mixed-criticality schedules with replication can be described with Bayesian networks. A Bayesian network $G = (V, A)$ is a probabilistic directed acyclic graphical model representing the joint distribution over the set V of random variables using conditional dependencies defined by edges A. The network contains one vertex for each job replica in the schedule. The directed edges connect the vertices if the corresponding job replica can affect the execution of the other (i.e., a conflicting higher-criticality job or preceding replica). Since the execution time of jobs is uncertain, we can view the job replicas as random variables. Then, the execution policy defines a joint probability distribution $\Pr\{J_{1,1}, \ldots, J_{n,n_i}\}$ over the given schedule that assigns a probability to each execution scenario.

To represent this distribution, we use Bayesian networks (BN) [16], which can be seen as an efficient way of representing joint distributions. An example of such a static mixed-criticality schedule and its representation by a Bayesian network can be seen in Figure 3. To compute the execution probabilities of jobs, one can use algorithms, such as variable elimination or junction trees, for the inference in Bayesian networks. The outcome of this procedure is the execution probability of each job, which can be seen in Table I. For example, the result of the analysis in Table I reveals that replica $J_{3,3}$ cannot be executed, thus, can be removed in the design phase. Please see [13] for more details.

IV. Conclusion

In this paper, we have introduced the job replication mechanism for the static time-triggered mixed-criticality model to help to overcome the rigidity of static scheduling. Job replication, i.e., scheduling a single job with multiple occurrences, increases the execution probability of jobs but introduces additional computation complexity for the analysis of the resulting static schedules. We have shown that the complexity is affected by two natural parameters—the number of criticality levels and the maximum number of replicas per job. To practically solve the problem of the computation of the execution probability, we have proposed a reduction to the theoretical framework of Bayesian networks for which many efficient algorithms exist.

For future research, we suggest looking into the integration of event-triggered and time-triggered paradigms using techniques such as schedule graph abstraction [12]. One of the possible applications might be to use static scheduling for the most critical functionality, as it often consists of core and essential components that do not change very often, but its correctness needs to be formally verifiable. However, static approaches lack the flexibility and efficiency of event-triggered environments, which could be used for applications with smaller criticality. Therefore, combining the two might bring the best of both paradigms.

References

[1] Lalatendu Behera and Purandar Bhaduri. Time-triggered scheduling for multiprocessor mixed-criticality systems. In *Distributed Computing and Internet Technology*, pages 135–151, Cham, 2018. Springer International Publishing.

[2] Ron Bell. Introduction to IEC 61508. In *Proceedings of the 10th Australian workshop on Safety critical systems and software-Volume 55*, pages 3–12. Australian Computer Society, Inc., 2006.

[3] Alan Burns and Robert I. Davis. A survey of research into mixed criticality systems. *ACM Comput. Surv.*, 50(6):82:1–82:37, November 2017.

[4] Alan Burns, Robert I Davis, Sanjoy Baruah, and Iain Bate. Robust mixed-criticality systems. *IEEE Transactions on Computers*, 67(10):1478–1491, 2018.

[5] Nadia Creignou and Miki Hermann. *On P completeness of some counting problems*. PhD thesis, INRIA, 1993.

[6] Xinyang Dong, Gang Chen, Mingsong Lv, Weiguang Pang, and Wang Yi. Flexible mixed-criticality scheduling with dynamic slack management. *Journal of Circuits, Systems and Computers*, 30(10):2150306, 2021.

[7] Jan Dvořák and Zdeněk Hanzálek. Using two independent channels with gateway for flexray static segment scheduling. *IEEE Transactions on Industrial Informatics*, 12(5):1887–1895, 2016.

[8] Zdenek Hanzalek, Tomas Tunys, and Premysl Sucha. An analysis of the non-preemptive mixed-criticality match-up scheduling problem. *Journal of Scheduling*, 19(5):601–607, Oct 2016.

[9] Xi Jin, Yu Tian, Chi Xu, Changqing Xia, Dong Li, and Peng Zeng. Mixed-criticality industrial data scheduling on 5G NR. *IEEE Internet of Things Journal*, 2021.

[10] H. Kopetz. *Event-triggered versus time-triggered real-time systems*, pages 86–101. Springer Berlin Heidelberg, Berlin, Heidelberg, 1991.

[11] Jaewoo Lee and Jinkyu Lee. MC−FLEX: Flexible mixed-criticality real-time scheduling by task-level mode switch. *IEEE Transactions on Computers*, 71(8):1889–1902, 2022.

[12] Mitra Nasri and Bjorn B Brandenburg. An exact and sustainable analysis of non-preemptive scheduling. In *2017 IEEE Real-Time Systems Symposium (RTSS)*, pages 12–23. IEEE, 2017.

[13] Antonín Novák and Zdenek Hanzalek. Computing the execution probability of jobs with replication in mixed-criticality schedules. *Annals of Operations Research*, 2022.

[14] Antonín Novák, Premysl Sucha, and Zdenek Hanzalek. Efficient algorithm for jitter minimization in time-triggered periodic mixed-criticality message scheduling problem. In *Proceedings of the 24th International Conference on Real-Time Networks and Systems*, RTNS '16, page 23–31, New York, NY, USA, 2016. Association for Computing Machinery.

[15] Antonín Novák, Premysl Sucha, and Zdenek Hanzalek. Scheduling with uncertain processing times in mixed-criticality systems. *European Journal of Operational Research*, 279(3):687–703, 2019.

[16] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[17] Yasmina Seddik and Zdenek Hanzalek. Match-up scheduling of mixed-criticality jobs: Maximizing the probability of jobs execution. *European Journal of Operational Research*, 262(1):46 – 59, 2017.

[18] Jens Theis, Gerhard Fohler, and Sanjoy Baruah. Schedule table generation for time-triggered mixed criticality systems. *Proceedings of the 1st International Workshop on Mixed Criticality Systems, RTSS*, pages 79–84, 2013.

[19] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189 – 201, 1979.

[20] Steve Vestal. Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance. In *Real-Time Systems Symposium, 2007. RTSS 2007. 28th IEEE International*, pages 239–243. IEEE, 2007.