Is there any optimal fixed-priority scheduling algorithm for probabilistic dependant tasks on one processor ?

Ismail Hawila
StatInf and Inria, France
Email: ismail.hawila@inria.fr

Liliana Cucu-Grosjean
Inria, France
Email: liliana.cucu@inria.fr

Slim Ben Amor
StatInf, France
Email: slim.ben-amor@statinf.fr

The evolution of critical software in industries like automotive or space includes the consideration of dependent programs, while more general operating systems are integrated. Those operating systems do not necessarily come with the possibility to choose the scheduling algorithm. For instance the operating systems NuttX comes with a fixed-priority assignment policy and finding the appropriate policy is expected to decrease the time latency of these operating systems. Moreover, automatic generation of programs by middleware, like ROS, excludes baremetal implementations that have been traditionally used by space and avionic industries. In this paper we consider the scheduling problem of directed acyclic graphs of real-time tasks or programs on one processor. The time parameters of the tasks are defined by probability cumulative distributions and the scheduling algorithm is a preemptive fixed-priority policy.

The uniprocessor scheduling problem of real-time tasks with time parameters defined by probability cumulative distributions has been originally studied for tasks with probabilistic execution times [1], [2] and then included probabilistic inter-arrival times [3], [4]. In general, the authors have provided results for preemptive fixed-priority scheduling policies, except for [2], [5], where the authors provide schedulability analyses applicable to dynamic schedulers like Earliest Deadline First. An interested reader may find more results on uniprocessor scheduling problems of real-time tasks with time parameters defined by probability cumulative distributions in [6]. In this paper, we name these systems, probabilistic real-time systems.

For the uniprocessor fixed-priority scheduling problem of probabilistic real-time systems with probabilistic execution times, the synchronous arrival time instant is proved worst-case in [4] and later invalidated in [7], while none of these results consider the utilization as a necessary input to define a critical instant. To the best of our knowledge, the most recent result, correcting [4] and answering to concerns raised in [7], is presented in [8].

The scheduling problem of directed acyclic graphs (DAGs) of real-time programs or tasks has received a constant attention from the real-time community, but either the results cover the case of a unique inter-arrival times for all dependent tasks, or a task has its internal program structure defined by a directed graph. Few results are provided for cases different from the ones cited previously and we detail their differences with respect to our current contribution at the end of this section. In this paper, we use DAGs to define precedence constraints between different tasks.

Concerning the case when all dependent tasks have an unique inter-arrival time, first contributions are proposed in the Operations Research community [9] and authors consider scheduling decisions at job level and those results are extended to more real-time oriented problems like dynamic scheduling in [10]. The most general results on the feasibility of dependent tasks with an unique inter-arrival time are proposed recently in [11]. This later paper also provides, the least pessimist bound for feasibility intervals of these systems when scheduling decisions are taken at the task level.

Concerning the case when tasks are independent and their internal structure is defined by directed graphs one may cite [12] as the original paper introducing such model, while later results provide optimal scheduling algorithms for shared resources [13], [14], or related complexity results [15]. All previous results are provided for the case of preemptive scheduling algorithms, while non-preemptive scheduling results have been introduced in [16]. To the best of our knowledge, the existing results extending such graph task models by introducing probabilistic descriptions for time parameters are presented in [5], [17] and they consider probabilistic execution times, while the inter-arrival times are not associated probabilities. Thus, this is the first paper suggesting fixed-priority preemptive scheduling results for real-time tasks with probabilistic (worst-case) execution times and inter-arrival times described by probability distributions on the case of one processor.

I. Our task model and notations

We consider a set τ of n real-time tasks τ_i scheduled according to a preemptive fixed-priority scheduling policy on a processor π. A task $\tau_i$is defined by C_i its execution time, T_i its minimal inter-arrival time and D_i its deadline. The execution and inter-arrival times of a task τ_i are defined by cumulative distribution functions F_{C_i} and F_{T_i} respectively defined in eq. 1 and 2 where $\Omega_0 = Q \times I$ is the product space between Q the set of possible states of the processor π and I the set of all possible input values of tasks τ_i [8].

$$F_{C_i}(c) = P_{C_i}((-\infty, c)) = P(\omega_0 \in \Omega_0 : C_i(\omega_0) \leq c) \quad (1)$$

$$F_{T_i}(t) = P_{T_i}((-\infty, t)) = P(\omega_0 \in \Omega_0 : C_i(\omega_0) \leq t) \quad (2)$$

One may underline that $F_{C_i}(c)$ defines the probability for the execution time C_i to be smaller than c. Indeed, within the real-time community, one is interested in the exceedance function $1 - F_{C_i}$ which defines the probability for the execution time C_i to be larger than c. Meanwhile, the $F_{T_i}(t)$ defines the probability for the inter-arrival time T_i to be smaller than t. A task $\tau_i, \forall i \leq n$ has instances released such that two consecutive instances, or jobs, $\tau_{i,k}$ and $\tau_{i,k+1}, \forall k > 0$ are separated by a time slot t. In our case, the deadlines are identically distributed as the inter-arrival times and their value for each instance of a task is equal to the inter-arrival time of that instance. We consider precedence constraints between tasks that are defined by a directed acyclic graph (DAG) $G = (\tau, E)$. A directed edge (τ_i, τ_j) belongs to E if a job $\tau_{i,k}$ is released before a job $\tau_{j,l}$ and $\tau_{j,l}$ waits for the completion of $\tau_{i,k}$ before it can start.

By the definition of precedence constraints, we consider precedence constraints imposing some order between different jobs of different tasks. This implies that two jobs that share data but their order of execution does not jeopardize the execution of other jobs that do not have an imposed order of execution. Therefore, the definition of the inter-arrival times and deadlines is done at design time such data are available on time as done in [18].

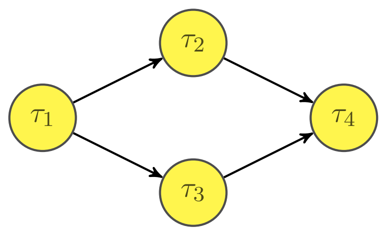

Fig. 1. A task set describing precedence constraints between tasks of a DAG.

The response time of a job $\tau_{i,k}$ is the time elapsed from its release to the end of its execution. A job that finishes its execution before its deadline is said to meet its deadline and on the contrary, a job that does not finish its execution before its deadline is said to miss its deadline.

For a set τ of tasks with all time parameters having an unique value with the probability of appearance equal to 1, we say that a schedule is feasible if all jobs of all tasks meet their deadlines and the precedence constraints are satisfied.

If there is, at least, one parameter of a task defined by a probability distribution, then one needs to understand the evolution of the remaining workload, i.e. remaining execution time to be executed after we reach the deadline, and study the existence of a time instant where the workload may be described by a bounded probability distribution. We say that a schedule is stable if for any time instant, the remaining workload is described by a bounded probability distribution. Consequently, we define a schedule of probabilistic tasks with precedence constraints defined by DAGs to be feasible if remaining workload is stable and all precedence constraints are satisfied. In practice, a feasible schedule of probabilistic tasks with precedence constraints defined by DAGs is a schedule with no job that has the distribution of its response time equal to infinity and all precedence constraints are satisfied.

Our open problem For a task set τ of n probabilistic tasks with precedence constraints defined by DAGs, is there any optimal fixed-priority preemptive scheduling algorithm? We understand here by optimal that if there exists a fixed-priority assignment providing a feasible schedule, then the optimal algorithm will find it. For our problem of tasks with precedence constraints defined by DAGs, a priority assignment should both ensure that the schedule is stable and the precedence constraints are satisfied.

REFERENCES

[1] M. K. Gardner and J. W. Liu, "Analyzing stochastic fixed-priority real-time systems," in *Tools and Algorithms for Construction and Analysis of Systems, 5th International Conference, TACAS '99*, ser. Lecture Notes in Computer Science, R. Cleaveland, Ed., vol. 1579. Springer, 1999, pp. 44–58.

[2] J. L. Díaz, D. F. García, K. Kim, C.-G. Lee, L. Lo Bello, J. M. López, S. L. Min, and O. Mirabella, "Stochastic analysis of periodic real-time systems," in *the 23rd IEEE Real-Time Systems Symposium (RTSS 2002)*, 2002, p. 289.

[3] L. Cucu and E. Tovar, "A framework for the response time analysis of fixed-priority tasks with stochastic inter-arrival times," *SIGBED Rev.*, vol. 3, no. 1, pp. 7–12, 2006.

[4] D. Maxim and L. Cucu-Grosjean, "Response time analysis for fixed-priority tasks with multiple probabilistic parameters," in *Proceedings of the IEEE 34th Real-Time Systems Symposium, RTSS 2013, Vancouver, BC, Canada, December 3-6, 2013*. IEEE Computer Society, 2013, pp. 224–235.

[5] S. Ben-Amor, D. Maxim, and L. Cucu-Grosjean, "Schedulability analysis of dependent probabilistic real-time tasks," in *the 24th International Conference on Real-Time Networks and Systems, RTNS 2016*. ACM, 2016, pp. 99–107.

[6] R. I. Davis and L. Cucu-Grosjean, "A survey of probabilistic schedulability analysis techniques for real-time systems," *Leibniz Trans. Embed. Syst.*, vol. 6, no. 1, pp. 04:1–04:53, 2019.

[7] K. Chen, M. Günzel, G. von der Brüggen, and J. Chen, "Critical instant for probabilistic timing guarantees: Refuted and revisited," in *IEEE Real-Time Systems Symposium, RTSS 2022,*. IEEE, 2022, pp. 145–157.

[8] K. Zagalo, "Stochastic analysis of stationary real-time systems. (analyse stochastique des systèmes temps-réel stationnaires)," Ph.D. dissertation, Sorbonne University, Paris, France, 2023.

[9] "A functional equation and its application to resource allocation and scheduling problems," *Management science*, vol. 16, no. 1, pp. 77–84, 1969.

[10] M. Silly, H. Chetto, and N. Elyounsi, "An optimal algorithm for guaranteeing sporadic tasks in hard real-time systems," in *the Second IEEE Symposium on Parallel and Distributed Processing, SPDP*, 1990, pp. 578–585.

[11] J. Goossens, E. Grolleau, and L. Cucu-Grosjean, "Periodicity of real-time schedules for dependent periodic tasks on identical multiprocessor platforms," *Real Time Syst.*, vol. 52, no. 6, pp. 808–832, 2016.

[12] M. Stigge, P. Ekberg, N. Guan, and W. Yi, "The digraph real-time task model," in *17th IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2011,*. IEEE Computer Society, 2011, pp. 71–80.

[13] H. Du, X. Jiang, M. Lv, T. Yang, and W. Yi, "Scheduling and analysis of real-time task graph models with nested locks," *J. Syst. Archit.*, vol. 114, p. 101969, 2021.

[14] M. Stigge, P. Ekberg, N. Guan, and W. Yi, "On the tractability of digraph-based task models," in *23rd Euromicro Conference on Real-Time Systems,*. IEEE Computer Society, 2011, pp. 162–171.

[15] S. K. Baruah, "Feasibility analysis of conditional DAG tasks is co-npnp-hard," in *the 29th International Conference on Real-Time Networks and Systems,*. ACM, 2021, pp. 165–172.

[16] L. Cucu and Y. Sorel, "Schedulability condition for systems with precedence and periodicity constraints without preemption," in *Proceedings of the 11th Real-Time Systems Conference (RTS)*, 2003.

[17] S. Ben-Amor and L. Cucu-Grosjean, "Graph reductions and partitioning heuristics for multicore DAG scheduling," *J. Syst. Archit.*, vol. 124, p. 102359, 2022.

[18] E. Ntaryamira, C. Maxim, T. Niyonsaba, and L. Cucu-Grosjean, "An efficient FIFO buffer management to ensure task level and effect-chain level data properties," in *2020 IEEE International Conference on Embedded Software and Systems, ICESS 2020, Shanghai, China, December 10-11, 2020*. IEEE, 2020, pp. 1–8.