Counterexamples for Analyzing Self-Suspending Tasks under EDF-Like Scheduling with Interactions with AI Tools

Michael Frichert*, Mario Günzel†, Marion Sudvarg‡, and Jian-Jia Chen*
*RWTH Aachen University, †MPI-SWS, ‡Missouri University of Science and Technology
*{michael.frichert,jian-jia.chen}@rwth-aachen.de, †mguenzel@mpi-sws.org, ‡msudvarg@mst.edu

***Abstract*—This article provides a short summary regarding how the authors interacted with AI tools in a journey of validating their concerns regarding analyzing self-suspending tasks under EDF-Like scheduling. Although the final counterexamples were mainly crafted by the authors for simplicity and readability, the AI tools made the validation processes simpler and more efficient than in the old days.**

I. Introduction

Self-suspension occurs when a task pauses its execution voluntarily and can be used to model various application scenarios in cyber-physical systems. In the dynamic self-suspension model, each task is described by $\tau_i = (C_i, S_i, D_i, T_i)$, where C_i denotes the worst-case execution time, S_i denotes the maximum suspension time, D_i denotes the relative deadline, and T_i denotes the inter-arrival time. Each job of τ_i may suspend itself arbitrarily often, but for at most S_i time units in total.

During ECRTS 2015 in Lund, Jian-Jia Chen and Geoffrey Nelissen presented counterexamples to previously published results on the schedulability analysis of self-suspending tasks, which led to a community effort to examine the literature, completed in 2017. The review paper [1] was published in 2019 and has provided very insightful information on how seemingly correct analyses can be unsound. At that time, the presented counterexamples were crafted by hand with careful examinations.

This article provides a short summary regarding how the authors interacted with AI tools in a journey of validating their concerns on the correctness of the presented analyses in [6] for self-suspending tasks under preemptive EDF scheduling. The resulting counterexamples are presented in [3].

II. Process to Find Counterexamples

The journey started on October 9th, 2026. Mario and Jian-Jia started a conversation about the paper [6], published in 2025. They reached a conclusion that none of them can actually verify the results, and there is a question whether the presented improved schedulability analysis in [6] actually fixes the incorrect analysis of EDF scheduling by Devi [2], reported in [4]. In the following we describe how Michael and Jian-Jia have interacted independently with AI tools in order to invalidate the analysis.

A. Sufficient Condition

In [6], Wang et al. have presented two sufficient schedulability analyses for EDF-Like (EL) scheduling, namely a jitter-based response time analysis called JRTA (Theorem 6 [6]) and a blocking-based analysis called ISA (Theorem 11 [6]). EL scheduling includes preemptive EDF as a special case [5].

a) Michael's Effort: Michael's effort started by briefly overviewing both schedulability analyses in [6] on October 9th. As the necessary condition in Theorem 11 [6] was not formally defined, Michael has instructed Anthropic Claude (Model Opus 5.5) to explain how the schedulability analysis can be applied. In the same prompt, he has further indicated his concerns on the correctness of the paper and asked whether it is possible to create a counterexample. Without further instruction, the AI model has reproduced both analyses in Python and created a basic three-step model-checking pipeline for a task set containing two tasks:

1) Generate task sets randomly.
2) Filter all task sets that were accepted by ISA.
3) For each accepted task set, generate all possible release and suspension patterns in discrete time. Afterwards, check whether a deadline miss occurs in any pattern.

After approximately 20 minutes of computation, the model has created a comprehensive summary presenting a counterexample for ISA. Then, Michael has validated the initial counterexample for Theorem 11 [6] by hand.

b) Jian-Jia's Effort: On October 11th, in parallel to Michael's effort, Jian-Jia used Anthropic Claude (Model Opus 5.5) to clarify some concerns he had. The conversation started after Jian-Jia asked Claude to read [4] and [6]:

> CLAUDE: The paper [4] was not referred to in [6] and the counterexample in [4] was not considered in [6].
>
> JIAN-JIA: Check whether the counterexample used in [4] invalidates Theorem 11 in [6].
>
> CLAUDE: Not directly. ISA rejects this task set, for two independent reasons. (.....) Its blocking term is very pessimistic, so a counterexample would likely need low per-task utilization, many tasks, or D > T to exploit the substitution gap. I can run a computational search for one: random small task sets, ISA verdict versus a simulated worst-case EDF schedule with adversarial suspension placement.

Hence, the AI tool determined that the previous counterexample [4] is not applicable and set up a search on its own. A counterexample was found immediately after this attempt (although different from the first suggestion) and validated by Jian-Jia. This session took roughly 30 minutes of computation and 10 minutes of summarizing the results.

c) Validating Process of the Authors: On October 12th, the counterexamples of Theorem 11 in [6] were shared to

all the authors of this article and the validating process of the counterexample started. As the sufficient condition of Theorem 11 is quite straightforward, the validation process is completed very quickly.

B. Necessary Condition

In the meantime, Michael and Jian-Jia continued to manually search for a counterexample of the necessary condition in [6], based on the aforementioned interactions. The necessary condition states that if a task set is feasibly scheduled by preemptive EDF, then for every $\tau_i \in \mathbb{T} = \{\tau_1, \ldots, \tau_n\}$, it holds that $\frac{S_i}{T_i} + \sum_{\tau_j \in \mathbb{T}} \frac{C_j}{T_j} \leq 1$. This seemingly simple extension of classic preemptive EDF scheduling by adding $\frac{S_i}{T_i}$ for each task was stated in [6] without any proof. Upon close examination Michael and Jian-Jia have manually found that it is possible to define a harmonic implicit-deadline task set with two tasks where one task τ_1 purposely violates the condition whereas the other task τ_2 maximizes the slack in which τ_1 can execute and suspend itself. This renders the task set to be schedulable although it fails the condition.

C. Key Insights

To summarize, although Michael and Jian-Jia have followed different ways to prompt the AI tool, that is, an open-ended question and a direct question building on previous work [4], both sessions quickly produced a counterexample to Theorem 11. The key insight from both interactions is that confirming an initial concern on the correctness can be considerably accelerated. In the past, reproducing the analyses and implementing an execution pattern generator would have required considerable effort. This effort was only justified when there was a strong evidence that an analysis was incorrect. Our case shows how a few interactions with an AI tool can be sufficient to turn an initial concern into a candidate counterexample in less than one hour. Moreover, we note that in a further session by Jian-Jia on October 16th using Proofreader [7] assisted by Anthropic Claude (Model Opus 5.5), the AI tool did not apply the correct rule of self-suspension at the beginning, which shows that the validation by the authors remains essential.

III. Remark

It is amazing how effective AI tools can assist the authors to test their concerns with a few prompts to guide the AI tools to search a particular area of interest. Many counterexamples presented in [1] took much longer to be assessed and verified.

Of course, the question is whether such counterexamples could be found easily if the authors would compose the counterexample search script themselves. We believe that the answer is “yes”. Since the paper [6] assumes discrete time, it is not very difficult to compose a script to enumerate many execution patterns. The search script composed by Anthropic Claude has less than 200 lines of Python code and follows a simple brute-force approach. Nevertheless, as discussed in Section II-C, writing such a script by hand is only justified when there is strong evidence on the incorrectness rather than an initial concern as in our case. The authors believe that there is no reason not to utilize AI tools to assist researchers for this purpose.

Furthermore, it is also worth noting that AI tools can help to avoid brute-force trial and error (generating random task sets), especially when insights of potential counterexamples are specified. With the insight provided by the authors into the prompts and validation process, the main purpose of using AI tools is to quickly validate whether the concerns are valid or not.

The resulting counterexamples in [3] only cover ISA and were hand-crafted for readability. At the time of writing this article, the authors have confirmed that the jitter-based response time analysis (JRTA) in Theorem 6 in [6] for preemptive EDF scheduling for arbitrary-deadline sporadic real-time tasks is also incorrect. The process is more involved even with the assistance of the AI tools, as it requires the aforementioned careful reproduction and validation of Algorithm 1 [6].

References

[1] J. Chen, G. Nelissen, W. Huang, M. Yang, B. B. Brandenburg, K. Bletsas, C. Liu, P. Richard, F. Ridouard, N. C. Audsley, R. Rajkumar, D. de Niz, and G. von der Brüggen. Many suspensions, many problems: a review of self-suspending tasks in real-time systems. *Real Time Syst.*, 55(1):144–207, 2019.

[2] U. C. Devi. An improved schedulability test for uniprocessor periodic task systems. In *15th Euromicro Conference on Real-Time Systems (ECRTS)*, pages 23–32, 2003.

[3] M. Frichert, M. Günzel, M. Sudvarg, and J.-J. Chen. Comments paper: Counterexample for schedulability analysis for self-suspending tasks under edf-like scheduling, 2026.

[4] M. Günzel and J. Chen. Correspondence article: Counterexample for suspension-aware schedulability analysis of EDF scheduling. *Real Time Syst.*, 56(4):490–493, 2020.

[5] M. Günzel, G. von der Brüggen, K. Chen, and J.-J. Chen. EDF-Like scheduling for self-suspending real-time tasks. In *RTSS*, pages 172–184. IEEE, 2022.

[6] Y. Wang, B. Lv, Q. Zhou, J. Li, and T. Tan. Schedulability analysis for self-suspending tasks under EDF-Like scheduling. *IEEE Trans. Computers*, 74(7):2364–2375, 2025.

[7] B. C. Ward. Proofreader: Informally auditing formal proofs with LLMs, 2026.