Adaptive Execution for Real-Time Observations of Astrophysical Transients

Marion Sudvarg*, Ye Htet†, Sanjoy Baruah†, Jeremy Buhler†, Roger Chamberlain†, Chris Gill†, Jim Buckley*
*Department of Physics †Department of Computer Science and Engineering
Washington University in St. Louis
{msudvarg, htet.ye, baruah, jbuhler, roger, cdgill, buckley}@wustl.edu

I. Background and Motivation

The National Academies of Sciences, Engineering, and Medicine, in collaboration with other scientific organizations of the United States Government (NASA, NSF, DOE), released the Astro2020 decadal survey [1] to identify scientific challenges for astronomy and astrophysics in the next decade. The survey highlighted the "space-based time-domain and multi-messenger program" as the highest-priority sustaining activity in space. The goal of this program is to coordinate concurrent observations of astrophysical phenomena by instruments with different observational modalities — e.g., gravity waves, neutrinos, cosmic rays, and across the electromagnetic spectrum from radio to gamma rays.

A key component of the broader multi-messenger program is the coordination of real-time follow-up observations of transient astrophysical phenomena —e.g., gamma-ray bursts (GRBs) and fast-radio bursts (FRBs)— by optical instruments. The idea is to detect and localize transients in real time using observational data from non-optical, wide field-of-view (FoV) instruments, thus enabling narrow-FoV optical telescopes to turn and look. The goal is to capture early stages of the transients' development in a broader electromagnetic band. This gives rise to several challenges, depending on the type of phenomenon. For example, GRBs are typically detected by space-based instruments. Due to their high energies, gamma-ray photons predominantly display particle-like (not wave-like) behavior, making them difficult to lens using traditional optical instruments. GRB detectors are therefore non-optical, and so burst localization requires data- and compute-intensive processing of instrument readouts from a large number of gamma-ray scatters. Due to slow intermittent links to the satellite telescopes, it may take hours before ground-based computers complete localization.

As an alternative, we have proposed —and demonstrated the feasibility of— localizing bright GRBs with **subdegree accuracy** in **under 200 ms** using embedded computational hardware **aboard the instrument in space** [2]. Recent advances in fast-slewing robotic optical telescopes, such as the TURBO telescopes under development by the Univ. of MN [3], suggest that sub-second optical follow-up observations may be possible in the near future, especially if a satellite mission like our proposed Advanced Particle-astrophysics Telescope (APT) instrument receives funding. For this year's RTSOPS, we identify open problems in real-time systems that must be addressed to realize the necessary coordination of localization/follow-up.

II. Problem Overview

We consider a distributed system that couples a wide-FoV non-optical detector with one or more geographically dispersed narrow-FoV optical follow-up instruments. If a transient event (e.g., a GRB) is detected, the goal is to localize the event as quickly and accurately as possible, then direct where the optical telescopes should point to make observations as soon as possible after detection. However, as we have shown in prior work [4], there is a fundamental tradeoff between *fast* and *accurate* localization. We therefore state the problem as follows:

The system should aim to *minimize* the expected latency from space-based *detection* to *secondary observations* of astrophysical transients.

III. Fast Localization of Transients

A. Related Work

To minimize observational latency, we need to be able to accurately localize transients as quickly as possible. The real-time systems community tends to focus on *predictability* of timing guarantees, rather than making a system as fast as possible. Nonetheless, for our application, *fast* execution is a prerequisite of *timely* execution, and speeding up localization on embedded hardware (such as what might fly aboard a satellite) presents multiple challenges.

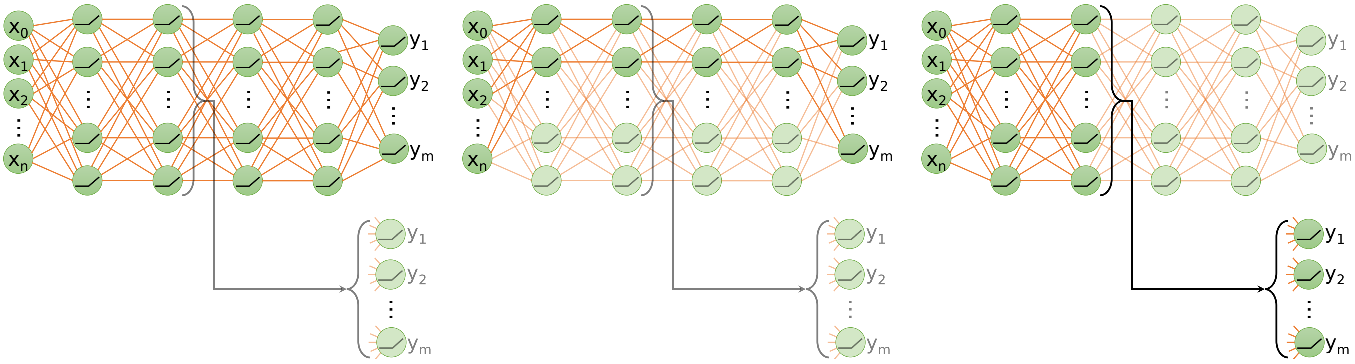
We have written about several of these challenges previously. In [2] and [5], we re-implemented the statistical gamma-ray photon trajectory reconstruction algorithm in [6] as a branch-and-bound tree search, which dynamically prunes the search space online. In the same papers, we presented novel parallel algorithms for efficient multilateration of multiple reconstructed trajectories under noisy conditions. In simulation, we demonstrated that these approaches enable accurate localization within 200 ms on a Raspberry Pi 3B+ even for bright bursts that generate large quantities of data. In [7], we explored acceleration of these algorithms on an NVIDIA Jetson, which couples a CPU and GPU on a single embedded system on chip (SoC). The considered algorithms assume that sensor data is heavily pre-processed and reduced prior to arrival at the CPU. In [8] and [9], we presented initial FPGA-based implementations of these preprocessing algorithms using high-level synthesis (HLS).

B. Integration of Deep Learning

With the growing success and richness of domain-specific AI/ML tools and techniques, there has been increasing interest in the astrophysics community [10] (and in the broader academic community) to replace traditional likelihood-based methods with neural networks. Deep learning can detect gravitational wave traces from binary neutron star mergers in large datasets [11]; for example, a month of LIGO data can be processed within 50 seconds [12]. The AGILE X-ray/gamma-ray satellite uses an anomaly detection autoencoder convolutional neural network (CNN) to detect GRBs in real-time using data from its anticoincident system [13]. These approaches are deployed on powerful ground-based computing clusters.

Preliminary results suggest that our own GRB localization pipeline may benefit from ML-based components, e.g., to reject background radiation, to predict a gamma ray's trajectory through the detector, to quantify the expected error in that prediction, or to perform the final multilateration step for localization [14]. To deploy these components on size, weight, and power (SWaP)-constrained platforms aboard a satellite like APT may require model compression to meet power and latency constraints. Techniques such as pruning and quantization [15], [16], early exit [17], [18], layer skipping [19], and slimming [20] of neural networks have been effective in reducing inference latency; such techniques are surveyed in [21]. Achieving the desired performance requires careful consideration of the underlying hardware, but deployments of pruning and quantization to GPUs [22] and FPGAs [23] have proven effective.

However, to our knowledge, deployment of slimming and early exit *together* on an FPGA remains an open problem. Intuitively, both slimming and early exit dynamically disable parts of the neural network, which lowers execution time (and therefore energy use) on a CPU-based deployment. Slimming does so by disabling nodes within layers to reduce the *width* of the network, whereas early exit disables entire layers to reduce the *length* of the network, as illustrated in Fig. 1. Traditionally, early exiting is realized by dynamically choosing the depth based on each input; "easier" inputs (i.e., those deemed easier to classify) use outputs at earlier stages of the network. This therefore decreases *average-case* execution time, but does not guarantee better performance (i.e., shorter latencies) in the *worst-case*. However, recent efforts have successfully produced networks that can adapt their width and depth under CPU or GPU resource and execution time constraints [24].



(a) Nominal network. (b) With slimming. (c) With early exit.

Fig. 1: Neural network model compression techniques.

Given the high degree of parallelism provided by an FPGA, online slimming logic (adaptive width) could be synthesized to reduce power draw, with deterministic early exits (adaptive depth) used to reduce latency. What is needed, therefore, is *(i)* a model that captures the accuracy, energy, and latency tradeoffs to dynamically select the best slimming and early exit configuration during runtime, and *(ii)* an FPGA implementation that can enforce this. For the latter, HLS may be capable of specifying nodes and layers as logical blocks with data paths between them enabled/disabled via dedicated signals. Investigation into these techniques is ongoing.

IV. Speed and Accuracy Tradeoffs for Follow-Up Observations

The aforementioned concerns with adapting neural network designs are part of a speed vs. accuracy tradeoff. Though we want both *speed* and *accuracy*, one may need to be sacrificed for the other. In prior work, we framed the problem as one of constrained optimization: accuracy should be maximized while guaranteeing completion within some deadline [4], [25]. The deadline may be dynamic (i.e., unique to each GRB localization job), but it is nonetheless firm.

However, this is only a limited expression of the problem we want to solve. As stated previously, we aim to minimize the expected latency from detection to follow-up observation. This latency has three components: *(i)* the time to perform localization, *(ii)* the time to transmit the source direction to a secondary telescope, and *(iii)* the time to physically search for the burst within the localized region. As we may not have control over *(ii)*, we consider instead the tradeoff between *(i)* and *(iii)*. Fundamentally, the challenge is that localization is imprecise, so that the inferred direction has some associated error. We instead consider localization as the process of constraining the source to a given *region* of the sky, rather than an exact direction. This region is likely to be larger than the FoV of the secondary optical telescope but can be searched by physically moving the telescope. This idea is illustrated in Fig. 2. This gives rise to the following optimization problem:

$$\textbf{minimize} \quad t(\mathcal{L}) + \frac{A(\mathcal{L})}{2s} \tag{1}$$

Here, $\mathcal{L}$ defines the computational mode selected for localization, $t(\mathcal{L})$ is the time to execute that mode, $A(\mathcal{L})$ is the area of the region that we expect to be constrained, and s is the speed at which the optical telescope can search the region. Note that we use $2s$ in the denominator to reflect that, in expectation, half the region must be searched before the source is found. Intuitively, we want to tune the localization algorithm so that it narrows the constrained search space faster than the optical instrument alone can. Our prior work in [4] demonstrated a technique to quantify how quickly localization error is reduced with additional execution time, but it remains an open problem to instead quantify the effect on the search space.

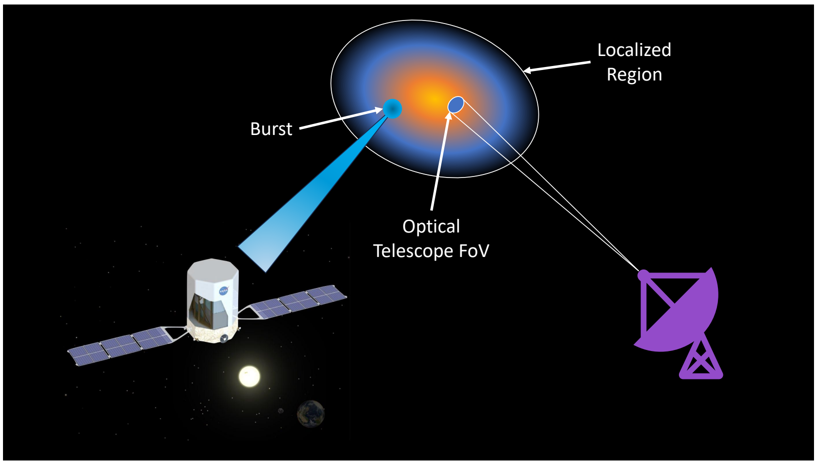


Fig. 2: The localized source region may be larger than the field of view (FoV) of the secondary optical telescope.

Moreover, in reality the predicted search region is not exact but is instead only bounded with some confidence. Therefore, a stochastic model that considers a probability density function describing the direction also may be needed. Then, both closed-form analytical solutions and approaches based in e.g., Markov Decision Process (MDP) models, combined with e.g., Monte Carlo simulations, and the development of advanced techniques to solve them such as recurrent neural networks, can be used to define sufficiently efficient procedures for online use that are optimal in expectation.

V. Elastic Job Scheduling

When adapting execution (e.g., of neural network inference or the localization algorithm) in the face of resource and latency constraints, we must consider the system as a whole: jobs do not run in isolation, but must be scheduled concurrently. For example, the proposed APT instrument will be composed of multiple detectors; a single transient may be sensed by more than one detector, or multiple transients (e.g., a GRB and a single high-energy cosmic ray) might overlap in the time domain. Moreover, instrument control jobs, e.g. that respond to power or thermal events, may arrive in sporadic or aperiodic fashion.

Buttazzo's elastic scheduling model for real-time tasks [26] provides a framework to adapt task utilizations to guarantee schedulability in the face of overload. Each task is parameterized by an "elastic" constant representing its relative adaptability (e.g., based on its importance). If the system is overloaded, task utilizations are reduced ("compressed") from their desired values proportionally to these elastic parameters until the system becomes schedulable; this may be realized by extending the periods of tasks, or decreasing their execution time budgets. Each task may also be assigned a minimum utilization below which it cannot be compressed. Though elastic scheduling has been extended to several recurrent task models [27], [28], [29], [30], [31], [32], [33], it has *not* yet been extended to independent jobs.

As one possible approach, we propose the following model. Consider a sequence of jobs $J_i = (C_i^{\min}, C_i^{\max}, A_i, D_i, E_i)$. $C_i^{\max}$ represents the job's desired execution time budget given no resource constraints, while $C_i^{\min}$ represents the extent to which it can adapt by decreasing its execution time. The job arrives at time A_i and must complete by its deadline D_i. When a job arrives, one must determine whether all current jobs are still schedulable; if not, their workloads are compressed proportionally to their elastic constants E_i. For EDF scheduling, the problem may be stated as follows. When a job arrives at time t_o, we consider the current set of active jobs J_i (indexed in deadline order). We aim to:

$$\textbf{minimize} \quad \lambda \tag{2a}$$
$$\text{s.t.} \quad C_i + \Delta_i \geq C_i^{\max} - \lambda E_i \tag{2b}$$
$$C_i + \Delta_i \geq C_i^{\min} \tag{2c}$$
$$\sum_{j \leq i} C_j \leq D_i - t_o \tag{2d}$$

Here, Δ_i represents the amount of time that job J_i has already executed. λ represents the "amount" of compression applied to the system. This is reflected by the first constraint, which says that the execution time of each job is compressed from its maximum value proportionally to its elastic constant. The second constraint prevents a task's execution time budget from being compressed below its minimum value. The third constraint guarantees schedulability: the remaining execution time for a given job, plus those jobs with higher priorities, must be less than the time remaining before its deadline.

Two problems arise when considering such a model. *(i)* How can this be solved efficiently during online job arrival? Since this is similar to the problem of utilization compression, the problem can be solved in linear time on the number of jobs using our prior technique in [34] for each constraint of the form in (2d). Since there is one such constraint for each job, this requires quadratic time complexity, though a faster algorithm might be possible. Also *(ii)*, what is the competitive ratio of this online algorithm compared to a clairvoyant algorithm that knows the future sequence of job arrivals? In other words, for a given sequence of jobs, how does the maximum compression λ given by the online algorithm compare to that which is necessary if all future job arrivals are known a priori?

Acknowledgment

Grants from NASA (80NSSC21K1741), NSF (CNS-2141256, CNS-2229290), and WUSTL OVCR (seed) support this work.

References

[1] National Academies of Sciences Engineering and Medicine, *Pathways to Discovery in Astronomy and Astrophysics for the 2020s.* Washington, DC, USA: The National Academies Press, 2023. [Online]. Available: https://nap.nationalacademies.org/catalog/26141/pathways-to-discovery-in-astronomy-and-astrophysics-for-the-2020s

[2] M. Sudvarg, J. Buhler, J. H. Buckley, W. Chen *et al.*, "A Fast GRB Source Localization Pipeline for the Advanced Particle-astrophysics Telescope," in *Proc. of 37th Int'l Cosmic Ray Conference — PoS(ICRC2021)*, vol. 395, Jul. 2021, pp. 588:1–588:9.

[3] U. of Minnesota, "U of M leading $1 million grant to build superfast 'TURBO' telescopes," University of Minnesota, March 2024. [Online]. Available: https://twin-cities.umn.edu/news-events/u-m-leading-1-million-grant-build-superfast-turbo-telescopes

[4] M. Sudvarg, J. Buhler, R. D. Chamberlain, C. Gill, J. Buckley, and W. Chen, "Parameterized workload adaptation for fork-join tasks with dynamic workloads and deadlines," in *Proc. of IEEE 29th Int'l Conf. on Embedded and Real-Time Computing Systems and Applications*, Aug. 2023.

[5] Y. Htet, M. Sudvarg *et al.*, "Prompt and Accurate GRB Source Localization Aboard the Advanced Particle Astrophysics Telescope (APT) and its Antarctic Demonstrator (ADAPT)," in *Proc. of 38th Int'l Cosmic Ray Conference*, vol. 444, Jul. 2023, pp. 956:1–956:9.

[6] S. Boggs and P. Jean, "Event reconstruction in high resolution Compton telescopes," *Astronomy and Astrophys. Supp. Series*, vol. 145, no. 2, pp. 311–321, 2000.

[7] J. Wheelock, W. Kanu, M. Sudvarg *et al.*, "Supporting Multi-messenger Astrophysics with Fast Gamma-ray Burst Localization," in *Proc. of IEEE/ACM HPC for Urgent Decision Making Workshop (UrgentHPC)*, Nov. 2021.

[8] M. Sudvarg *et al.*, "Front-End Computational Modeling and Design for the Antarctic Demonstrator for the Advanced Particle-astrophysics Telescope," in *Proc. of 38th Int'l Cosmic Ray Conference*, vol. 444, Jul. 2023, pp. 764:1–764:9.

[9] M. Sudvarg, C. Zhao, Y. Htet, M. Konst, T. Lang, N. Song, R. D. Chamberlain, J. Buhler, and J. H. Buckley, "HLS taking flight: Toward using high-level synthesis techniques in a space-borne instrument," in *Proc. of 21st International Conference on Computing Frontiers.* ACM, 2024.

[10] E. A. Huerta *et al.*, "Enabling real-time multi-messenger astrophysics discoveries with deep learning," *Nature Reviews Physics*, vol. 1, no. 10, pp. 600–608, Oct 2019. [Online]. Available: https://doi.org/10.1038/s42254-019-0097-4

[11] R. Qiu, P. G. Krastev, K. Gill, and E. Berger, "Deep learning detection and classification of gravitational waves from neutron star-black hole mergers," *Physics Letters B*, vol. 840, p. 137850, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0370269323001843

[12] P. Chaturvedi, A. Khan, M. Tian, E. A. Huerta, and H. Zheng, "Inference-optimized AI and high performance computing for gravitational wave detection at scale," *Frontiers in Artificial Intelligence*, vol. 5, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frai.2022.828672

[13] N. Parmiggiani, A. Bulgarelli, A. Ursi, A. Macaluso, A. D. Piano, V. Fioretti, A. Aboudan, L. Baroncelli, A. Addis, M. Tavani, and C. Pittori, "A deep-learning anomaly-detection method to identify gamma-ray bursts in the ratemeters of the agile anticoincidence system," *The Astrophysical Journal*, vol. 945, no. 2, p. 106, Mar. 2023.

[14] Y. Htet, M. Sudvarg, J. Buhler, R. Chamberlain, J. Buckley, and the APT Collaboration, "A Computational Pipeline for Prompt Gamma-Ray Burst Localization Aboard APT and ADAPT," in *21st Meeting of the High Energy Astrophysics Division.* American Astronomical Society, Apr. 2024.

[15] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2704–2713.

[16] A. Kuzmin, M. Nagel, M. van Baalen, A. Behboodi, and T. Blankevoort, "Pruning vs quantization: Which is better?" in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 62 414–62 427. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/c48bc80aa5d3cbbdd712d1cc107b8319-Paper-Conference.pdf

[17] S. Teerapittayanon, B. McDanel, and H.-T. Kung, "Branchynet: Fast inference via early exiting from deep neural networks," in *Proc. of 23rd International Conference on Pattern Recognition (ICPR).* IEEE, 2016, pp. 2464–2469.

[18] T. Bolukbasi, J. Wang, O. Dekel, and V. Saligrama, "Adaptive neural networks for fast test-time prediction," *CoRR*, vol. abs/1702.07811, 2017. [Online]. Available: http://arxiv.org/abs/1702.07811

[19] X. Wang, F. Yu, Z. Dou, and J. E. Gonzalez, "Skipnet: Learning dynamic routing in convolutional networks," *CoRR*, vol. abs/1711.09485, 2017. [Online]. Available: http://arxiv.org/abs/1711.09485

[20] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proc. of IEEE International Conference on Computer Vision*, 2017, pp. 2736–2744.

[21] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, "Dynamic neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7436–7456, 2022.

[22] T. Zhang, S. Ye, X. Feng, X. Ma, K. Zhang, Z. Li, J. Tang, S. Liu, X. Lin, Y. Liu, M. Fardad, and Y. Wang, "StructADMM: Achieving ultrahigh efficiency in structured pruning for DNNs," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 5, pp. 2259–2273, 2022.

[23] G. B. Hacene, V. Gripon, M. Arzel, N. Farrugia, and Y. Bengio, "Quantized guided pruning for efficient hardware implementations of deep neural networks," in *Proc. of 18th IEEE International New Circuits and Systems Conference (NEWCAS)*, 2020, pp. 206–209.

[24] L. Hou, Z. Huang, L. Shang, X. Jiang, X. Chen, and Q. Liu, "Dynabert: Dynamic bert with adaptive width and depth," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9782–9793, 2020.

[25] M. Sudvarg, J. Buhler, R. Chamberlain, C. Gill, and J. Buckley, "Work in Progress: Real-Time GRB Localization for the Advanced Particle-astrophysics Telescope," in *Proc. of 15th Workshop on Operating Systems Platforms for Embedded Real-Time Applications (OSPERT)*, Jul. 2022, pp. 57–61.

[26] G. C. Buttazzo, G. Lipari, and L. Abeni, "Elastic Task Model for Adaptive Rate Control," in *IEEE Real-Time Systems Symposium*, 1998.

[27] T. Chantem, X. S. Hu, and M. D. Lemmon, "Generalized Elastic Scheduling," in *IEEE International Real-Time Systems Symposium*, 2006.

[28] J. Orr, C. Gill, K. Agrawal, S. Baruah *et al.*, "Elasticity of workloads and periods of parallel real-time tasks," in *Proc. of 26th International Conference on Real-Time Networks and Systems.* ACM, 2018, pp. 61–71. [Online]. Available: https://doi.org/10.1145/3273905.3273915

[29] J. Orr and S. Baruah, "Multiprocessor scheduling of elastic tasks," in *Proc. of 27th International Conference on Real-Time Networks and Systems.* ACM, 2019, pp. 133–142. [Online]. Available: https://doi.org/10.1145/3356401.3356403

[30] J. Orr, J. C. Uribe, C. Gill, S. Baruah *et al.*, "Elastic scheduling of parallel real-time tasks with discrete utilizations," in *Proc. of 28th International Conference on Real-Time Networks and Systems.* ACM, 2020, pp. 117–127. [Online]. Available: https://doi.org/10.1145/3394810.3394824

[31] S. Baruah, "Improved uniprocessor scheduling of systems of sporadic constrained-deadline elastic tasks," in *Proceedings of the 31st International Conference on Real-Time Networks and Systems (RTNS 2023).* New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3575757.3575759

[32] M. Sudvarg, S. Baruah, and C. Gill, "Elastic Scheduling for Fixed-Priority Constrained-Deadline Tasks," in *2023 IEEE 26th International Symposium on Real-Time Distributed Computing (ISORC)*, 2023, pp. 11–20.

[33] M. Sudvarg, A. Li, D. Wang, S. Baruah, J. Buhler, C. Gill, N. Zhang, and P. Ekberg, "Elastic Scheduling for Harmonic Task Systems," in *2024 Real-Time and Embedded Technology and Applications Symposium (RTAS).* IEEE, 2024.

[34] M. Sudvarg, C. Gill, and S. Baruah, "Linear-time admission control for elastic scheduling," *Real-Time Systems*, vol. 57, no. 4, pp. 485–490, 10 2021. [Online]. Available: https://doi.org/10.1007/s11241-021-09373-4