Real-Time Systems Optimization with Black-box Constraints and Hybrid Variables

Sen Wang[1], Dong Li[1], Shao-Yu Huang[2], Xuanliang Deng[1], Ashrarul H. Sifat[1],
Changhee Jung[2], Ryan Williams[1], Haibo Zeng[1]
[1]*Virginia Tech,* [2]*Purdue University*

***Abstract*—When optimizing real-time systems, designers often face a challenging problem where the schedulability constraints are non-convex, non-continuous, or lack an analytical form to understand their properties. Although the optimization framework NORTH proposed in [1] is general (it works with arbitrary schedulability analysis) and scalable, it can only handle problems with continuous variables, which limits its application. In this paper, we extend the applications of the framework NORTH to problems with a hybrid of continuous and discrete variables. This is achieved in a coordinate-descent method, where the continuous and discrete variables are optimized separately during iterations. The new framework, NORTH+, improves around 20% solution quality than NORTH in experiments.**

I. Introduction

Over the years, the real-time systems community has developed an impressive list of scheduling algorithms and schedulability analysis techniques. However, many schedulability analyses are not friendly for optimization purposes. Some analysis methods are complicated and do not provide closed-form expressions, such as those based on demand bound functions [2], real-time calculus [3], abstract event model interfaces [4], or timed automata [5], [6].

Another challenge in optimizing real-time systems is their ever-increasing complexity. The functionality of real-time systems such as automotive and aircraft is growing fast, especially given the recent trend of introducing autonomous features and system-to-system connectivity [7], [8]. The existing optimization algorithms do not scale to the hundreds of software tasks that modern automotive systems contain [9].

The existing approaches for optimizing real-time systems do not adequately address the above two challenges as they lack applicability or scalability. Besides, many standard mathematical optimization frameworks cannot efficiently handle complicated schedulability analysis constraints. Therefore, in recent work [1], Wang *et al.* introduced *NORTH* (Numerical Optimizer with Real-Time Highlight), an optimization framework for optimizing systems with black-box schedulability constraints based on numerical optimization algorithms. However, the application of NORTH is limited to systems with only continuous variables.

In this paper, we extend NORTH [1] to handle problems with *hybrid* (i.e., a mixture of continuous and discrete) variables. The new framework, named NORTH+, achieves this extension based on an iterative algorithm. During the iterations, the continuous and discrete variables are optimized independently. The continuous optimization component remains the same as NORTH, while the discrete optimization can leverage heuristics to ensure algorithm scalability. Simulated experiments indicate that NORTH+ brings around 20% performance improvements compared with NORTH [1].

II. Problem description

Generally speaking, real-time systems design can be mathematically described as an optimization problem:

$$\min_{\mathbf{x}} \quad \mathcal{F}(\mathbf{x}) \tag{1}$$

$$\text{subject to} \quad \text{Sched}(\mathbf{x}) = 0 \tag{2}$$

where the vector of design variables $\mathbf{x} = \{\mathbf{x}^{\mathcal{C}}, \mathbf{x}^{\mathcal{D}}\}$ includes continuous variables $\mathbf{x}^{\mathcal{C}}$ and discrete variables $\mathbf{x}^{\mathcal{D}}$. It represents the design choices, such as the task execution times, periods, and priority assignments. $\mathcal{F}(\mathbf{x})$ maps the design variables $\mathbf{x}$ into the design objective (a scalar). In addition, real-time system design is subject to the schedulability analysis constraint:

$$\text{Sched}(\mathbf{x}) = \begin{cases} 0, & \text{system is schedulable} \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

Here, we treat the schedulability analysis as a black-box that only returns binary results, i.e., schedulable or not.

III. Methdology

The overview of NORTH+ is shown in Fig. 1.

A. Performing continuous optimization

We briefly review [1] in this section, which forms the foundation of this paper. The framework of [1] is shown in Fig. 2, which has two major steps during iterations:

- Numerical Method-Based Optimization (NMBO): an optimization method based on trust-region optimizers [10]. Given an initial solution $\mathbf{x}$, the algorithm first performs unconstrained optimization to find a descent direction $\mathbf{\Delta}$ and then adjusts $\mathbf{\Delta}$ to keep $\mathbf{x} + \mathbf{\Delta}$ feasible. It terminates when $\|\mathbf{\Delta}\|_2$ becomes smaller than a given threshold.
- Variable Elimination (VE): After NMBO terminates, we try to find a feasible descent direction (i.e., we can take a big step along this direction without violating the schedulability constraints) by searching within the sub-space of the $\mathbf{\Delta}$ above. When $\mathbf{x} + \mathbf{\Delta}$ violates the schedulability constraints, it is often the case that only a small subset of tasks miss their deadline. Therefore, excluding these tasks from further optimization frees the numerical optimizers from schedulability constraints and improves performance in future iterations.

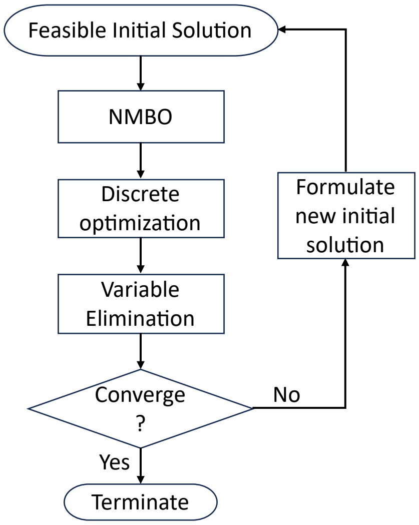


Figure 1: NORTH+ is an extension of the NORTH optimization framework [1]. The NMBO and VE components are shown in Fig. 2 and explained more in Section III-A, the discrete optimization is introduced in Section III-B. The continuous and discrete variables in NORTH+ are optimized separately: when optimizing continuous variables, the discrete variables are treated as constants, and vice versa.

- Formulate new initial solution: Eliminated variables will be considered constant in subsequent iterations, and a new initial solution will be formulated based solely on the remaining variables.

B. *Performing discrete optimization*

Although the primary goal of NORTH+ is optimizing the objective function (1), enhancing system schedulability can be regarded as a secondary goal. When NORTH optimizes the continuous variables, it tends to push them towards the boundaries of schedulable regions. An example is shown in Fig. 3. Thus, some variables are often close to violating the schedulability constraints when the numerical optimizer terminates. This situation prevents further optimization opportunities.

Therefore, we can broaden the system's schedulability region to improve performance. For example, when the discrete variables are priority assignments, many classical algorithms can be utilized to assign priorities, such as Rate Monotonic (RM) [11], [12] or DkC [13]. These algorithms should be executed within each iteration, as illustrated in Fig. 1, as optimal priority assignments may change as the continuous variables are optimized.

IV. Experiments

In the experiments, we consider an example application where the objective function is derived from control performance optimization [14]:

$$\min_{\mathbf{T},\mathbf{P}} \sum_{i=0}^{N-1} (\alpha_i \mathbf{T}_i + \beta_i r_i(\mathbf{T},\mathbf{P})) \tag{4}$$

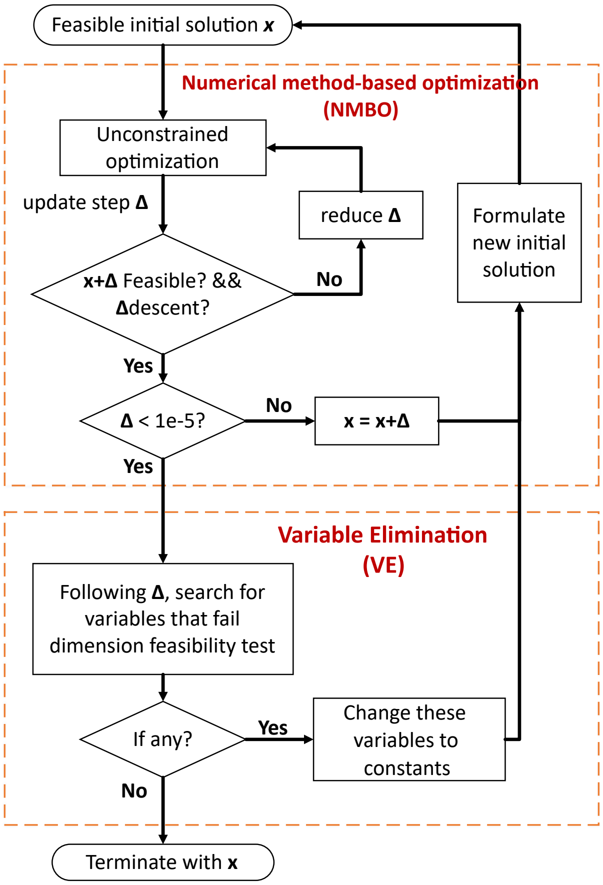

Feasible initial solution x
Numerical method-based optimization (NMBO)
Unconstrained optimization
update step Δ
reduce Δ
x+Δ Feasible? && Δdescent?
No
Yes
Δ < 1e-5?
No
x = x+Δ
Yes
Formulate new initial solution
Variable Elimination (VE)
Following Δ, search for variables that fail dimension feasibility test
If any?
Yes
Change these variables to constants
No
Terminate with x

Figure 2: Optimization framework NORTH from [1] that optimizes only continuous variables.

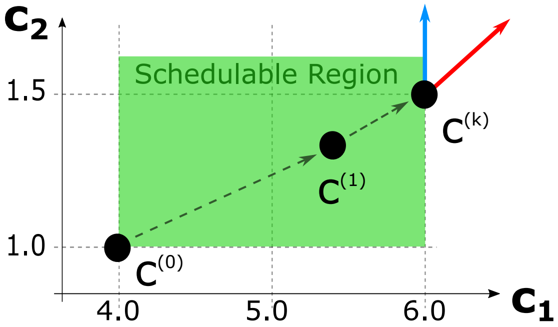
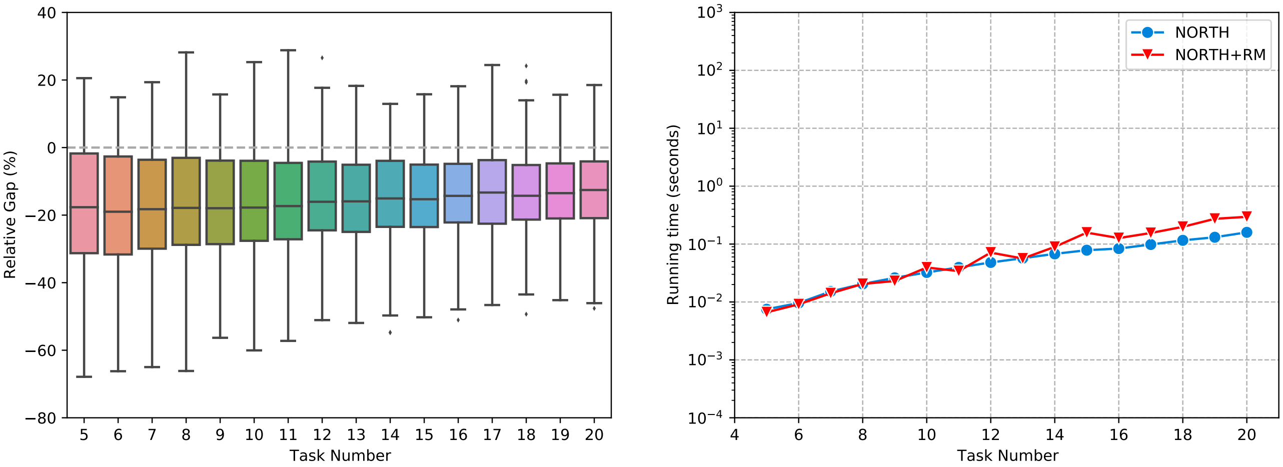


Figure 3: From [1], an example that visualizes how NORTH optimizes the continuous variables $\mathbf{C}$ in the figure. Expanding the schedulable region at $\mathbf{C}^{(k)}$ gives NORTH more chances of optimization.

where the variables $\mathbf{T}_i$ are the task τ_i's periods; N dentoes the number of tasks; α_i and β_i are control weight parameters; $\mathbf{P}$ are the priority assignments for all the tasks; $r_i(\mathbf{T},\mathbf{P})$ is the task τ_i's response time.

We evaluate the framework on simulated task sets. Ten thousand task sets were generated randomly following [1], [14]. The task sets consisted of periodic tasks $\boldsymbol{\tau}$ with implicit deadlines scheduled by fixed-task priority schedulers on a uni-



(a) Relative performance gap

(b) Run-time speed

Figure 4: Performance comparison between NORTH and NORTH+RM.

processor platform. The worst-case execution time C_i of each task τ_i was generated randomly from [1, 100]; the task periods were upper-bounded by $T_i \leq 5\sum_{i=1}^{n} C_i$. In the objective function (4), α_i was randomly generated in the range [1, 1000]; β_i was generated in the range [1, 10000].

In the experiments, the black-box schedulability analysis constraint follows the classical response time analysis [15]:

$$r_i = C_i + \sum_{j \in \text{hp}(i)} \lceil \frac{r_i}{T_j} \rceil C_j \tag{5}$$

where $\text{hp}(i)$ denotes the tasks with higher priority than the task τ_i. Let D_i denote the deadline of the task τ_i, we have:

$$\text{Sched}(\mathbf{T}, \mathbf{P}) = \begin{cases} 0, & \forall i, r_i(\mathbf{T}, \mathbf{P}) \leq D_i \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

We compared the performance between NORTH [1] and NORTH+RM because they work with black-box schedulability constraints. In NORTH+RM, task priorities are adjusted with RM each time after NORTH optimizes the period variables, as illustrated in Fig. 1. Fig. 4 shows experiment results, where the relative gap is defined as follows:

$$\frac{\mathcal{F}_{\text{NORTH+RM}} - \mathcal{F}_{\text{NORTH}}}{\mathcal{F}_{\text{NORTH}}} \times 100\% \tag{7}$$

V. Discussion, Open Challenges and Future Work

Experiments show that adding an extra discrete optimization step to NORTH improves applicability and performance. Although finding the optimal solutions for the discrete optimization step in NORTH+ could be difficult, simple heuristics may work reasonably well in practice. In simulated experiments, assigning priorities based on RM brought around 20% performance improvements.

A. *Challenge: Optimality vs Applicability and Efficiency*

Fig. 4a shows that NORTH+RM does not consistently outperform NORTH. NORTH+ is not designed to provide optimal solutions. Given a real-time system optimization problem with possibly black-box constraints and many variables, NORTH+ can provide a solution quickly with reasonably good quality, albeit without a guarantee of optimality.

Providing an optimality guarantee for problems with black-box constraints is challenging. These black-box constraints could be non-convex and non-continuous, and cause the optimal solutions, denoted as $\mathbf{x}^*$, to potentially exist anywhere. In worst-case scenarios, many solution candidates we searched for may be infeasible or non-optimal until we find $\mathbf{x}^*$, regardless of the optimization algorithms. In this case, enumerating all the possible solution candidates may be necessary to provide the optimality guarantee. This detrimentally impacts algorithm scalability.

If special properties of the black-box constraints are known, such as continuity or convexity, we can efficiently narrow down the search space. Although the problem is becoming less "black-box", domain-specific observation and techniques may be available to improve efficiency and potentially provide an optimality guarantee, such as sustainable schedulability analysis [14], convexity [16], dynamic programming [17]. However, the availability of these domain-specific observations also implies a more limited range of applicability.

Another promising direction that aligns with the black-box assumption (i.e., without reducing applicability) is machine learning (ML) methods [18]–[20]. ML can construct mathematical models to approximate the black-box constraints and infer desirable properties. Once such a model is established, classical optimization algorithms can be employed to find good solutions. However, a significant challenge with these methods is ensuring feasibility, i.e., guaranteeing that the solutions found are always feasible.

B. Future Work in Methdology Generalization

1) Multi-objective optimization: Classical techniques from multi-objective optimization can be utilized with NORTH+ if there are multiple objective functions. These techniques include assigning weight parameters to combine multiple objectives into a scalar value or transforming specific objective functions into constraints. For example, an objective function aiming to minimize response time could be transformed into a constraint that requires the response time to be smaller than a threshold. NORTH+'s general applicability works with both strategies mentioned above.

2) Discrete optimization: In this paper, we find that improving the system schedulability can indirectly improve the overall performance of NORTH. This observation encourages combining discrete optimization methods and schedulability analysis to achieve better performance. We leave this part as the future work.

VI. Acknowledgment

This work is partially supported by NSF Grants No. 1812963 and 1932074.

References

[1] S. Wang, R. K. Williams, and H. Zeng, "A general and scalable method for optimizing real-time systems with continuous variables," *IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 119–132, 2023.

[2] S. Baruah, A. K. Mok, and L. E. Rosier, "Preemptively scheduling hard-real-time sporadic tasks on one processor," *IEEE Real-Time Systems Symposium*, pp. 182–190, 1990.

[3] L. Thiele, S. Chakraborty, and M. Naedele, "Real-time calculus for scheduling hard real-time systems," in *IEEE International Symposium on Circuits and Systems*, pp. 101–104, 2000.

[4] R. Henia, A. Hamann, M. Jersak, R. Racu, K. Richter, and R. Ernst, "System level performance analysis–the symta/s approach," *IEE Proceedings-Computers and Digital Techniques*, vol. 152, no. 2, pp. 148–166, 2005.

[5] K. G. Larsen, P. Pettersson, and W. Yi, "Uppaal in a nutshell," *International Journal on Software Tools for Technology Transfer*, vol. 1, pp. 134–152, 1997.

[6] M. Nasri, G. Nelissen, and B. B. Brandenburg, "Response-time analysis of limited-preemptive parallel dag tasks under global scheduling," in *ECRTS*, 2019.

[7] L. Heintzman, A. Hashimoto, N. Abaid, and R. K. Williams, "Anticipatory planning and dynamic lost person models for Human-Robot search and rescue," in *IEEE International Conference on Robotics and Automation*, pp. 8252–8258, May 2021.

[8] PerceptIn, "2021 rtss industry challenge." http://2021.rtss.org/industry-session/, 2021.

[9] A. H. Simon Kramer, Dirk Ziegenbein, "Real world automotive benchmarks for free," 2015.

[10] Nocedal and J. Wright, *Numerical Optimization, 2nd edition*. Springer New York, NY, 2006.

[11] P. Pillai and K. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," *Proceedings of the eighteenth ACM symposium on Operating systems principles*, 2001.

[12] E. Bini, G. Buttazzo, and G. Lipari, "Minimizing cpu energy in real-time systems with discrete speed management," *ACM Trans. Embed. Comput. Syst.*, vol. 8, pp. 31:1–31:23, 2009.

[13] R. I. Davis and A. Burns, "Priority assignment for global fixed priority pre-emptive scheduling in multiprocessor real-time systems," *2009 30th IEEE Real-Time Systems Symposium*, pp. 398–409, 2009.

[14] Y. Zhao, R. Zhou, and H. Zeng, "An optimization framework for real-time systems with sustainable schedulability analysis," *2020 IEEE Real-Time Systems Symposium (RTSS)*, pp. 333–344, 2020.

[15] M. Joseph and P. K. Pandya, "Finding response times in a real-time system," *Comput. J.*, vol. 29, pp. 390–395, 1986.

[16] H. Aydin, V. Devadas, and D. Zhu, "System-level energy management for periodic real-time tasks," *2006 27th IEEE International Real-Time Systems Symposium (RTSS'06)*, pp. 313–322, 2006.

[17] S.-Y. Huang, J. Zeng, X. Deng, S. Wang, A. Sifat, B. Bharmal, J.-B. Huang, R. Williams, H. Zeng, and C. Jung, "Rtailor: Parameterizing soft error resilience for mixed-criticality real-time systems," in *2023 IEEE Real-Time Systems Symposium (RTSS)*, IEEE, 2023.

[18] S. Lee, H. Baek, H. Woo, K. G. Shin, and J. Lee, "Ml for rt: Priority assignment using machine learning," *2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pp. 118–130, 2021.

[19] Z. Bo, Y. Qiao, C. Leng, H. Wang, C. Guo, and S. Zhang, "Developing real-time scheduling policy by deep reinforcement learning," *2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pp. 131–142, 2021.

[20] S. Wang, J. Chen, X. Deng, S. A. Hutchinson, and F. Dellaert, "Robot calligraphy using pseudospectral optimal control in conjunction with a novel dynamic brush model," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6696–6703, 2020.