Exploring Machine Learning for Faster Mapping and Scheduling of Automotive Applications on ADAS Platforms

By Rafael Sterzinger, Wolfgang Koch, and Ralph Hoch

ABSTRACT

Goal: Faster Task Mapping & Scheduling on ADAS Platforms.

Method: MIP + ML + BO.

Key Contributions:

- Simplified MIP with load-balancing.
- 65% reduction in solving time through better configs found by BO.
- 77% average precision in prediction decision variables correctly for more effective branching.

Motivation:

- **ADAS** enhances safety through **sensors** and **automated actions.**
- Current scheduling is fixed, lacking runtime migration for better resource utilization.
- Increasing **complexity** of ADAS (thousands of tasks, dependencies) demands **faster mapping and scheduling**.
- ML techniques can accelerate this process.

Related Work:

- Previously: **Simulated Annealing** and **Genetic Algorithms** for heuristic solutions. [1]
- Now: Use **ML** to speed up **MIP**, exploiting recurring problem structures, similar to:
  - Power Generation Scheduling [2]
  - Server Load Balancing [3]

Problem Formulation:

1. **Sets and Parameters**
   - J: **Set of tasks:** Period p_j, Deadline d_j, Earliest-Activation a_j, Worst-Case Computation Time w_j
   - C: **Set of cores:** Macro-Tick of core m_c
2. **Decision Variables**
   - $x_{j,c}$: 1 if task j is assigned to core c, 0 otherwise (binary)
   - $k_{j,h}$: Periodicity of task j in chain h (integer)
   - s_j: Start time of task j (integer)
3. **Objective Function**

$$\text{Minimize } \frac{1}{|C|}\sum_{c\in C}\left|\sum_{j\in J} x_{j,c} - \mu\right|,$$

4. **Constraints:**

Only one core per task:

$$\sum_{c\in C} x_{j,c} = 1 \quad \forall j \in J$$

Task finishes before its deadline:

$$s_j + x_{j,c}\cdot\left\lceil\frac{w_j}{m_c}\right\rceil \le d_j \quad \forall j \in J, \forall c \in C$$

Start time is after the earliest activation:

$$s_j \ge a_j \quad \forall j \in J$$

No overlap of two tasks assigned to the same core:

$$s_i + \left\lceil\frac{w_i}{m_c}\right\rceil \le s_j \vee s_j + \left\lceil\frac{w_j}{m_c}\right\rceil \le s_i$$

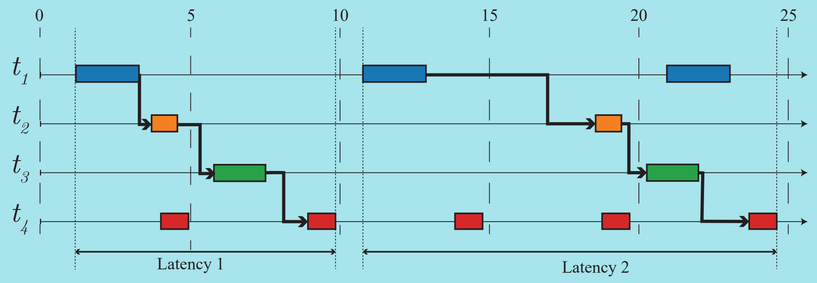
BAYESIAN OPTIMIZATION

Objective: Learn optimal configurations, that minimizes solving time, for settings: *Presolving, Heuristics, Separation, and Emphasis.*

Configuration Reduction: These high-level settings reduce the amount of configurations to $|\mathcal{C}| = 4^3 \times 10$, however, a full exploration is still computationally intensive.

Method: Use BO to learn relation between configurations and the resulting solving time to find optimal settings for the MIP-Solver.

Dependency Constraints: Tasks are **organized in chains** with a specified **maximum end-to-end latency.** They consist of source, processing, and sink tasks. For each source task, there needs to be a correct sequence of the remaining tasks that is within budget.



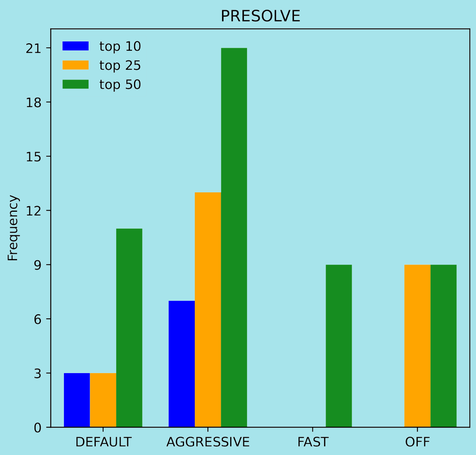
An example of a task chain - consisting of four tasks with periodicity, 10, 15, 15, 5 - illustrating its execution order and resulting different latencies.

Bayesian Optimization Process:

1. Split into **training** and **test** sets.
2. Conduct initial **burn-in phase** with a Latin-Hypercube sequence.
3. Perform **sampling** based on **probability of improvement:**

$$PI(c) = P\Big(f(c) \ge f(c^+) + \kappa\Big)$$

   to approximate **worst-case** solving time.
4. Evaluate on training set to approximate worst-case solving time.
5. Select **top ten configurations and aggregate** emphasis settings.



The impact of *Presolving* is significant, with a clear preference for *aggressive*.

Test Case	Default		Optimal		#Chains
	$\hat{y}_{\max}$	$y_{\max}$	$\hat{y}_{\max}$	$y_{\max}$	Fit/Eval.
ADAS 100%	115.55	201.53	61.30	74.88	5/5
ADAS 200%	18.66	3.93	6.46	3.76	6/20
ADAS 300%	67.66	12.30	9.68	12.16	7/30

Estimated and actual worst-case solving time when the MIP-Solver is in its default configuration and our learned optimal one.

PRIMAL HEURISTICS

Objective: Learn primal heuristics to enhance MIP-Solver performance by selecting configurations and improving the solving process.

Significance: Primal heuristics enable early identification of feasible solutions, facilitating aggressive pruning of the Branch-and-Bound search tree of the MIP-Solver.

Method: Utilize ML to learn to predict and fix decision variables early on in the context of ADAS platforms to quicker find good solutions.

90th percentile, and worst-case solving times for learned solver configurations: simpler instances benefit greatly, reducing worst-case time by around 65%.

Primal Heuristics Process:

1. **Dataset:** Small-scale instances solved to optimality to learn the binary variables. Representation: *linkage graph* where nodes are decision variables and edges link variables with shared constraints.
2. **Training:** The GCN learns to predict the probability of each decision variable being active in the optimal solution
3. **Inference:** For new instances, the GCN estimates these probabilities to guide the solver's branching decisions to improve the search process.

Test Case	ADAS 400%		ADAS 500%	
	AP	AC	AP	AC
GCN	**76.39**	**73.47**	**76.70**	**73.52**
LR	63.51	71.86	64.81	72.12
XGB	61.60	57.30	60.94	57.25

For the different classifiers, there is a clear ranking: GCN > LR > XGB. The maximum average precision is attained by the GCN with 77%.

References:

[1] S. D. McLean, S. S. Craciunas, E. Alexander Juul Hansen, and P. Pop, "Mapping and Scheduling Automotive Applications on ADAS Platforms using Metaheuristics," in 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation

[2] R. Sterzinger, J. Poland, M. B. Paulus, and D. Chetelat, "Learning to Predict Security Constraints for Large-Scale Unit Commitment Problems," in 2023 IEEE PES Innovative Smart Grid Technologies Europe

[3] M. Paulus and A. Krause, "Learning to Dive in Branch and Bound," in 2023 36th Advances in Neural Information Processing Systems

Project Website

TECHNISCHE UNIVERSITÄT WIEN

FFG Promoting Innovation.

Corresponding Author: Rafael Sterzinger; E-Mail: rafael.sterzinger@tuwien.ac.at
Poster for the International Conference on Machine Learning and Applications - ICMLA, 2024, Miami, Florida, USA

This work has been partly funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) in the "ICT of the Future" program