Improvement of Optimization using Learning Based Models in Mixed Integer Linear Programming Tasks

Xiaoke Wang*, Batuhan Altundas*, Zhaoxin Li*, Aaron Zhao, and Matthew Gombolay

***Abstract*— Mixed Integer Linear Programs (MILPs) are essential tools for solving planning and scheduling problems across critical industries such as construction, manufacturing, and logistics. However, their widespread adoption is limited by long computational times, especially in large-scale, real-time scenarios. To address this, we present a learning-based framework that leverages Behavior Cloning (BC) and Reinforcement Learning (RL) to train Graph Neural Networks (GNNs), producing high-quality initial solutions for warm-starting MILP solvers in Multi-Agent Task Allocation and Scheduling Problems. Experimental results demonstrate that our method reduces optimization time and variance compared to traditional techniques while maintaining solution quality and feasibility.**

I. INTRODUCTION

Mixed Integer Linear Programs (MILPs) serve as a fundamental framework for combinatorial optimization problems, facilitating solutions across a wide range of planning and scheduling tasks in logistics [1], construction [2] and manufacturing [3]. These problems often involve making time-sensitive, resource-constrained decisions about what actions to take, when to take them, and how to coordinate them—challenges central to planning and scheduling problems in a wide range of industrial sectors [4]. As MILP aims to solve NP-hard problems such as Task Allocation and Scheduling [5, 6], there are significant challenges in terms of computation time, particularly for large-scale or time-sensitive applications [7]. Traditional solution techniques used in the MILP Solvers, such as Branch-and-Bound (B&B) and constraint generation, require computational resources to converge into an optimal solution [8]. To enable the practical deployment of intelligent robotic systems in construction environments, reducing solver latency becomes crucial.

Warm-starting has emerged as a promising strategy to accelerate MILP solvers by providing high-quality initial solutions, reducing the number of iterations needed for convergence [9]. In particular, BC—a supervised learning approach—has emerged as a promising method for learning policies replicating expert demonstrations in optimization tasks. RL also has been adopted to fine-tune BC's performance further. Our paper explores the use of BC and RL to warm-start MILPs for planning and scheduling applications in complex, real-world construction environments—reducing computational overhead while ensuring solution quality and operational reliability.

II. RELATED WORKS

A. Classical Methods for Solving MILPs

Warm-starting has long been used to accelerate MILP solvers by reusing information from previous instances. The general approach is to start each B&B from the previous solver run's final B&B leaves [10]. Ralphs et al. [11] proposed another method that utilizes previously computed B&B trees and dual-derived information to efficiently re-solve new problem instances.

*These authors have contributed equally.

B. Machine Learning for MILPs

Machine learning has increasingly been integrated into MILP solving, both by enhancing solver internals and by generating useful external guidance. On the solver side, approaches include using generative models to learn branching policies that mimic strong branching decisions [8] and learning more effective primal heuristics [12]. In addition, by leveraging partial or sub-optimal solutions generated by learning-based methods, previous work has shown to achieve state-of-the-art performance [13]. BC, for instance, treats optimization trajectories as supervised learning datasets of state-action pairs, enabling models to imitate expert strategies [6]. However, BC often struggles to generalize to unseen instances due to distribution shift. To mitigate this, online fine-tuning BC policies with RL has proven effective [9, 14], allowing models to adapt based on solver feedback and improve performance in unfamiliar environments.

C. Warm Starting in MILPs

Recent works have proposed learning branching strategies [8] within MILP solvers. While promising, such an approach often requires tight integration with solver internals and needs to run model training multiple times, which is computationally heavy. Other works optimize via initializing solvers with expert-like solutions [1, 15], but it is limited in scalability guarantees [7]. GNN has shown the ability to provide sub-optimal solutions at different scales for NP-Hard Problems such as task allocation and scheduling [3, 6]. In this paper, we will build on these advances and investigate the use of the BC+RL fine-tuning framework to train GNNs to warm-start MILP, which retains compatibility with off-the-shelf solvers and supports large-scale, temporally constrained multi-agent scheduling.

III. METHODOLOGY

A. Multi-Agent Task Domain

We develop a simulation environment tailored to construction-inspired multi-agent task allocation and motion planning scenarios, an example shown in Figure 1, which tackles the challenge of optimizing long-term sequential decision-making in a continuous domain with obstacles. Our environment incorporates agents with heterogeneous velocities and task makespan to represent the heterogeneity