CSE667: Decision Making for Multi Robot Systems

Technical Report

Title: Scheduling Operator Assistance for Shared Autonomy in Multi-Robot Teams [1]

Authors: Yifan Cai, Abhinav Dahiya, Nils Wilde, Stephen L. Smith

Report By: Akanksha Singal (2021008) and Arnav Gupta (2021236)

May 5, 2025

Keywords

Shared Autonomy, Multi-robot Systems, Human-Robot Interaction, Operator Scheduling, Mixed Integer Linear Programming (MILP)

1 Impact of the Work

The paper[1] presents a novel approach for human operator allocation in robot systems.

Challenges

The main challenges involves:

- Human operators cannot supervise and assist a large number of robots on their own.
- Efficient allocation of limited human operator among multiple autonomous robots, each operating independently
- Scalability issues when solving the scheduling problem optimally for large robot teams with many tasks
- Managing between robot task readiness and operator availability, which complicates the scheduling sequence
- Balancing computational efficiency with solution quality, ensuring practical usability without significant sacrifices in performance

Significance

This work is significant as it allows efficient shared autonomy in multi-robot systems by reducing the mission completion time (makespan) through intelligent scheduling of human operator assistance. The proposed solution addresses scalability issues associated with exact optimization approaches like MILP, allowing practical deployment of human-in-the-loop decision support systems in large and complex multi-robot environments.

Contributions

The main contributions of the paper are:

- Formulation of the multi-robot teleoperation scheduling problem as a Mixed Integer Linear Program (MILP) to generate optimal schedules for humans and robots to minimize the time of robot mission completion.

- Development of Iterative Greedy algorithm, which allows for scalability of the problem.

- Introduction of Greedy Insertion and Block Removal approach to iteratively improve teleoperation schedules by minimizing robot and operator idle times.

- Empirical validation demonstrating that the proposed algorithm consistently outperforms other greedy baselines showing an efficient and scalable solution.

Applicability

The proposed methods are applicable in a wide range of domains where human operators assist autonomous robots, including search-and-rescue missions, smart manufacturing, autonomous logistics, and healthcare robotics. In these domains, reducing operator workload and minimizing overall task completion time is crucial for success, safety, and efficiency. The framework in the paper can be integrated into systems to provide real-time scheduling advice to human supervisors managing large robot fleets.

2 Technicalities of the problem definition

The problem objective is of scheduling operator assistance in a multi-robot system, where each robot $k \in K$ must complete a sequence of tasks p_k, composed of tasks e_j^k for $j = 1, \ldots, N_k$. Each task can be executed autonomously in time α_j^k or under teleoperation in time β_j^k (where $\beta_j^k \leq \alpha_j^k$). A human operator can teleoperate at most one robot at a time. Also, the task mode cannot change once started.

The objective is to find a teleoperation schedule S, i.e., an ordered subset of tasks assigned for teleoperation, that minimizes the overall mission *makespan* $\mu(S)$, defined as the time when the last robot completes its mission.

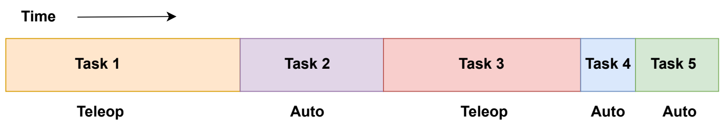

Figure 1: Scheduling Timeline Diagram for a Single Robot

Problem Definition and Formal Expressions

The objective is to determine a teleoperation schedule S that minimizes the overall mission completion time, or *makespan* $\mu(S)$, defined as the time when all robots have completed their respective missions.

The formal optimization problem is:

$$\text{Minimize} \quad \mu(S)$$

Given the set K of robots, the missions $\{p_1, \ldots, p_K\}$ for each robot, and the autonomous and teleoperation completion times α_j^k and β_j^k for each task.

The problem is NP-hard due to the combinatorial nature of choosing which tasks to teleoperate and the dynamic dependencies between robot task completion times and operator availability. As the problem size grows, solving the exact MILP formulation becomes computationally expensive, motivating the development of an efficient anytime heuristic (Iterative Greedy) for practical large-scale applications.

Table of Symbols and Definitions

Symbol	**Definition**
K	Set of robots, $K = \{1, \ldots, K\}$.
M	Set of human operators, $M = \{1, \ldots, M\}$ (used in multi-operator extension).
p_k	Mission assigned to robot k, represented as a sequence of tasks.
e_{kj}	j-th task of robot k.
α_j^k	Completion time for task e_{kj} under autonomous operation.
β_j^k	Completion time for task e_{kj} under teleoperation ($\beta_j^k \leq \alpha_j^k$).
S	Teleoperation schedule, a sequence of teleoperated tasks.
$\mu(S)$	Makespan of the schedule S, i.e., total time until all robot missions are completed.
x_j^k	Binary variable: 1 if task e_{kj} is teleoperated, 0 otherwise.
x_{jm}^k	Binary variable: 1 if task e_{kj} is teleoperated by operator m (in multi-operator case).
τ_j^k	Scheduled start time for task e_{kj}.
ε_j^k	Scheduled finish time for task e_j^k, computed as $\tau_j^k + (1 - x_j^k)\alpha_j^k + x_j^k\beta_j^k$.
μ	Optimization variable representing the team's makespan in the MILP formulation.
N_k	Total number of tasks in the mission p_k assigned to robot k.
$\overline{N}$	Total number of tasks across all robots, $\overline{N} = \sum_k N_k$.

Table 1: Table of symbols and their definitions used in the paper.

3. Approach

The core problem addressed in the paper is scheduling a single human operator's assistance among multiple robots, each with its own sequence of tasks. Every task can either be performed autonomously or with teleoperation, with the latter being faster. The operator, however, can assist only one robot at any time, and thus the goal is to identify an optimal subset of tasks for teleoperation and determine their order to minimize the overall mission completion time (makespan).

3.1 Formal Problem Formulation (MILP)

The task scheduling is modeled as a Mixed Integer Linear Program (MILP). For each robot $k \in K$, let the task sequence be $p_k = \{e_{k1}, e_{k2}, \ldots, e_{kN_k}\}$, where:

- α_j^k is the duration of task e_{kj} under autonomous execution,

- β_j^k is the faster duration under teleoperation ($\beta_j^k \leq \alpha_j^k$),
- $x_{kj} \in \{0,1\}$ indicates if task e_{kj} is teleoperated,
- τ_{kj} and ϵ_{kj} are the start and end times for the task.

The end time is computed as:

$$\epsilon_{kj} = \tau_{kj} + (1 - x_{kj})\alpha_j^k + x_{kj}\beta_j^k$$

The objective is to minimize:

$$\mu(S) = \max_k \epsilon_{kN_k}$$

Subject to the constraints:

$$\begin{aligned} \tau_{k1} &\geq 0 \quad \forall k \\ \tau_{kj} &\geq \epsilon_{k(j-1)} \quad \forall j \geq 2 \\ x_{kj} + x_{li} = 2 \Rightarrow \tau_{kj} &\geq \epsilon_{li} \text{ or } \tau_{li} \geq \epsilon_{kj} \quad \forall (k \neq l) \\ \mu &\geq \epsilon_{kN_k} \quad \forall k \end{aligned}$$

This MILP captures operator exclusivity, task dependencies, and global completion constraints. However, solving it becomes computationally expensive for large values of $|K|$ and $N = \sum_k N_k$ due to combinatorial explosion in binary variables and time-indexed constraints.

3.2 Anytime Heuristic: Iterative Greedy Algorithm

To enable scalability, the paper introduces an **Iterative Greedy** heuristic that produces near-optimal schedules efficiently. It starts with all tasks set to autonomous execution and iteratively inserts beneficial teleoperation tasks.

Phase 1: Greedy Insertion

- Identify the robot k^* with the highest mission time.
- From its remaining tasks, select e_{k^*j} that yields the largest local makespan reduction if switched to teleoperation.
- Check operator availability and insert if feasible.

Phase 2: Block Removal

- Analyze the operator timeline to locate idle gaps caused by scheduling conflicts.
- Fill these slots with low-impact teleoperation tasks from other robots that do not extend $\mu(S)$.

These phases run iteratively until no further improvement is possible. The algorithm maintains feasibility by enforcing non-overlap in inserted tasks and avoids full enumeration by evaluating only greedy candidates.

Complexity and Benefits:

- Greedy Insertion focuses on critical-path robots, directly targeting makespan bottlenecks.
- Block Removal improves operator utilization by eliminating unproductive idle time.
- The algorithm yields schedules within 5% of the MILP optimal for small instances and scales linearly with the number of robots and tasks.

This hybrid design of MILP-backed formalism and efficient greedy refinement allows the framework to be both theoretically sound and practically viable.

4. Future Work

- **Online and Receding Horizon Scheduling:** Extend the current offline scheduling model to an online setting, where the operator's schedule is dynamically updated as robots progress through their tasks. This enables responsiveness to unexpected delays, failures, or new task arrivals during mission execution.
- **Integration with Motion and Path Planning:** Coupling teleoperation task selection with robot motion planning can avoid infeasible operator assignments due to travel delays or robot collisions, especially in cluttered or dynamic environments.
- **Multi-Operator Scheduling with Skill Constraints:** Support heterogeneous human operators, where skill sets vary across tasks. The scheduler must handle operator-task matching while balancing load and avoiding overlapping assignments.

References

[1] Yifan Cai, Abhinav Dahiya, Nils Wilde, and Stephen L Smith. Scheduling operator assistance for shared autonomy in multi-robot teams. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 3997–4003. IEEE, 2022.